

Advances in Intelligent Systems and Computing 531

Weidong Chen

Koh Hosoda

Emanuele Menegatti

Masahiro Shimizu

Hesheng Wang *Editors*

Intelligent Autonomous Systems 14

Proceedings of the 14th International
Conference IAS-14

 Springer

Advances in Intelligent Systems and Computing

Volume 531

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within “Advances in Intelligent Systems and Computing” are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Advisory Board

Chairman

Nikhil R. Pal, Indian Statistical Institute, Kolkata, India
e-mail: nikhil@isical.ac.in

Members

Rafael Bello Perez, Universidad Central “Marta Abreu” de Las Villas, Santa Clara, Cuba
e-mail: rbellop@uclv.edu.cu

Emilio S. Corchado, University of Salamanca, Salamanca, Spain
e-mail: escorchado@usal.es

Hani Hagra, University of Essex, Colchester, UK
e-mail: hani@essex.ac.uk

László T. Kóczy, Széchenyi István University, Győr, Hungary
e-mail: koczy@sze.hu

Vladik Kreinovich, University of Texas at El Paso, El Paso, USA
e-mail: vladik@utep.edu

Chin-Teng Lin, National Chiao Tung University, Hsinchu, Taiwan
e-mail: ctlin@mail.nctu.edu.tw

Jie Lu, University of Technology, Sydney, Australia
e-mail: Jie.Lu@uts.edu.au

Patricia Melin, Tijuana Institute of Technology, Tijuana, Mexico
e-mail: epmelin@hafsamx.org

Nadia Nedjah, State University of Rio de Janeiro, Rio de Janeiro, Brazil
e-mail: nadia@eng.uerj.br

Ngoc Thanh Nguyen, Wroclaw University of Technology, Wroclaw, Poland
e-mail: Ngoc-Thanh.Nguyen@pwr.edu.pl

Jun Wang, The Chinese University of Hong Kong, Shatin, Hong Kong
e-mail: jwang@mae.cuhk.edu.hk

More information about this series at <http://www.springer.com/series/11156>

Weidong Chen · Koh Hosoda
Emanuele Menegatti · Masahiro Shimizu
Hesheng Wang
Editors

Intelligent Autonomous Systems 14

Proceedings of the 14th International
Conference IAS-14

 Springer

Editors

Weidong Chen
Shanghai Jiao Tong University
Shanghai
China

Masahiro Shimizu
Osaka University
Osaka
Japan

Koh Hosoda
Osaka University
Osaka
Japan

Hesheng Wang
Shanghai Jiao Tong University
Shanghai
China

Emanuele Menegatti
University of Padua
Padua
Italy

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-48035-0 ISBN 978-3-319-48036-7 (eBook)
DOI 10.1007/978-3-319-48036-7

Library of Congress Control Number: 2016955328

© Springer International Publishing AG 2017

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

For the first time, the International Conference on Intelligent Autonomous Systems (IAS) is held in China and organized by Shanghai Jiao Tong University. This year, the 14th IAS is held in Shanghai from July 3 to July 7, 2016 at Shanghai Jiao Tong University with the goal of bringing together researchers and engineers from around the world to present their latest research accomplishments, innovations, and visions in the field of robotics and artificial intelligence.

The theme of IAS-14 is “**Frontier of intelligent autonomous systems,**” reflecting the ever growing interests in research, development, and applications in the dynamic and exciting areas of robotics.

After going through a rigorous peer-review process, totally, 85 papers have been selected for oral presentation. The scientific program consists of 24 technical sessions in 4 tracks, running over two days in comfortable slots of 20 min. The conference program is enriched by four plenary talks from Maarja Kruusmaa, Davide Scaramuzza, Jun Ota, Barry Trimmer, and one keynote talk by Dong Sun.

We are grateful for the assistance of the staff members and students of Shanghai Jiao Tong University. We would also thank all members of the organization committee for their help in organizing this event. Special thanks should be given to all the authors for contributing their research works, and all the participants for making the conference a memorable event.

We hope that these proceedings, and especially the participation in the IAS-14 Conference and its related events, will inspire new ideas, foster new research, and create new friendships which can grow into fruitful collaborations. IAS-14 promises to be a great event for all participants, with excellent technical and social programs.

Shanghai, China
June 2016

Weidong Chen
Masahiro Shimizu

Contents

Part I Embodied-Brain Systems Science

Clustering Latent Sensor Distribution on Body Map for Generating Body Schema	3
Tomohiro Mimura, Yoshinobu Hagiwara, Tadahiro Taniguchi and Tetsunari Inamura	
Improvement of EMG Pattern Recognition by Eliminating Posture-Dependent Components	19
Akira Ishii, Toshiyuki Kondo and Shiro Yano	
Quantification of Temporal Parameters for Tripedalism	31
Arito Yozu, Dai Owaki, Masashi Hamada, Takuya Sasaki, Qi An, Tetsuro Funato and Nobuhiko Haga	
Proposal of a Stance Postural Control Model with Vestibular and Proprioceptive Somatosensory Sensory Input	39
Ping Jiang, Shouhei Shirafuji, Ryosuke Chiba, Kaoru Takakusaki and Jun Ota	
Simultaneous Localization, Mapping and Self-body Shape Estimation by a Mobile Robot	53
Akira Taniguchi, Lv WanPeng, Tadahiro Taniguchi, Toshiaki Takano, Yoshinobu Hagiwara and Shiro Yano	
Objective Measurement of Dynamic Balance Function by the Simultaneous Measurement of the Center of Gravity (COG) and Center of Pressure (COP)	69
Masahiko Mukaino, Fumihiro Matsuda, Ryoma Sassa, Kei Ohtsuka, Nobuhiro Kumazawa, Kazuhiro Tsuchiyama, Shigeo Tanabe and Eiichi Saitoh	

Development of a Master–Slave Finger Exoskeleton Driven by Pneumatic Artificial Muscles	77
Takuya Urino, Shuhei Ikemoto and Koh Hosoda	
Temporal Structure of Muscle Synergy of Human Stepping Leg During Sit-to-Walk Motion	91
Qi An, Hiroshi Yamakawa, Atsushi Yamashita and Hajime Asama	
Part II Field Robot	
Fast and Accurate Crop and Weed Identification with Summarized Train Sets for Precision Agriculture	105
Ciro Potena, Daniele Nardi and Alberto Pretto	
Supervised Autonomy for Exploration and Mobile Manipulation in Rough Terrain	123
Max Schwarz, Sebastian Schüller, Christian Lenz, David Droeschel and Sven Behnke	
Behavior-Based Collision Avoidance Using a Cylinder-Coordinate Octree	143
Daniel Schmidt, Fabian Göckel and Karsten Berns	
An Integrated Robotic System for Autonomous Brake Bleeding in Rail Yards	157
Huan Tan, Shiraj Sen, Arpit Jain, Shuai Li, Viktor Holovashchenko, Ghulam Baloch, Omar Al Assad, Romano Patrick, Douglas Forman, Yonatan Gefen, Pramod Sharma, Frederick Wheeler, Charles Theurer and Balajee Kannan	
Part III Flying Robot	
I Believe I Can Fly—Gesture-Driven Quadrotor Control Based on a Fuzzy Control System	177
Leon Spohn, Marcus Bergen, Nicolai Benz, Denis Vonscheidt, Hans-Jörg Haubner and Marcus Strand	
ROS-Gazebo Supported Platform for Tag-in-Loop Indoor Localization of Quadrocopter	185
Shuyuan Wang and Tianjiang Hu	
The Project PRISMA: Post-Disaster Assessment with UAVs	199
Carmine Tommaso Recchiuto and Antonio Sgorbissa	
An Automatic Collision Avoidance Approach to Assist Remotely Operated Quadrotors	213
Bruno Giovanini, Hugo A. Oliveira and Paulo F.F. Rosa	

Part IV Hand

Shared Control with Flexible Obstacle Avoidance for Manipulator 229
 Zhixuan Wei, Weidong Chen and Hesheng Wang

An RGB-D Visual Application for Error Detection in Robot Grasping Tasks 243
 Ester Martinez-Martin, David Fischinger, Markus Vincze and Angel P. del Pobil

Sensorless In-Hand Caging Manipulation 255
 Yusuke Maeda and Tomohiro Asamura

Development of New Cosmetic Gloves for Myoelectric Prosthetic Hand by Using Thermoplastic Styrene Elastomer 269
 Yoshiko Yabuki, Kazumasa Tanahashi, Suguru Hoshikawa, Tatsuhiro Nakamura, Ryu Kato, Yinlai Jiang and Hiroshi Yokoi

Part V Human Robot Interaction

Cloud-Based Task Planning for Smart Robots 285
 Elisa Tosello, Zhengjie Fan, Alejandro Gatto Castro and Enrico Pagello

Tracking Control of Human-Following Robot with Sonar Sensors 301
 Wei Peng, Jingchuan Wang and Weidong Chen

Active Sensing for Human Activity Recognition by a Home Bio-monitoring Robot in a Home Living Environment 315
 Keigo Nakahata, Enrique Dorronzoro, Nevrez Imamoglu, Masashi Sekine, Kahori Kita and Wenwei Yu

Part VI Legged Robot

An Underactuated Biped Robot Guided via Elastic Elements: EKF-Based Estimation of Ankle Mechanical Parameters 329
 Roberto Bortoletto, Thomas Reilly, Enrico Pagello and Davide Piovesan

Higher Jumping of a Biped Musculoskeletal Robot with Foot Windlass Mechanism 343
 Xiangxiao Liu, Yu Duan, Andre Rosendo, Shuhei Ikemoto and Koh Hosoda

A Motion Planning Architecture for Conveyance Tasks with a Quadruped Robot 357
 Giulio Cerruti, Wei-Zhong Guo and Fulvio Mastrogiovanni

Part VII Localization and Path Planning

A Simple and Efficient Path Following Algorithm for Wheeled Mobile Robots	375
Goran Huskić, Sebastian Buck and Andreas Zell	
Localization Issues for an Autonomous Robot Moving in a Potentially Adverse Environment	389
Antonio D'Angelo and Dante Degl'Innocenti	
Vector-AMCL: Vector Based Adaptive Monte Carlo Localization for Indoor Maps	403
Richard Hanten, Sebastian Buck, Sebastian Otte and Andreas Zell	
A Virtual Force Guidance Law for Trajectory Tracking and Path Following	417
Xun Wang, Jianwei Zhang, Daibing Zhang and Lincheng Shen	
A 2D Voronoi-Based Random Tree for Path Planning in Complicated 3D Environments	433
Zheng Fang, Chengzhi Luan and Zhiming Sun	
3D FieldLut Algorithm Based Indoor Localization for Planar Mobile Robots Using Kinect	447
Xiaoxiao Zhu, Qixin Cao and Wenshan Wang	
A Feature-Based Mutual Information and Wavelet Method for Image Fusion	459
Yulong Liu, Yiping Chen, Cheng Wang and Ming Cheng	
Part VIII Measurement	
An Intelligent RGB-D Video System for Bus Passenger Counting	473
Daniele Liciotti, Annalisa Cenci, Emanuele Frontoni, Adriano Mancini and Primo Zingaretti	
A Powerful and Cost-Efficient Human Perception System for Camera Networks and Mobile Robotics	485
Marco Carraro, Matteo Munaro and Emanuele Menegatti	
Influence of Stimulus Color on Steady State Visual Evoked Potentials	499
Leeyee Chu, Jacobo Fernández-Vargas, Kahori Kita and Wenwei Yu	
Accelerated Adaptive Local Scanning of Complicated Micro Objects for the PSD Scanning Microscopy: Methods and Implementation	511
Mehdi Rahimi and Yantao Shen	

Part IX Medical Engineering

Development of a Robotic Thumb Rehabilitation System Using a Soft Pneumatic Actuator and a Pneumatic Artificial Muscles-Based Parallel Link Mechanism 525
 Kouki Shiota, Tapio V. J. Tarvainen, Masashi Sekine, Kahori Kita and Wenwei Yu

An fMRI Study on Vibration Stimulation Synchronized Mirror Therapy 539
 Kazuya Imai, Kahori Kita and Wenwei Yu

Robot Patient Imitating Paralysis Patients for Nursing Students to Learn Patient Transfer Skill. 549
 Chingszu Lin, Zhifeng Huang, Masako Kanai-Pak, Jukai Maeda, Yasuko Kitajima, Mitsuhiro Nakamura, Noriaki Kuwahara, Taiki Ogata and Jun Ota

Part X Multi-robot Systems

A Self-reconfigurable Robot M-Lattice 563
 Shihe Tian, Zhen Yang, Zhuang Fu and Hui Zheng

Adaptive Synchronized Formation Control Considering Communication Constraints. 573
 Zhe Liu, Weidong Chen, Junguo Lu, Jingchuan Wang and Hesheng Wang

A Distributed Self-healing Algorithm for Global Optimal Movement Synchronization of Multi-robot Formation Network 587
 Xiangyu Fu, Weidong Chen, Zhe Liu, Jingchuan Wang and Hesheng Wang

Consensus of Discrete-Time Linear Networked Multi-agent Systems Subject to Actuator Saturation 601
 DU Boyang, Zhang Guoliang, XU Jun, Zeng Jing and Zhang Yong

Partitioning Strategies for Multi-robot Area Coverage with No Communication. 615
 Cristiano Nattero and Fulvio Mastrogiovanni

Max-Sum for Allocation of Changing Cost Tasks 629
 James Parker, Alessandro Farinelli and Maria Gini

Part XI Robot Control

LuGre Model Based Hysteresis Compensation of a Piezo-Actuated Mechanism. 645
 Guangwei Wang and Qingsong Xu

Sample-Data Control of Optimal Tracking for a Class of Non-linear Systems via Discrete-Time State Dependent Riccati Equation	659
Fan Yang, Guoliang Zhang, Zhenan Pang and Lei Yuan	
Real-Time Flight Test Track Filtering and Association Using Kalman Filter and QDA Classifier	675
Kundong Wang and Yao Ge	
Simulation of Time Delay Compensation Controller for a Mobile Robot Using the SMC and Smith Predictor	687
Dong-Hyuk Lee, Jae-Hun Jung, Ha-Neul Yoon, Young-Sik Park and Jang-Myung Lee	
Differential Wheeled Robot Navigation Based on the Smoothing A* Algorithm and Dead-Reckoning	695
Daowei Jiang and Liang Yuan	
Adaptive Impedance Control for Docking of Space Robotic Arm Based on Its End Force/Torque Sensor	713
Gangfeng Liu, Changle Li, Caiwei Song, Liyi Li and Jie Zhao	
A Robotic Hardware-in-the-Loop Simulation System for Flying Objects Contact in Space	725
Chenkun Qi, Xianchao Zhao, Feng Gao, Anye Ren and Yan Hu	
Particle Filter on Episode for Learning Decision Making Rule	737
Ryuichi Ueda, Kotaro Mizuta, Hiroshi Yamakawa and Hiroyuki Okada	
Part XII Robot Design	
Efficient Measurement of Fibre Orientation for Mapping Carbon Fibre Parts with a Robotic System	757
Morris Antonello, Matteo Munaro and Emanuele Menegatti	
A Honeycomb Artifacts Removal and Super Resolution Method for Fiber-Optic Images	771
Zhong Zheng, Bin Cai, Jieting Kou, Wei Liu and Zengfu Wang	
Water-Surface Stability Analysis of a Miniature Surface Tension-Driven Water Strider Robot	781
Jihong Yan, Xinbin Zhang, Jie Zhao and Hegao Cai	
Mechanism Allowing a Mobile Robot to Apply a Large Force to the Environment	795
Shouhei Shirafuji, Yuri Terada and Jun Ota	

Part XIII Robot Learning

Self-improving Robot Action Management System with Probabilistic Graphical Model Based on Task Related Memories 811
 Yuki Furuta, Yuto Inagaki, Kei Okada and Masayuki Inaba

View-Based Teaching/Playback with Photoelasticity for Force-Control Tasks 825
 Yoshinori Nakagawa, Soichi Ishii and Yusuke Maeda

Discovering the Relationship Between the Morphology and the Internal Model in a Robot System by Means of Neural Networks 839
 Angel J. Duran and Angel P. del Pobil

Part XIV Robot Navigation

Combining Feature-Based and Direct Methods for Semi-dense Real-Time Stereo Visual Odometry 855
 Nicola Krombach, David Droeschel and Sven Behnke

Outdoor Robot Navigation Based on View-Based Global Localization and Local Navigation 869
 Yohei Inoue, Jun Miura and Shuji Oishi

Using OpenStreetMap for Autonomous Mobile Robot Navigation 883
 Patrick Fleischmann, Thomas Pfister, Moritz Oswald and Karsten Berns

A Lane Change Detection and Filtering Approach for Precise Longitudinal Position of On-Road Vehicles 897
 Tianyi Li, Ming Yang, Xiaojun Xu, Xiang Zhou and Chunxiang Wang

Part XV Robot Vision

Synchronous Dataflow and Visual Programming for Prototyping Robotic Algorithms 911
 Sebastian Buck, Richard Hanten, C. Robert Pech and Andreas Zell

Depth-Based Frontal View Generation for Pose Invariant Face Recognition with Consumer RGB-D Sensors 925
 Giorgia Pitteri, Matteo Munaro and Emanuele Menegatti

Lighting- and Occlusion-Robust View-Based Teaching/Playback for Model-Free Robot Programming 939
 Yusuke Maeda and Yoshito Saito

Part XVI Robotic Arm

Development of a Portable Compliant Dual Arm Robot 955
 Zhifeng Huang, Chingszu Lin, Ping Jiang, Taiki Ogata and Jun Ota

Design, Analysis and Simulation of a Device for Measuring the Inertia Parameters of Rigid Bodies 965
 Yu Liu, Song Huang, Li Jiang and Hong Liu

Development of a Myoelectric Hand Incorporating a Residual Thumb for Transmetacarpal Amputees 977
 Yuta Murai, Suguru Hoshikawa, Shintaro Sakoda, Yoshiko Yabuki, Masahiro Ishihara, Tatsuhiro Nakamura, Takehiko Takagi, Shinichiro Takayama, Yinlai Jiang and Hiroshi Yokoi

Part XVII Robotics for ITER Remote Handling

Structural Design of Multi-joint Foldable Robot Manipulator for Remote Inspection in Experimental Advanced Superconducting Tokamak (EAST) 993
 Baoyuan Wu, Weibin Guo, Yi Liu, Qiang Zhang, Ling Zhou, Qingquan Yan, Zhong Zheng and Zengfu Wang

The Design of Pipe Cutting Tools for Remote Handling in Maintenance Manipulator for Tokamak. 1003
 Xizhe Zang, Zhenkun Lin, Yixiang Liu, Yanhe Zhu and Jie Zhao

Optimal Trajectory Planning for Manipulators with Flexible Curved Links 1013
 Liang Zhao, Hesheng Wang and Weidong Chen

Predictive Display for Telerobot Under Unstructured Environment. . . . 1027
 Qing Wei, NaiLong Liu and Long Cui

Control System Design and Implementation of Flexible Multi-joint Snake-Like Robot for Inspecting Vessel. 1037
 Yi Liu, Qingquan Yan, Qiang Zhang, Weibin Guo, Odbal, Baoyuan Wu and Zengfu Wang

Research on the Tokamak Equipment CAsk (TECA) for Remote Handling in Experimental Advanced Superconducting Tokamak (EAST). 1049
 Lifu Gao, Weibin Guo, Baoyuan Wu, Daqing Wang, Yuan Liu, Yi Liu, Qiang Zhang, Zengfu Wang and Liangbin Guo

3D SLAM for Scenes with Repetitive Texture Inside Tokamak Chamber 1061
 Wei Liu, Zhong Zheng, Odbal, Bin Cai and Zengfu Wang

Design and Implementation of Wormlike Creeping Robot System Working at the Bottom of the Nuclear Fusion Vessel 1073
 Qiang Zhang, Ling Zhou, Yi Liu and Zengfu Wang

Part XVIII Sensor Network

Indoor Positioning System Based on Distributed Camera Sensor Networks for Mobile Robot 1089
Yonghoon Ji, Atsushi Yamashita and Hajime Asama

Precise and Reliable Localization of Intelligent Vehicles for Safe Driving 1103
Liang Li, Ming Yang, Lindong Guo, Chunxiang Wang and Bing Wang

A Unified Controller for the Connectivity Maintenance of a Robotic Router Networks 1117
Li Xiangpeng, Huang Haibo, Yang Hao and Sun Dong

The Elderly’s Falling Motion Recognition Based on Kinect and Wearable Sensors 1129
Pang Nana, Dong Min, Zhao Yue, Chen Xin and Bi Sheng

Online Adjusting Task Models for Ubiquitous Robotic Systems 1143
Wenshan Wang, Qixin Cao, Qiang Qiu and Gilbert Cheruiyot

Improved Skeleton Estimation by Means of Depth Data Fusion from Multiple Depth Cameras 1155
Marco Carraro, Matteo Munaro, Alina Roitberg and Emanuele Menegatti

Author Index 1169

Part I
Embodied-Brain Systems Science

Clustering Latent Sensor Distribution on Body Map for Generating Body Schema

Tomohiro Mimura, Yoshinobu Hagiwara,
Tadahiro Taniguchi and Tetsunari Inamura

Abstract A computational theory model is constructed to generate a body map and a body schema. It is assumed that the fetus has a body map and a body schema to control and understand its body and generates them based on tactile information obtained by random motion in the womb called general movement. As a first step, in this research, a computational theory model is proposed to generate the body Map based on tactile information and to estimate nodes of the body schema by machine learning methods.

Keywords Body map · Body schema · Clustering

1 Introduction

Human beings can unconsciously control their bodies based on a musculoskeletal system that is constructed by integrating multimodal information in their brains [1]. Human beings also can understand surrounding environments through their bodies, because they have a topological map in their brains that represents the position of each recipient cell in their bodies [2, 3]. In this study, we assume that human beings have two types of models in their brains: a multilink system called a body schema

T. Mimura (✉) · Y. Hagiwara (✉) · T. Taniguchi (✉)
Ritsumeikan University, 1-1-1 Noji Higashi, Kusatsu, Shiga 525-8577, Japan
e-mail: t.mimura@em.ci.ritsumei.ac.jp

Y. Hagiwara
e-mail: yhagiwara@em.ci.ritsumei.ac.jp

T. Taniguchi
e-mail: taniguchi@em.ci.ritsumei.ac.jp

T. Inamura (✉)
National Institute of Informatics, 2-1-2 Hitotsubashi,
Chiyoda-ku, Tokyo 101-8430, Japan
e-mail: inamura@nii.ac.jp

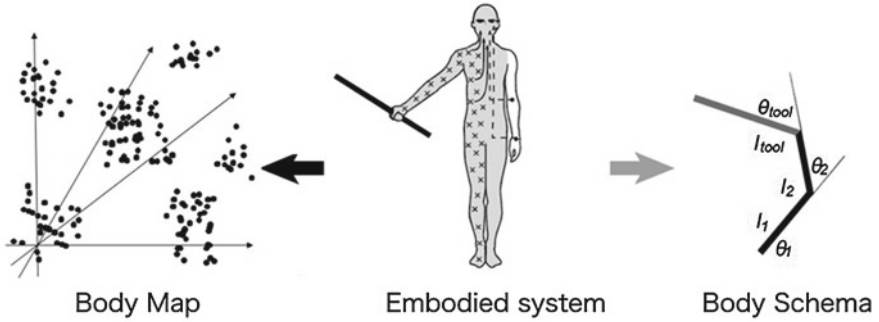


Fig. 1 A body map and a body schema in the human brain

used to control their bodies and a topological map called a body map used to understand the surrounding environment through their bodies as shown in Fig. 1.

In previous research, Prechtl et al. reported that a fetus obtains a body schema through random motion in the womb called “general movement” [4]. General movement is defined as spontaneous activity by the fetus manifested as a smooth and random whole-body motion observed from 8.5 to 9.5 weeks. It is assumed that the fetus generates a body schema based on the relationship between motion and tactile information obtained by general movement because neighboring recipient-cells return similar tactile information owing to the high viscosity of amniotic fluid.

In this study, we aim to construct a computational model that generates a body map and a body schema based on tactile information obtained by random motion, i.e., general movement.

The computational model generates a body map by using multidimensional scaling (MDS) [5] and information metric based on tactile information, and estimate nodes of a body schema by using Dirichlet process Gaussian mixture model (DPGMM) [6] from the generated body map.

2 Previous Research

2.1 Body Map

Human beings have body maps that are topological maps for their bodies in the brain [2]. Like an exhaustive patchwork body maps cover not only external body parts such as fingers, shoulders, elbows, and so on but also viscera.

In a previous study of body maps, Olsson et al. proposed a method to calculate an information measure based on the information of sensors attached to the robot AIBO and to replace each sensor with a two-dimensional map based on the calculated information measure [7]. However, this study is not intended for estimation of a multilink system.

Mori et al. provided a constructive model of a human fetus that forms behaviors and understanding regarding its body through general movements in the womb of its mother [8]. Sugiura et al. solved a similar problem by using the time difference of arrival of signals and multidimensional scaling [9]. Kuniyoshi et al. generated topographic somatosensory cluster map that is plotted the fetus's sensor points on a 2 dimensional space as a result of a self-organizing process [10].

2.2 *Body Schema*

Even though the human body has multiple degrees of freedom, human beings can control their bodies without considering the angle of joints and the length of links. It is assumed that a body schema enables human beings to unconsciously control their bodies. Humans have a multilink system in their brains for understanding their bodies. Henry et al. define body schema as a unconscious process that dynamically works when human beings control their motion and pose [11].

In a previous study of body schema, Martinez et al. estimated a multilink system using an active-learning method based on vision information [12]. Nabeshima et al. proposed a model that enables robots to extend their body schema, when robots use tools, based on multimodal information. It is difficult to apply these methods to our study which to estimate body map and body schema based on tactile information because these methods require multimodal information including vision information.

3 Proposed Method

3.1 *Overview*

Figure 2 shows an approach of this study used to generate a body map and a body schema. The proposed method consists of the following three steps. First, a topological map is generated based on tactile information obtained from receptor cells on the embodied system, as shown in Fig. 2a. Second, nodes of body schema are estimated by clustering the generated topological map, as shown in Fig. 2b. The third step entails generating a tree structure based on tactile information and the estimated nodes of the body schema, as shown in Fig. 2c. As a preliminary stage to generate a body schema, our approach generates a tree structure instead of a multilink system. In this paper, we propose a method to generate a body map and a method to estimate the nodes of the body schema, as shown in Fig. 2a and b, respectively.

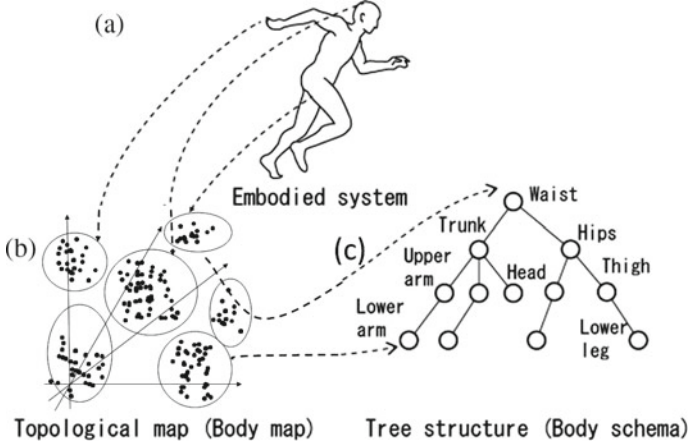


Fig. 2 Overview of the proposed approach. **a** Body map generation. **b** Node estimation. **c** Body schema generation

3.2 Generation of the Body Map

To generate a body map, we used the summation of conditional entropy and multi-dimensional scaling (MDS) [5]. Olsson et al. defined the summation of conditional entropy as an information metric between sensors [13]. The information metric d satisfies the following metric space axioms:

$$\begin{aligned}
 D(X, Y) &= D(Y, X) \quad (\text{Symmetry}) \\
 D(X, Y) &= 0 \text{ iff } Y = X \quad (\text{Equivalence}) \\
 D(X, Z) &\leq D(X, Y) + D(Y, Z) \quad (\text{Triangle Inequality}),
 \end{aligned}$$

when the model has M tactile sensors, the information metric $d(S_i, S_j)$ between two tactile sensors i and j is calculated from the equation:

$$D(S_i, S_j) = H(S_i|S_j) + H(S_j|S_i), \quad (1)$$

where S_i and S_j are sets of tactile data at two sensors i and j , respectively. The conditional entropy $H(S_i|S_j)$ is defined by the equation:

$$\begin{aligned}
 H(S_j|S_i) &:= \sum_{x \in S_i} p(x) H(S_j|S_i = x) \\
 &:= - \sum_{x \in S_i} \sum_{y \in S_j} p(x, y) \log_2 p(y|x),
 \end{aligned} \quad (2)$$

where x and y are tactile data and elements of S_i and S_j , respectively.

$p(y|x)$ and $p(x)$ have the following relationships from Bayes' theorem:

$$p(y|x) = \frac{p(x, y)}{p(x)}, \quad (3)$$

$$p(x) = \sum_{y \in \mathcal{S}_j} p(x, y), \quad (4)$$

and $p(x, y)$ is calculated by using the equation:

$$p(x, y) = \frac{\sum_{t=0}^T f(i, j, t, x, y)}{T}. \quad (5)$$

Here t is an index number of tactile-information data divided by a specified time interval Δt and T is a maximum index number of tactile-information data. $f(i, j, t, x, y)$ is defined by

$$f(i, j, t, x, y) = \begin{cases} 1 & \text{if } a_{i,t} = x \wedge a_{j,t} = y, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

$$(7)$$

The tactile-information data $a_{i,t}$ and $a_{j,t}$ are quantized tactile information and have values between 0 and N .

The information metric is thus defined as the distance between two tactile sensors S_i, S_j .

In this study, we create a body map using MDS to resemble the distance of the information metric. MDS is a technique of placing data in Euclidean space using information between data (e.g., distance and similarity etc.).

3.3 Estimation of Nodes for the Body Schema

This subsection describes a method to estimate the nodes of the body schema by clustering the body map which is generated in the previous subsection.

A Dirichlet process Gaussian mixture model (DPGMM) [6] based on the non-parametric Bayes model is used for clustering the body map because the number of nodes of the body schema is unknown. The DPGMM can estimate the suitable number of clusters, which is different from a clustering method such as a Gaussian mixture model (GMM) [14]. The nonparametric Bayes model is a probabilistic model including the Dirichlet process, the beta Bernoulli process, and so on. Specifically, we use the stick-breaking process (SBP) [15] which is one of the construction methods of the Dirichlet process.

The DPGMM can cluster data by estimating the mixture number of Gaussian distributions by the Dirichlet process without indicating the number of clusters. Then the DPGMM can estimate suitable nodes of the body schema from the body map.

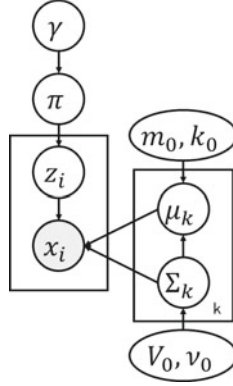


Fig. 3 Graphical model of the DPGMM

Figure 3 shows a graphical model of the DPGMM. The generation model for the DPGMM consists of the equations:

$$\pi \sim \text{GEM}(\gamma), \quad (8)$$

$$z_i \sim \text{Mult}(z|\pi), \quad (9)$$

$$\Sigma_k^{-1} \sim \mathcal{W}(A|V_0, \nu_0), \quad (10)$$

$$\mu_k \sim \mathcal{N}(\mu|m_0, (k_0 A)^{-1}), \quad (11)$$

$$x_i \sim \mathcal{N}(x|\mu_{z_i}, \Sigma_{z_i}), \quad (12)$$

where $\text{Mult}(\cdot)$ in Eq. (9) is the multinomial distribution, $\mathcal{N}(\cdot)$ in Eq. (11) is the multinormal Gaussian distribution, and $\mathcal{W}(\cdot)$ in Eq. (10) is the Wishart distribution. The multinomial distribution, multinormal Gaussian distribution, and Wishart distribution are calculated, respectively, by using

$$\text{Mult}(x|\pi) = \frac{n!}{z_1 \cdots z_k} \pi_1^{z_1} \cdots \pi_k^{z_k}, \quad (13)$$

$$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(\sqrt{2\pi})^m \sqrt{|\Sigma|}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right], \quad (14)$$

$$\mathcal{W}(A|V, \nu) = \frac{|A|^{(\nu-d-1)/2} \cdot \exp\left[-\frac{1}{2}\text{tr}(V^{-1}A)\right]}{2^{\nu d/2} \cdot \pi^{d(d-1)/4} \cdot |V|^{\nu/2} \cdot \prod_{j=1}^d \Gamma\left(\frac{\nu+1-j}{2}\right)}. \quad (15)$$

The stick-breaking representation of the GMM probability is used in this model.

4 Verification Experiments in Virtual Space

4.1 Overview of Experiments

To verify the effectiveness of the proposed method, two experiments for generating a body map and estimating nodes of the body schema were performed in virtual space. As a first experiment for generating a body map, we compared experimental results obtained by using the proposed method based on the information metric to those obtained by the conventional method based on the Pearson correlation coefficient. As a second experiment for estimating nodes of the body schema, we compared experimental results obtained by the proposed method based on the DPGMM and other clustering methods.

4.2 Virtual Multilink Model for Verification Experiments

The human body has a complex structure and multiple degrees of freedom. It is difficult to generate a musculoskeletal model based on a complete human body. A simple tree structure model shown in Fig. 4a was used for verification experiments instead of a human musculoskeletal model. Open Dynamics Engine (ODE) [16] was used as a simulator for verification experiments in virtual space.

As shown in Fig. 4b, 850 tactile sensors were attached to the surface of five links. (Note that the choice of colors Fig. 4a and b is arbitrary) Each tactile sensor records its three-dimensional position and direction at time intervals of $\Delta t = 0.25$ s. The total time was 1500 s in both experiments. Thus the number of sampled tactile data, T , for each sensor was 6000.

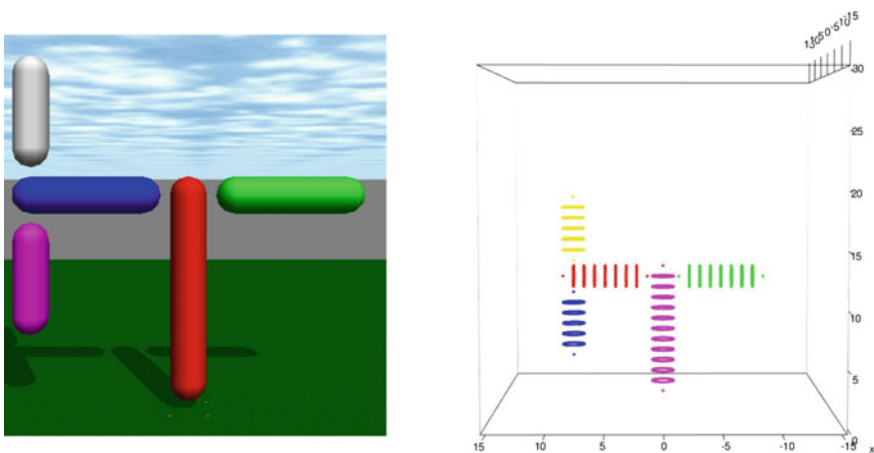


Fig. 4 Tree structure model for the verification experiments in virtual space. *Right* image is (a) and *left* image is (b). **a** Exterior of the model. **b** Placement of sensors

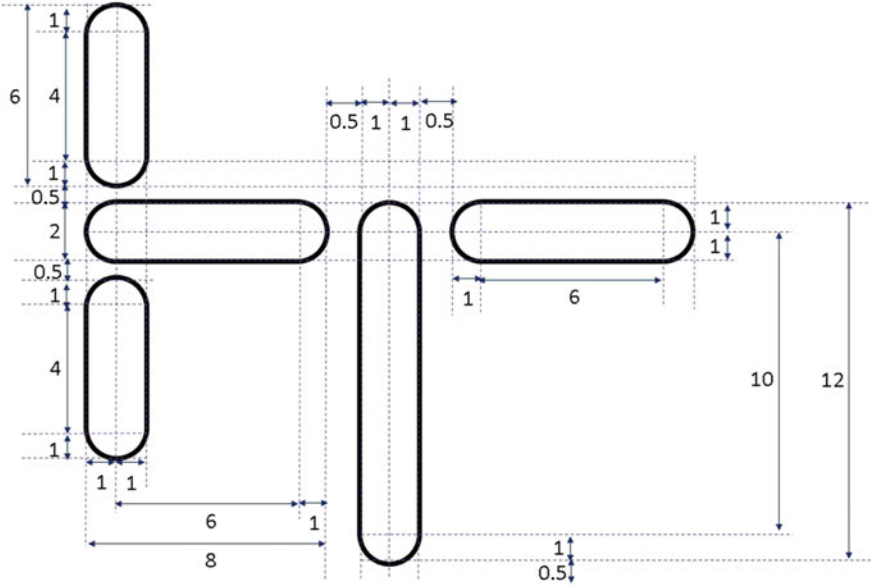


Fig. 5 Design of the tree structure model

Figure 5 shows details of the tree structure model in Fig. 4a, indicating the length and width of each link.

The tree structure model consists of five links and four universal joints. Each link has a capsule shape, and the universal joint has two axes. General movements by the fetus were reproduced by random commands for each joint.

Tactile information $f_{i,t}$ of a sensor i at time t is calculated by using the following equations:

$$f_{i,t} = \begin{cases} \rho \times (p'_{i,t} \cdot n_{i,t})^2 & \text{if } 0 < p'_{i,t} \cdot n_{i,t}, \\ 0 & \text{otherwise} \end{cases}, \quad (16)$$

where $n_{i,t}$ is the normal vector to the surface of sensor i at time t , ρ is the density of amniotic fluid and is set to 1010 kg/m^3 , and $p'_{i,t}$ is the speed of sensor i at time t and is calculated by using the equation

$$p'_{i,t} = \frac{p_{i,t} - p_{i,t-1}}{\Delta t}, \quad (18)$$

where $p_{i,t}$ is the tree-dimensional position of sensor i at time t and is obtained from ODE.

The tactile-information data $a_{i,t}$ in Eq. (16) is a quantized value of $f_{i,t}$. The Adjusted Rand Index (ARI) [17] is used to evaluate the similarity between grand truth and a clustering result in the evaluation experiments.

If the similarity between grand truth and a clustering result becomes 1, it means that the clustering result is completely correct.

4.3 Experiment for Generating the Body Map

4.3.1 Experimental Condition

To verify the validity of the proposed method, we performed an experiment to compare the body map generated by the proposed method based on the information metric to that generated by using the conventional method based on the Pearson correlation coefficient. The Pearson correlation coefficient is used to indicate the similarity of correlation.

The Pearson correlation coefficient $r(S_i, S_j)$ between sets of tactile data at two sensors i and j is defined by

$$r(S_i, S_j) = \frac{\sum_{t=0}^T (f_{i,t} - \bar{f}_i)(f_{j,t} - \bar{f}_j)}{\sqrt{\sum_{t=0}^T (f_{i,t} - \bar{f}_i)^2} \sqrt{\sum_{t=0}^T (f_{j,t} - \bar{f}_j)^2}}, \quad (19)$$

where $f_{i,t}$ is calculated by using the Eqs. (16) and (17), t is the tactile data index over the time sequence 0 to T , and \bar{f}_i and \bar{f}_j are calculated by using the equations

$$\bar{f}_i = \frac{\sum_{t=0}^T f_{i,t}}{T} \quad \bar{f}_j = \frac{\sum_{t=0}^T f_{j,t}}{T}$$

The Pearson correlation coefficient varies between -1 and 1 . If two data have a strong correlation, the Pearson correlation coefficient becomes 1 . In the experiment, D_{Person} is used as the distance between two data points and is calculated by using

$$D_{Person}(S_i, S_j) = 1 - r(S_i, S_j). \quad (20)$$

4.3.2 Experimental Result

Figure 6a shows body maps generated by using the conventional method based on the Pearson correlation coefficient and MDS. The left-hand and right-hand sides in

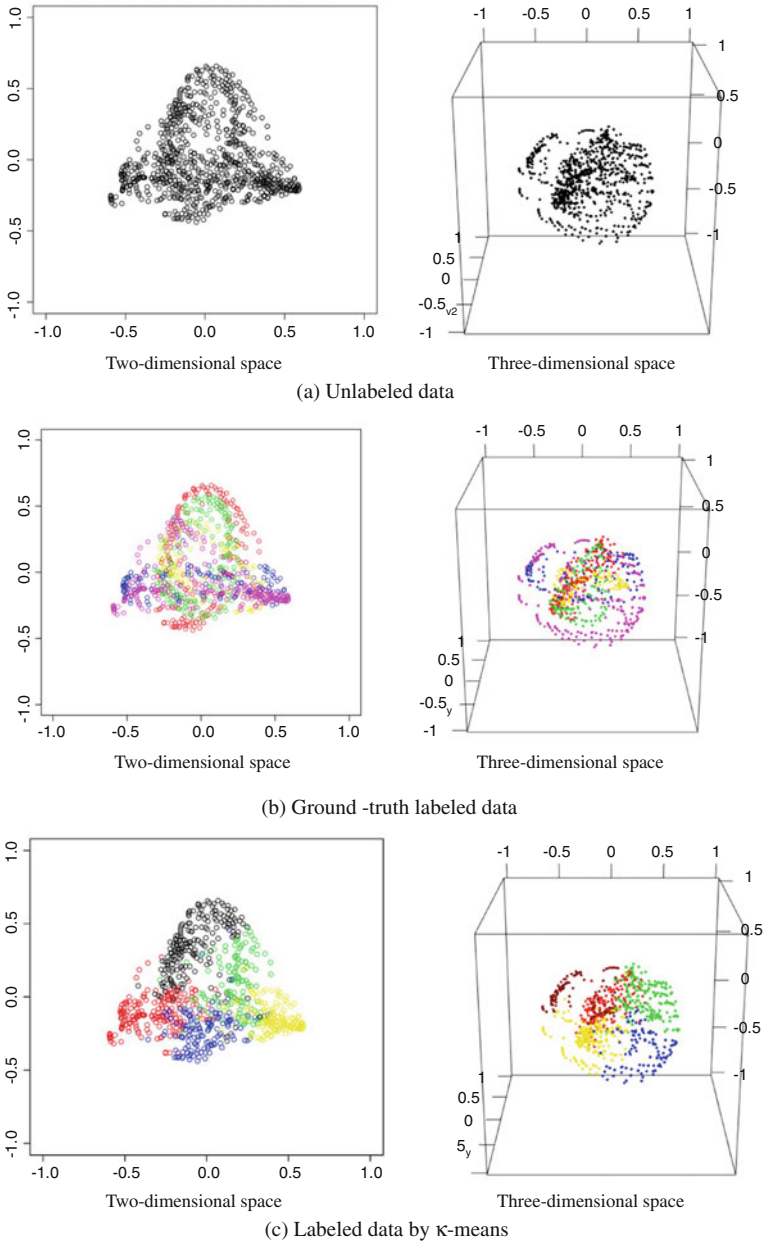


Fig. 6 Body maps generated by using the conventional method

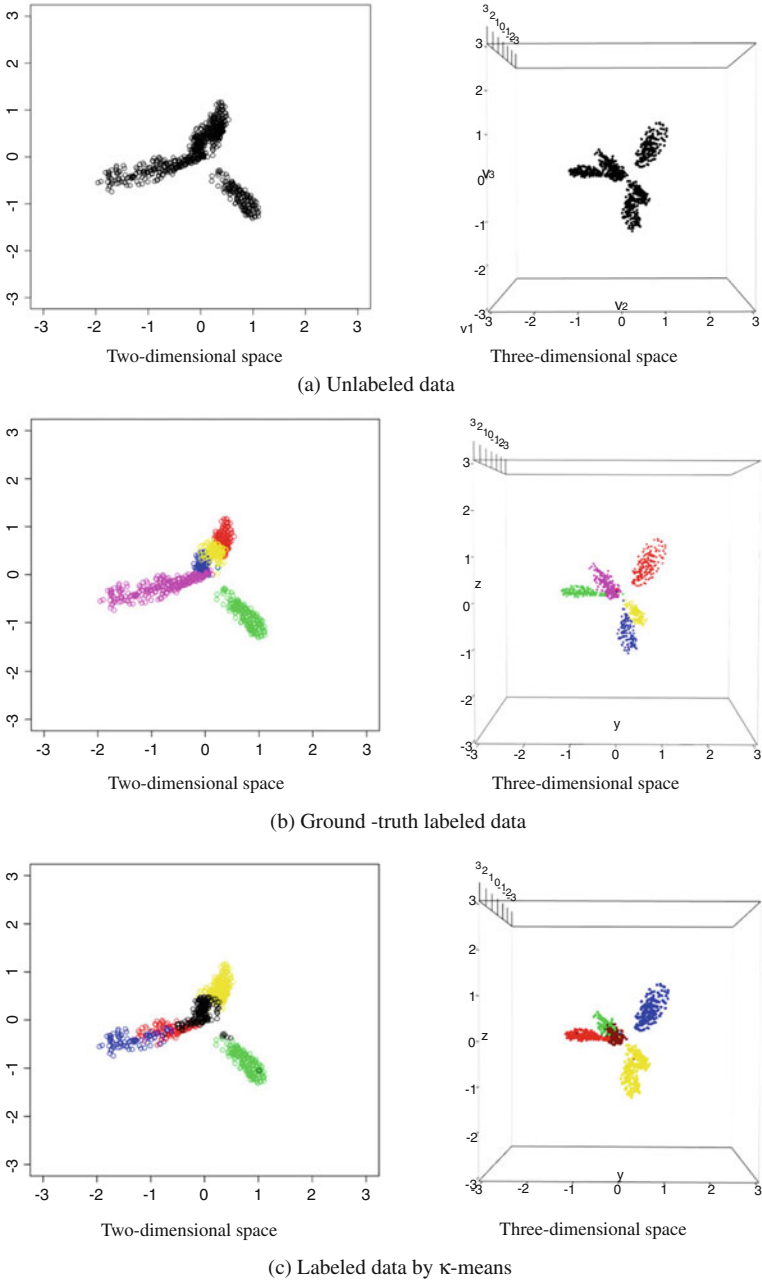


Fig. 7 Body maps generated by using the proposed method

Table 1 Comparison of clustering results between the proposed method and the conventional method in different dimensional spaces

Dimensional spaces	2D	3D	4D	5D	6D	7D	8D	9D
Conventional method	0.086	0.073	0.101	0.093	0.115	0.116	0.124	0.118
Proposed method	0.546	0.732	0.974	0.973	0.972	0.972	0.972	0.972

Fig. 6a show the generated body maps in two and three dimensional space, respectively. To evaluate the accuracy of the generated body maps, we compared ground-truth labeled data and labeled data by using k -means method [14] in each body map shown in Fig. 6a. Figure 6b and c show the ground-truth labeled data and labeled data by k -means, respectively. As shown in Fig. 6b and c, it was confirmed that the ground-truth data and labeled data by k -means are different in the body maps generated by using the conventional method.

In contrast, Fig. 7a shows body maps generated by the proposed method based on the information metric and MDS. Figure 7b and c also show the ground-truth labeled data and labeled data by k -means, respectively. As shown in Fig. 7b and c, it was confirmed that the ground-truth-labeled data and the labeled data by k -means are similar in the body maps generated by using the proposed method.

We evaluated the obtained result using the adjusted rand index (ARI), which quantifies the performance of a clustering task. Table 1 shows the evaluation of clustering results by k -means in the body maps generated by using the conventional and proposed methods. The first row in Table 1 shows the ARI of clustering results in the body maps generated by using the conventional method with different dimensional spaces from two to nine. The ARI values in the conventional method are low for all dimensional spaces. In contrast, the ARI values in the proposed method are high compared with the ARI values in the conventional method, as shown in the second row in Table 1. It is expected that the body maps generated by using the proposed method are effective for estimating the nodes of the body schema.

4.4 Experiment for Estimating Nodes of the Body Schema

4.4.1 Experimental Condition

The purpose of this experiment is to estimate nodes of the body schema based on the body map. The proposed method estimates nodes of the body schema by clustering the body map. Generally, k -meas [14], Ward’s method [18], and Gaussian Mixture Model (GMM) [14] are used for clustering data in which the number of clusters are known. However, it is difficult to apply these methods for clustering a body map because the number of clusters is unknown. Therefore the Dirichlet Process Gaussian Mixture Model (DPGMM) [6] is used for clustering the body map in the proposed method. In this experiment, the effectiveness of the proposed method

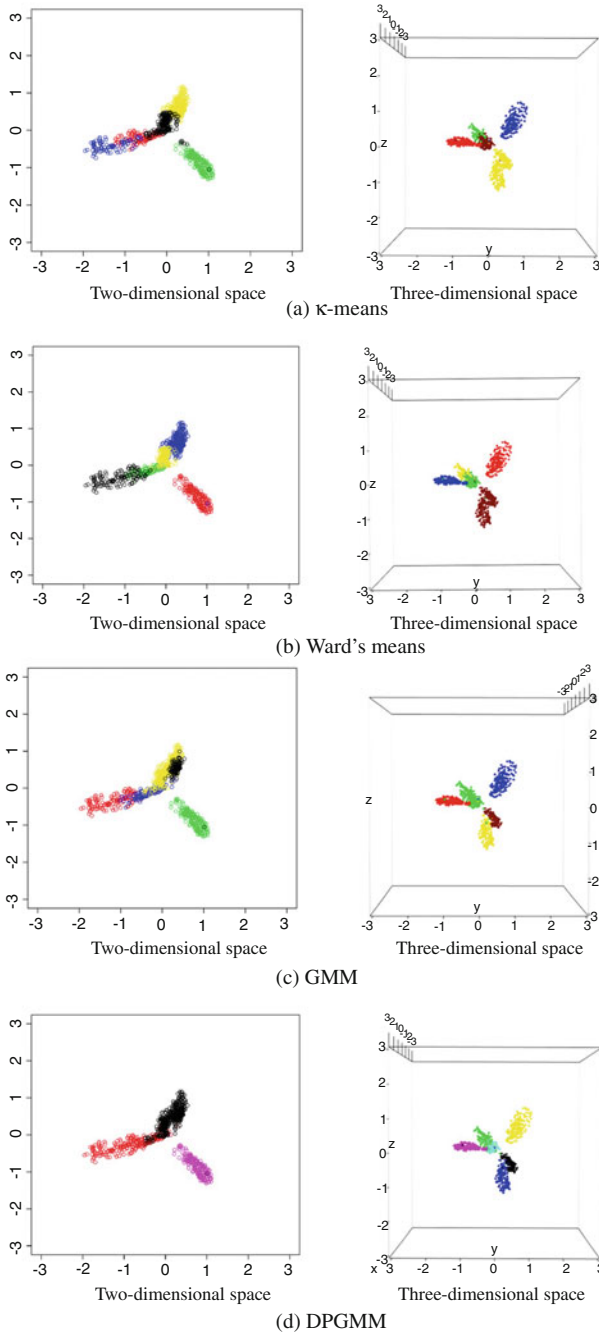


Fig. 8 Comparison of clustering results by k -means, Ward's method, the GMM, and the DPGMM

Table 2 ARI results of clustering two to nine dimensional body maps

Clustering method	2D	3D	4D	5D	6D	7D	8D	9D
<i>k</i> -means	0.546	0.732	0.974	0.973	0.972	0.972	0.972	0.972
Ward's method	0.628	0.736	0.973	0.973	0.973	0.973	0.973	0.973
GMM	0.512	0.963	0.974	0.978	0.974	0.975	0.979	0.980
DPGMM	0.650	0.753	0.910	0.977	0.974	0.978	0.842	0.831

Table 3 The number of estimated nodes in body schema

	2D	3D	4D	5D	6D	7D	8D	9D
The number of nodes in body schema	3	4	5	5	5	5	5	5

with the DPGMM is verified by comparing clustering results of the body map with baseline clustering-methods such as *k*-means, Ward's method, and the GMM.

4.4.2 Experimental Result

We evaluated clustering results by *k*-means, Ward's method, the GMM, and the DPGMM using the ARI [19]. Figure 8a, b, c, and d show clustering results of the body map by *k*-means, Ward's method, the GMM, and the DPGMM, respectively. Figure 8 confirms that the clustering results obtained by using the DPGMM are similar to other clustering results obtained by *k*-means, Ward's method, and the GMM.

Table 2 shows ARI values of clustering results of the body map obtained by using various clustering methods. Table 2 confirms that ARI values obtained by using the DPGMM are almost the same as those obtained by using other clustering methods: *k*-means, Ward's method, and the GMM. It is expected that an accurate estimation of nodes of the body schema can be obtained in the tree structure model in virtual space. Table 3 shows the number of estimated links in the body schema. It means successful body maps of generation by the proposed method and successful estimation of the number of links in body schemas by the DPGMM.

5 Conclusion

In this study, we proposed a novel method to generate a body map from tactile information using MDS and the sum of conditional entropy and estimated nodes of the body schema obtained by using the DPGMM. We performed two experiments for generating a body map and for estimating nodes of the body schema. In the exper-

iment for generating the body map, the effectiveness of the proposed method was confirmed by comparing with the conventional method based on the Pearson correlation coefficient. In the experiment for estimating nodes of the body schema, the effectiveness of the proposed method based on the DPGMM was confirmed by comparing with k -means, Ward's method, and the GMM as baseline clustering methods. These experimental results show that the proposed method can estimate nodes of the body schema only from tactile information.

In future work, we plan to build a generation model to generate a body schema from only tactile information.

Acknowledgements This research was partially supported by a Grant-in-Aid for Scientific Research on Innovative Areas 2015–2017 (15H01670) funded by the Ministry of Education, Culture, Sports, Science, and Technology, Japan.

References

1. Kuniyoshi, Y., Suzuki, S.: Dynamic emergence and adaptation of behavior through embodiment as coupled chaotic field. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004 (IROS 2004). Proceedings, vol. 2, pp. 2042–2049. IEEE (2004)
2. Blakeslee, S., Blakeslee, M.: The Body Has a Mind of Its Own: How Body Maps in Your Brain Help You Do (Almost) Everything Better. Random House (2008)
3. Yamada, Y., Fujii, K., Kuniyoshi, Y.: Impacts of environment, nervous system and movements of preterms on body map development: fetus simulation with spiking neural network. In: IEEE IC DL-EPIROB2013 (2013)
4. Prechtl, H.F.: State of the art of a new functional assessment of the young nervous system. An early predictor of cerebral palsy. *Early Human Dev.* **50**(1), 1–11 (1997)
5. Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* **29**(1), 1–27 (1964)
6. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: *Bayesian Data Analysis*, vol. 2. Taylor & Francis (2014)
7. Olsson, L.A., Nehaniv, C.L., Polani, D.: From unknown sensors and actuators to actions grounded in sensorimotor perceptions. *Connect. Sci.* **18**(2), 121–144 (2006)
8. Mori, H., Kuniyoshi, Y.: A human fetus development simulation: self-organization of behaviours through tactile sensation. In: IEEE 9th International Conference on Development and Learning (ICDL), pp. 82–87 (2010)
9. Sugiura, K., Matsubara, D., Katai, O.: Construction of robotic body schema by extracting temporal information from sensory inputs. In: SICE-ICASE International Joint Conference, pp. 302–307 (2006)
10. Kuniyoshi, Y., Yorozu, Y., Ohmura, Y., Terada, K., Otani, T., Nagakubo, A., Yamamoto, T.: From humanoid embodiment to theory of mind. In: *Embodied Artificial Intelligence*, pp. 202–218. Springer (2004)
11. Gallagher, S.: Body image and body schema: a conceptual clarification. *J. Mind Behav.* **7**(4), 5417554 (1986)
12. Martinez-Cantin, R., Lopes, M., Montesano, L.: Body schema acquisition through active learning. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 1860–1866. IEEE (2010)
13. Olsson, L., Nehaniv, C.L., Polani, D.: Measuring informational distances between sensors and sensor integration. In: *Artificial Life X*. MIT Press (2006)
14. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer (2006)

15. Sethuraman, J.: A constructive definition of dirichlet priors. Technical Report, DTIC Document (1991)
16. Smith, R.: Open dynamics engine. <http://www.ode.org/>
17. Hubert, L., Arabie, P.: Comparing partitions. *J. Classif.* **2**(1), 193–218 (1985)
18. Ward Jr., J.H.: Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* **58**(301), 236–244 (1963)
19. Yeung, K.Y., Ruzzo, W.L.: Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics* **17**(9), 763–774 (2001)

Improvement of EMG Pattern Recognition by Eliminating Posture-Dependent Components

Akira Ishii, Toshiyuki Kondo and Shiro Yano

Abstract Recently, myoelectric interfaces have been intensively studied in various research fields. Because electromyography (EMG) is a bioelectrical signal, it can be influenced by many disturbing factors, e.g., electrode displacement, postural changes, and individual-dependent features like condition of muscles, subcutaneous fat, skin surface, etc., thus, it is difficult to realize high classification accuracy. To solve the problem, an EMG pattern classification method, which decomposes raw EMG signals into user/motion-dependent components by using a bilinear model, has been proposed. This enabled to reduce the time for classifier re-learning, however classification accuracy has not yet been sufficient. In the current study, we propose a signal decomposing method in consideration of the effect by forearm postures, in order to extract informative factors that correctly reflect hand gestures. We investigated the influences of postural changes exert on the classification accuracy of hand gestures, and tried to separate not only user dependent factor, but also posture-dependent component from EMG signals. As a result, we found that postural change decreases classification accuracy of approximately 20 % and we confirmed availability of our proposed method.

Keywords Electromyography · Hand gesture recognition · Singular value decomposition

A. Ishii · T. Kondo (✉) · S. Yano
Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology, 2-24-16, Naka-cho, 184-8588, Koganei, Tokyo, Japan
e-mail: t_kondo@cc.tuat.ac.jp

A. Ishii
e-mail: ishii@livingsys.lab.tuat.ac.jp
URL: <http://www.livingsys.lab.tuat.ac.jp/>

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_2

1 Introduction

In recent years several wearable devices, such as a glass-type computer, and prosthetic limbs, have been researched and developed. To use these devices appropriately, natural user interfaces (e.g., gesture, gaze and voice recognition) are required. Among such, surface electromyography (EMG) has received much attention to use as natural input for the devices (e.g., prosthetic hand [1], wheelchair [2], etc.), because it has the advantage that it directly reflects user’s motor intention.

However, since EMG is a bioelectrical signal, it can be influenced by many factors (e.g., electrode displacement, posture, and individual-dependent features) [3–5]. Therefore, it is difficult to realize high classification accuracy [6–8].

To overcome these problems, Matsubara [9] proposed an EMG pattern classification method, which decomposes EMG signals into *user-dependent* and *motion-dependent* factors by using a bilinear model. Then they used only the motion dependent factor for the classification. This reduced time cost for classifier relearning, however, classification accuracy has not yet been sufficient. One of possible hypotheses is that the motion-dependent factor extracted by Matsubara’s algorithm may still include unnecessary components that depress the classification accuracy. In this paper, we investigate the validity of Matsubara’s algorithm, and we further propose an EMG signal decomposing method in consideration of the effect by forearm postures using a two-step bilinear model.

2 Method

Before explaining our proposed method, we concisely review a bilinear EMG model proposed in [9].

2.1 Bilinear Model

EMG signal y_k where k indicates electrode channel can be represented with two linear components \mathbf{z} and \mathbf{x} .

$$y_k = \mathbf{z}^T \mathbf{W}_k \mathbf{x} \quad (1)$$

where T represents transpose, $\mathbf{z} \in \mathbf{R}^I$ is a variable representing user-dependent factor, and $\mathbf{x} \in \mathbf{R}^J$ is a variable representing motion-dependent factor. $\mathbf{W}_k \in \mathbf{R}^{I \times J}$ is the weight matrix of the bilinear model.

We have a set of EMG data observed from multiple subjects over a set of motions as $\mathcal{D} = \{\{\mathbf{y}_k^{um} \in \mathbf{R}^N, l^m\}, u = 1 \sim U, m = 1 \sim M, k = 1 \sim K\}$, where K , U , M and N are the number of the dimensions of the EMG signal, the number of users, the number

of motions, and the number of samples in each motion, respectively. Accordingly, \mathbf{y}_k^{um} is a set of N samples of k_{th} EMG signals while user u performs motion m . Each motion m has corresponding class label l^m .

The objective function for optimizing the parameters of a bilinear model is formulated as,

$$\begin{aligned} E &= \sum_{u=1}^U \sum_{n=1}^N \sum_{m=1}^M \sum_{k=1}^K \|\mathbf{y}_{kn}^{um} - \mathbf{X}_m^T \mathbf{W}_k^T \mathbf{z}^u\|^2 \\ &= \sum_{u=1}^U \sum_{m=1}^M \sum_{k=1}^K \|\mathbf{y}_k^{um} - \mathbf{X}_m^T \mathbf{W}_k^T \mathbf{z}^u\|^2. \end{aligned} \quad (2)$$

In the optimization process, we try to find variables $\{\mathbf{x}^m, \mathbf{z}^u, \mathbf{W}_k\}$ for all u, m, k that minimize the objective function (2), in which $\mathbf{z}^u \in \mathbf{R}^I$ is the user-dependent factor for user u , and $\mathbf{x}_n^m \in \mathbf{R}^J$ is the motion-dependent factor for motion m . $\mathbf{X}_m = [\mathbf{x}_1^m \dots \mathbf{x}_N^m] \in \mathbf{R}^{J \times N}$ is a set of motion-dependent factors corresponding to motion m .

Here we define the matrix \mathbf{Y} as

$$\mathbf{Y} = \begin{bmatrix} (\mathbf{y}_1^{11})^T & \dots & (\mathbf{y}_1^{1M})^T \\ \vdots & \ddots & \vdots \\ (\mathbf{y}_K^{U1})^T & \dots & (\mathbf{y}_K^{UM})^T \end{bmatrix} \in \mathbf{R}^{KU \times NM}, \quad (3)$$

where VT represents vector-transpose, which is defined as a process stacking for a $KU \times NM$ matrix into a $NMK \times U$ matrix as

$$\mathbf{Y}^{VT} = \begin{bmatrix} \mathbf{y}_1^{11} & \dots & \mathbf{y}_1^{U1} \\ \vdots & \ddots & \vdots \\ \mathbf{y}_K^{1M} & \dots & \mathbf{y}_K^{UM} \end{bmatrix} \in \mathbf{R}^{NMK \times U}. \quad (4)$$

Furthermore, we define a motion-dependent factor \mathbf{X} as $\mathbf{X} = [\mathbf{X}_1, \dots, \mathbf{X}_M] \in \mathbf{R}^{J \times NM}$ and user-dependent factor \mathbf{Z} as $\mathbf{Z} = [\mathbf{z}^1, \dots, \mathbf{z}^U] \in \mathbf{R}^{I \times U}$. Based on these definitions and (1), we can obtain two equivalent Eqs. (5) and (6).

$$\mathbf{Y} = [\mathbf{W}^{VT} \mathbf{Z}]^{VT} \mathbf{X} \quad (5)$$

$$\mathbf{Y}^{VT} = [\mathbf{W} \mathbf{X}]^{VT} \mathbf{Z} \quad (6)$$

where \mathbf{W} is $IK \times J$ matrix. Here \mathbf{X} and \mathbf{Z} can be derived by using singular value decomposition (SVD) without unknown parameter matrix \mathbf{W} . First we initialize \mathbf{X} using SVD for \mathbf{Y} as $\mathbf{Y} \xrightarrow{SVD} \mathbf{U} \Sigma \mathbf{V}^T$ and \mathbf{X} can be estimated as the first J rows of \mathbf{V}^T . Then, we estimated \mathbf{X} using (5), and we derive \mathbf{Z} as the first I rows of \mathbf{V}^T where $[\mathbf{Y} \mathbf{X}^T]^{VT} \xrightarrow{SVD} \mathbf{U} \Sigma \mathbf{V}^T$. Next, using (6) and the estimated \mathbf{Z} , we derive the new

\mathbf{X} as the first J rows of \mathbf{V}^T , where $[\mathbf{Y}^{VT}\mathbf{Z}^T]^{VT} \xrightarrow{SVD} \mathbf{U}\Sigma\mathbf{V}^T$. The learning procedure usually converges within ten iterations [10]. Finally we obtain the weight matrix $\mathbf{W} = \left[[\mathbf{Y}\mathbf{X}^T]^{VT} \mathbf{Z}^T \right]^{VT}$ from (6). Here, the dimensions of \mathbf{X} and \mathbf{Z} , i.e., I and J were suitably set, respectively. Learning algorithm of the bilinear model is summarized in Algorithm 1.

Algorithm 1 Learning a Bilinear Model

Input: \mathbf{Y} (\mathbf{Y}^{VT})

Output: $\mathbf{X}, \mathbf{Z}, \mathbf{W}$

initialize set \mathbf{X} as the first J rows of \mathbf{V}^T of $\mathbf{Y} \rightarrow \mathbf{U}\Sigma\mathbf{V}^T$;

repeat

 Update \mathbf{Z} by the first I rows of \mathbf{V}^T of $[\mathbf{Y}\mathbf{X}^T]^{VT} \rightarrow \mathbf{U}\Sigma\mathbf{V}^T$;

 Update \mathbf{X} by the first J rows of \mathbf{V}^T of $[\mathbf{Y}^{VT}\mathbf{Z}^T]^{VT} \rightarrow \mathbf{U}\Sigma\mathbf{V}^T$;

until Convergence;

return $\mathbf{W} = \left[[\mathbf{Y}\mathbf{X}^T]^{VT} \mathbf{Z}^T \right]^T$;

To apply the model to novel users, we asked them to perform a pre-specified motion m to obtain EMG data \mathbf{Y}_m . Since we had already known a corresponding motion-dependent component \mathbf{X}_m in learning process, we can estimate user-dependent component \mathbf{z} of novel users by considering (6).

$$\mathbf{z} = \left[[\mathbf{W}\mathbf{X}_m]^{VT} \right]^+ \mathbf{Y}_m^{VT} \quad (7)$$

where $[\cdot]^+$ represents its Moore-Penrose pseudo-inverse matrix. Using derived user-dependent factor \mathbf{z} for novel users, we can estimate motion-dependent factor $\mathbf{X}_{m'}$ from new observations $\mathbf{Y}_{m'}$ of other motion m' by (5).

$$\mathbf{X}_{m'} = \left[[\mathbf{W}^{VT}\mathbf{z}]^{VT} \right]^+ \mathbf{Y}_{m'} \quad (8)$$

Thus we can use this $\mathbf{X}_{m'}$ to classify the motion.

2.2 Proposed Method

In Matsubara's study, EMG signals were decomposed into user-dependent and motion-dependent components, respectively. After the decomposition, they classified motion using only the motion-dependent component [9]. However, the accuracy 73% acquired by their method seems not to be sufficient. We considered that the motion dependent component that they extracted still includes unnecessary components influencing on the classification, and if it is true, we may be able to further

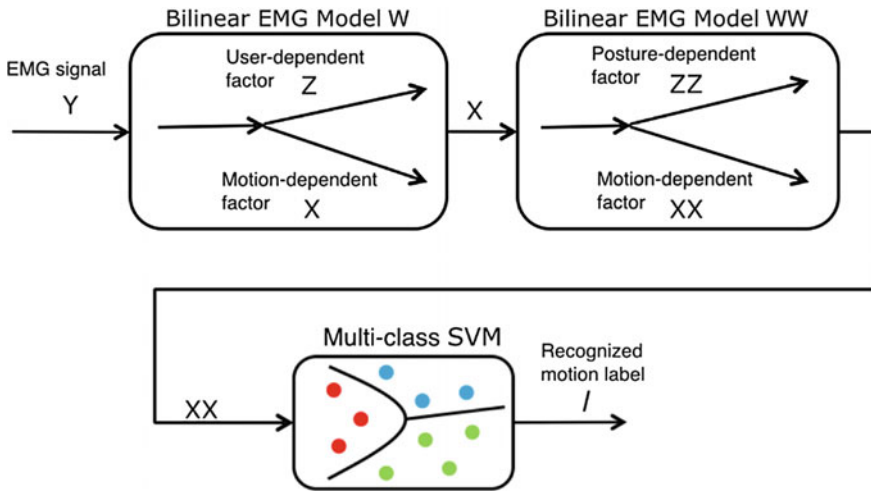


Fig. 1 Concept of a two-step bilinear model

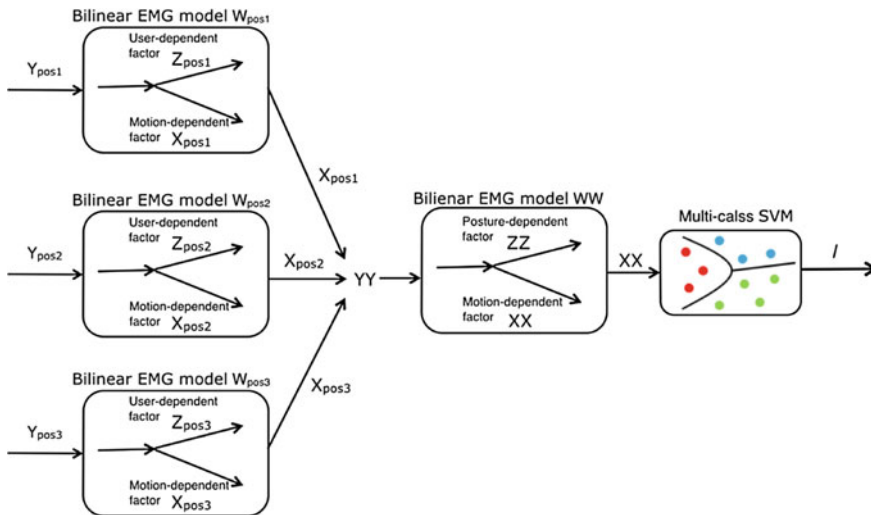


Fig. 2 Proposed method and posture

decompose the EMG signals by using multi-step bilinear model, and finally we can extract truly motion-dependent components that correctly reflect motions.

Based on the consideration, we investigate the influence of arm posture on the classification accuracy. For this aim, we use a two-step bilinear model to remove not only user-dependent but also posture-dependent components from the EMG signals.

As shown in Fig. 1, we first remove user-dependent component from EMG signal by using a first-step bilinear model. We names the motion-dependent component

extracted with the first-step bilinear model as the first motion-dependent factor \mathbf{X} . As the result, we can obtain the first-step bilinear model $\mathbf{W}_{\text{pos1}} \in \mathbf{R}^{I_1 K \times J_1}$ with EMG signal $\mathbf{Y}_{\text{pos1}} \in \mathbf{R}^{K U \times N M}$ in posture 1 (see Fig. 2). We also estimate the parameters \mathbf{W}_{pos2} and \mathbf{W}_{pos3} with respect to the postures 2 and 3. We can extract each first motion-dependent factor $\mathbf{X}_{\text{pos1}} \in \mathbf{R}^{J_1 \times N M}$, \mathbf{X}_{pos2} , \mathbf{X}_{pos3} by using the first-step bilinear model.

Next, we remove posture-dependent component from the first motion dependent factor by using second-step bilinear model. We names the motion-dependent component extracted with the second-step bilinear model as a second motion-dependent factor. Here we define $\mathbf{Y}\mathbf{Y}$ as follows.

$$\mathbf{Y}\mathbf{Y} = \begin{bmatrix} \mathbf{X}_{\text{pos1}} \\ \mathbf{X}_{\text{pos2}} \\ \mathbf{X}_{\text{pos3}} \end{bmatrix} \quad (9)$$

Then we can obtain the second-step bilinear model $\mathbf{W}\mathbf{W} \in \mathbf{R}^{I_2 J_1 \times J_2}$ with $\mathbf{Y}\mathbf{Y}$. Using the second-step bilinear model, we can remove the posture dependent component from the first motion dependent factor, and obtain the second motion dependent factor $\mathbf{X}\mathbf{X}$. After the signal processing, we finally construct appropriate classifier by using multi-class support vector machine (SVM).

In the following experiment, the numbers of I and J were determined according to the singular values in the diagonal matrix $\mathbf{\Sigma}$, where the top S values is effective to account for over 90% of them as follows.

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \sigma_N \end{bmatrix},$$

$$\frac{\sum_{i=1}^S \sigma_i^2}{\sum_{i=1}^N \sigma_i^2} \geq 0.90. \quad (10)$$

3 Experiment

In this study we assumed to classify five hand gestures shown in Fig. 3 under three distinct forearm postures (Fig. 4).

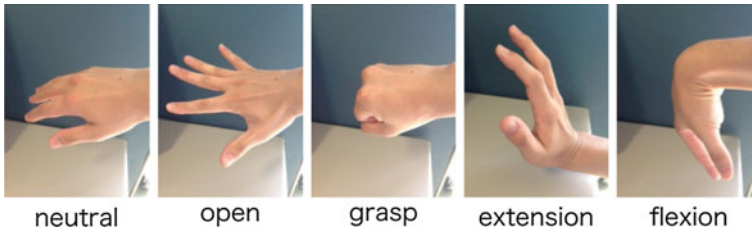


Fig. 3 Hand gestures

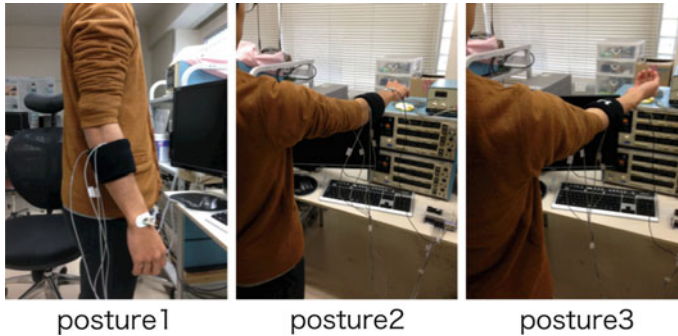


Fig. 4 Forearm postures

Table 1 Placement of electrodes

Ch	Muscle	Function
1	Flexor carpi ulnaris	Hand flexion
2	Extensor carpi radialis	Hand extension
3	Flexor digitorum profundus	Finger flexion
4	Extensor digitorum	Finger extension

3.1 Data Collection

We collected EMG data from ten healthy subjects (nine males and one female) aged 22–23 years. All were right handed. We placed four active myoelectric electrodes on the surface of subject’s forearm. The target muscles are listed in Table 1. The subjects were asked to perform each of the five hand gestures under each posture, in a fixed order: *neutral*, *open*, *neutral*, *grasp*, *neutral*, *extension*, *neutral* and *flexion* (see Fig. 5), and we measured EMG of 5 s for each hand gesture. Here the neutral gesture was introduced as a resting state. Although subjects performed the neutral gesture several times in a task, we only extracted and used the first piece of data for further analysis.

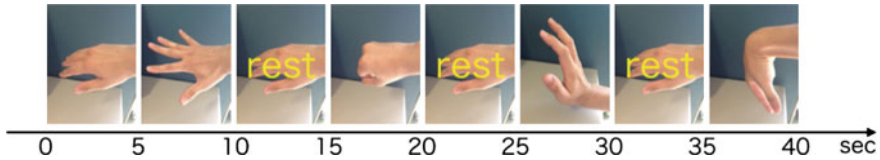


Fig. 5 Experimental task

3.2 *Signal Preprocessing and Feature Extraction*

We measured the four channels of EMG signals at a 1024 Hz sampling rate. After dividing the EMG signals into the epoch of each hand gesture, we removed first 1 s of each epoch because it may include transition state. The EMG signals were band-pass filtered between 10 and 500 Hz, full-wave rectified, and a low-pass filtered with a cutoff frequency of 10 Hz for smoothing. Using a fixed length of time window (128 samples) and time shift (32 samples), we calculated a root mean square feature of EMG (hereafter, RMS). Thus we obtained 125 data samples for each gesture.

3.3 *Analysis*

We analyzed the validity of our proposed method as follows. First, we trained the first-step bilinear model and a classifier (i.e., SVM) to investigate the effect caused by postural change. We compared classification accuracies between (1) the case that the features used for training and test stemmed from an identical posture, and (2) the case where the features were extracted from different postures.

Second, we trained the second-step bilinear model to eliminate posture-dependent factor, and evaluated classification accuracy based on the second motion-dependent factor \mathbf{XX} .

4 **Result**

4.1 *Analysis 1*

According to the analysis by only using first-step bilinear model, we found that mean classification accuracy for different postural data were obviously inferior to that for identical postural data (see Table 2). Thus we confirmed that postural changes have serious degrading effect on classification accuracy. Detail comparisons of the classification accuracy for each hand gesture are specified in Table 2 and Fig. 6.

Table 2 Mean classification accuracy (%) in Analysis 1

Training	Test	Neutral	Open	Grasp	Extension	Flexion
Posture 1	Posture 1	79.46 (25.03)	50.24 (22.43)	59.60 (34.67)	72.64 (29.42)	74.40 (22.72)
Posture 1	Posture 2	60.48 (47.78)	24.32 (33.47)	40.48 (43.62)	71.68 (29.84)	33.36 (42.84)
Posture 1	Posture 3	48.64 (43.49)	31.12 (41.25)	31.68 (42.85)	54.88 (36.23)	31.04 (45.25)
Posture 2	Posture 1	70.40 (37.34)	17.52 (31.50)	61.36 (44.14)	32.32 (23.39)	5.28 (10.67)
Posture 2	Posture 2	85.92 (14.20)	29.12 (25.26)	71.68 (32.70)	44.56 (33.66)	51.12 (30.89)
Posture 2	Posture 3	59.52 (42.23)	11.84 (28.53)	21.28 (34.83)	41.20 (32.01)	4.00 (11.47)
Posture 3	Posture 1	31.12 (43.97)	38.88 (41.38)	18.80 (28.89)	59.04 (35.61)	1.20 (3.60)
Posture 3	Posture 2	50.24 (42.93)	32.16 (39.18)	24.24 (36.65)	47.92 (41.19)	1.76 (4.54)
Posture 3	Posture 3	79.28 (22.13)	41.76 (32.14)	27.20 (33.89)	46.16 (31.75)	46.08 (32.84)
Same posture		81.55 (3.09)	40.37 (8.68)	52.83 (18.78)	54.45 (12.88)	54.16 (12.34)
Different posture		53.40 (12.29)	25.97 (9.18)	32.97 (14.59)	51.17 (12.66)	12.77 (13.82)

The value in parenthesis represents standard error

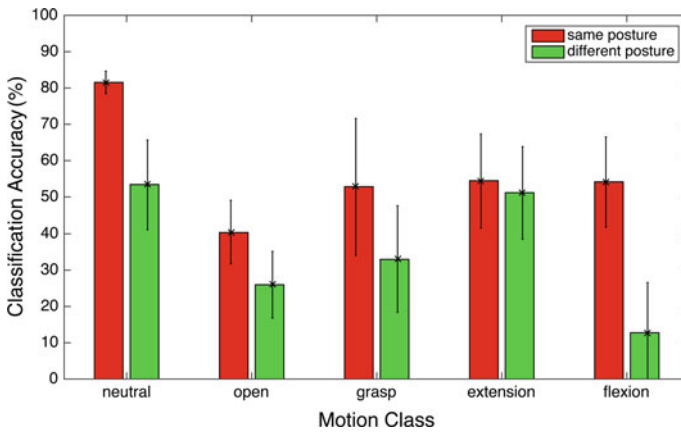


Fig. 6 Classification accuracy (Analysis 1)

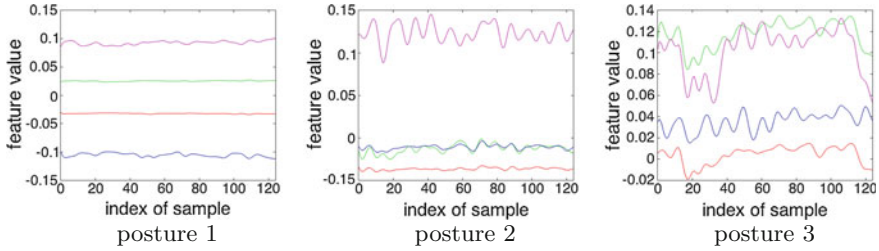


Fig. 7 First motion-dependent factors (*neutral*)

Table 3 Mean classification accuracy (%) in Analysis 2

Leave-one-out	Neutral	Open	Grasp	Extension	Flexion
Posture1	91.68 (10.20)	55.76 (30.30)	54.64 (30.60)	52.96 (21.70)	62.24 (13.70)
Posture2	85.28 (16.83)	50.88 (22.06)	44.00 (25.38)	51.28 (23.75)	50.40 (34.50)
Posture3	91.84 (8.57)	56.32 (24.73)	50.80 (25.46)	58.48 (26.80)	48.64 (25.75)
Average	89.60 (3.06)	54.32 (2.44)	49.81 (4.40)	54.24 (3.08)	53.76 (6.04)

The value in parenthesis represents standard error

In addition, Fig. 7 indicates typical waveforms of first motion-dependent factors in different postures. As shown in the figure, we can see variety in RMS features, even though these correspond to an identical hand gesture (*neutral*).

4.2 Analysis 2

Classification accuracy based on the second motion-dependent factor is shown in Table 3. Moreover, Fig. 8 represents comparison of classification accuracies among the results obtained in Analyses 1 and 2.

From these results, we found that our proposed method is the same or more superior to the case where the features used for training and test were extracted from same postures, even in the case of different postures. It suggest that our proposed method is promising, even though its classification accuracy is still insufficient.

5 Conclusion

In this study, we investigated the effect of postural change on classification accuracy of EMG-based hand gesture recognition. We assumed that EMG signals include posture-dependent factor, and we proposed a posture-dependent components elim-

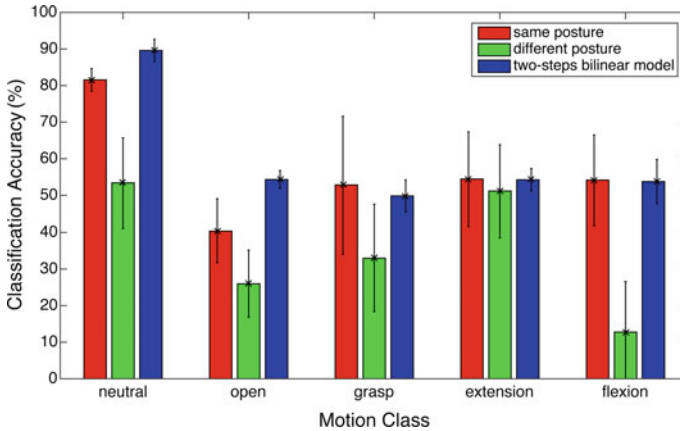


Fig. 8 Comparison of mean classification accuracies between Analyses 1 and 2

ination method using a two-step bilinear model. As a result, we found that EMG signal is much influenced by postural changes, and we showed that EMG signals include posture-dependent component. In addition, we confirmed that the two-step bilinear model can remove posture-dependent component from EMG signals and extract significant feature that is correctly reflecting hand gesture. We confirmed that the proposed method would be effective. However, in this study, classification accuracy was still insufficient except for the neutral gesture. Thus we have to reconsider and improve our experimental conditions, e.g., selection of muscles used for classification, placement of electrodes.

Acknowledgements We sincerely thank the subjects who participated in the experiments as volunteer. The research was partially supported by MEXT fund for Smart Space Technology toward Sustainable Society, and JSPS KAKENHI, (Grant number 26120005, 16H03219).

References

1. Tsukamoto, M., Kondo, T., Ito, K.: A prosthetic hand control based on nonstationary EMG at the beginning of motion. *J. Robot. Mechatron.* **19**(4), 381–387 (2007)
2. Sasaki, Y., Kondo, T.: A proposal of EMG-based teleoperation interface for distance mobility. In: *Proceedings of the 2011 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2011)*, pp. 2904–2909 (2011)
3. OSkoei, M.A., Hu, H.: Control Myoelectric control systems—A survey. *Biomed. Signal Process.* **2**(4), 275–294 (2007)
4. Kataoka, I., Kondo, T., Khoa, Q.D.: A myoelectric control interface compensating for electrode displacements. In: *Proceedings of the third ICT International Student Project Conference (ICT-ISPC)*, pp. 57–60 (2014). doi:[10.1109/ICT-ISPC.2014.6923217](https://doi.org/10.1109/ICT-ISPC.2014.6923217)
5. Hakonen, M., Piitulainen, H., Visala, A.: Current state of digital signal processing in myoelectric interfaces and related applications. *Biomed. Signal Process. Control* **18**, 334–359 (2015). doi:[10.1016/j.bspc.2015.02.009](https://doi.org/10.1016/j.bspc.2015.02.009)

6. Mesin, L., Merletti, R., Rainoldi, A.: Surface EMG: the issue of electrode location. *J. Electromyogr. Kinesiol.* **19**(5), 719–726 (2009)
7. Fougner, A., Scheme, Chan, A.D., Englehart, K., Stavadahl, Ø.: Resolving the limb position effect in myoelectric pattern recognition. *IEEE Trans. Neural Syst. Rehabil. Eng.* **19**(6), 644–651 (2011)
8. Wheeler, K.R., Chang, M.H., Knuth, K.H.: Gesture-based control and EMG decomposition. *Syst. Man Cybern Part C: Appl. Rev. IEEE Trans.* **36**(4), 503–514 (2006)
9. Matsubara, T., Morimoto, J.: Bilinear modeling of emg signals to extract user-independent features for multiuser myoelectric interface. *IEEE Trans. Biomed. Eng.* **60**(8), 2205–2213 (2013)
10. Tenenbaum, J.B., Freeman, W.T.: Separating style and content with bilinear models. *Neural Comput.* **12**(6), 1247–1283 (2000)

Quantification of Temporal Parameters for Tripedalism

Arito Yozu, Dai Owaki, Masashi Hamada, Takuya Sasaki, Qi An, Tetsuro Funato and Nobuhiko Haga

Abstract Bipedalism is one of the distinctive features of humans. However, humans in certain conditions use tripedalism for their locomotion. Patients who cannot bear their weight on their own legs or patients with balance disorders often use a cane. Temporal parameters have been defined for bipedalism, while they have not been defined for tripedalism. Therefore, in clinical rehabilitation, evaluation of patients' gait using a cane is still very much a qualitative issue. In this study, we propose how we can define the temporal parameters for tripedalism. We calculated six quantitative numbers: (1) gait cycle of leg 1, (2) lag of foot strike between leg 1 and leg 2, (3) lag of foot strike between leg 1 and leg 3, (4) stance phase of leg 1, (5) stance phase of leg 2, and (6) stance phase of leg 3. With a set of these six elements, the foot strike and foot-off pattern of the three legs is uniquely determined. Because these elements are measurable quantitatively, we are able to express the tripedal gait in a quantitative manner with this number set. We call this number set "gait matrix". The application of this gait matrix may be useful for evaluating patients' gait using a cane in rehabilitation.

A. Yozu (✉) · N. Haga

Department of Rehabilitation Medicine, The University of Tokyo Hospital,
7-3-1 Hongo, Bunkyo-Ku, Tokyo 113-8655, Japan
e-mail: yodu-jscn@umin.net

D. Owaki

Research Institute of Electrical Communication, Tohoku University, 2-1-1 Katahira,
Aoba-Ku, Sendai 980-8577, Japan

M. Hamada · T. Sasaki

Department of Neurology, Graduate School of Medicine, The University of Tokyo,
7-3-1 Hongo, Bunkyo-Ku, Tokyo 113-8655, Japan

Q. An

Department of Precision Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-Ku,
Tokyo 113-8656, Japan

T. Funato

Graduate School of Informatics and Engineering, The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan

Keywords Tripedalism • Gait • Locomotion • Temporal parameters • Cane • Rehabilitation

1 Introduction

Bipedalism is one of the distinctive features of humans. However, humans in certain conditions use assistive devices for gait. Patients who cannot bear their weight on their own legs or patients with balance disorders often use a cane or crutch. The use of a single cane and one’s own two legs is a kind of tripedalism.

For bipedalism, temporal parameters have been defined since long ago (Fig. 1) [1, 2]. A gait cycle is an interval of time during which one sequence of regularly recurring succession of events is completed. In bipedalism, there are four events during one gait cycle: foot strike, contralateral foot off, contralateral foot strike, and ipsilateral foot off. The gait cycle continues until the second ipsilateral foot strike. There are two phases in one limb: stance and swing. The stance phase is defined by the interval during which the foot is in contact with the ground and the swing phase by the interval during which the foot is in the air. The stance phase is commonly divided into three periods: initial double limb support (foot strike to contralateral foot off), single limb support (contralateral foot off to contralateral foot strike), and

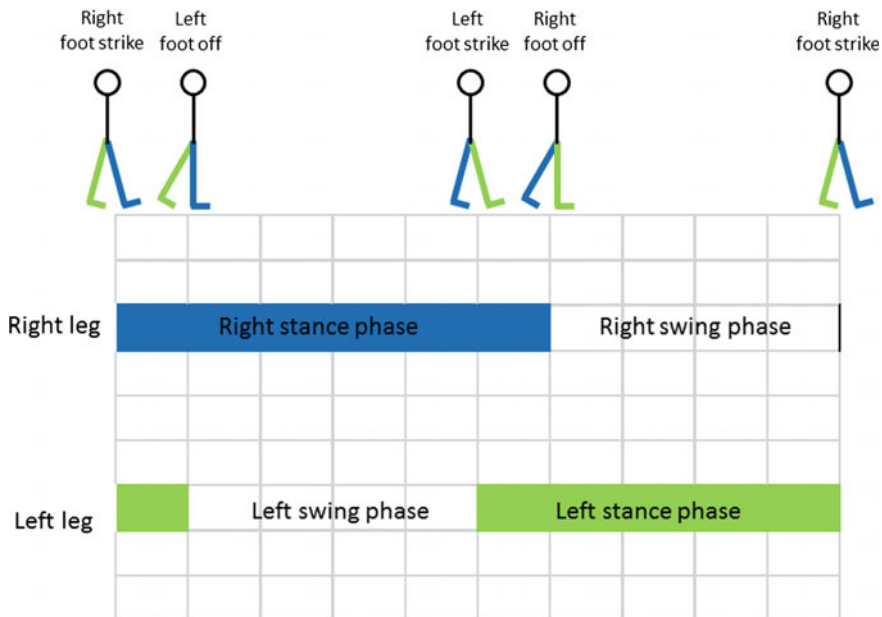


Fig. 1 Temporal parameters for bipedal gait

second double limb support (contralateral foot strike to ipsilateral foot off). These phases and periods can be expressed quantitatively as percentages of the gait cycle.

Unfortunately, quantitative temporal parameters for tripedalism have not been defined. In clinical rehabilitation, teaching the use of a cane is a qualitative issue. For example, “Hold the cane in the opposite hand of the affected leg and move the cane in unison with the affected leg. Each time you step with the affected leg, also move the cane [3].” Commonly, evaluation of patients’ gait with a cane is also qualitative according to visual observation. Some studies have analyzed the gait with a cane quantitatively [4–7], but those studies evaluated only the parameters of the patients’ own legs. No study has analyzed the cane and the patient’s own legs together. If there is a quantitative way to express the temporal parameters for tripedalism, we would be able to teach and evaluate patients more precisely.

In this study, we developed a concept for the definition of the temporal parameters for tripedalism, by which we will be able to quantify the temporal aspects of gait such as single cane gait.

2 Methods

First, the legs are numbered, arbitrarily ordered, as leg 1, leg 2, and leg 3. Second, we describe the stance and swing phase of each leg along the time axis (Fig. 2). Here, we define our gait cycle as foot strike of leg 1 until the next foot strike of leg 1. As long as we deal with a repetitive cyclic gait, duration of the gait cycle will not change even if we choose another leg for leg 1 (Fig. 3).

Third, we define “the lag of foot strike (LFS) for leg 2” as the time lag between the foot strike of leg 1 and that of leg 2 (Fig. 4). We also define “LFS for leg 3” as the time lag between the foot strike of leg 1 and that of leg 3. We can express this by the percentage of the gait cycle.

Fourth, we express the duration of the stance phase of each leg by the percentage of the gait cycle (Fig. 5). This procedure is the same as used for bipedalism.

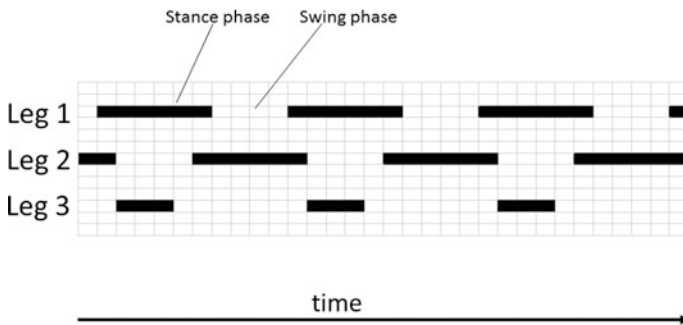


Fig. 2 Example of stance and swing phase for tripedalism

Fig. 3 Gait cycle of Leg 1.
As long as we deal with a repetitive cyclic gait, duration of the gait cycle will not change even if we chose another leg for leg 1

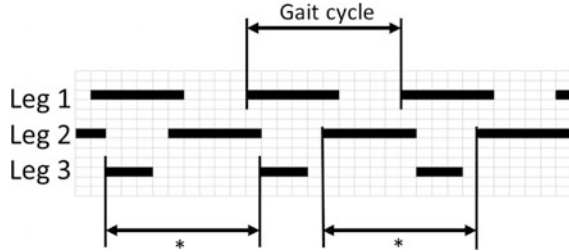


Fig. 4 Lag of foot strike for leg 2 and leg 3

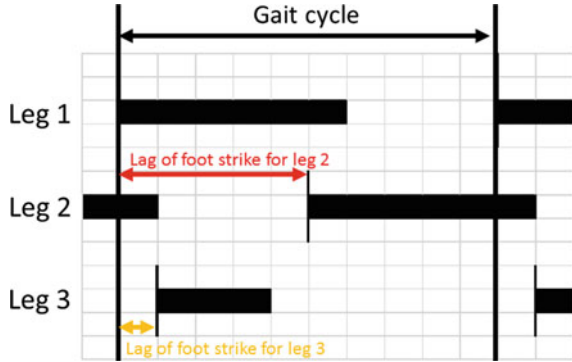
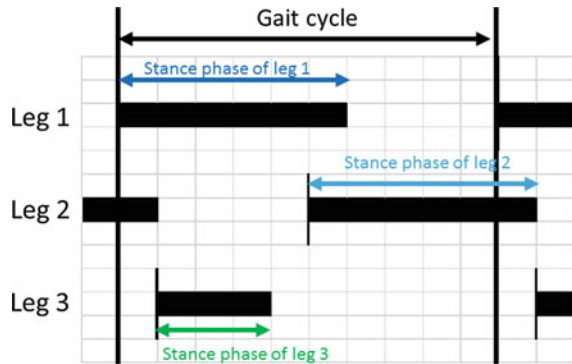


Fig. 5 Stance phase of leg 1, leg 2 and leg 3



According to the above, we now have a set of six numbers to express the tripedal gait:

$$\begin{pmatrix} \text{Gait cycle of leg 1,} & \text{Stance phase of leg 1} \\ \text{LFS of leg 2,} & \text{Stance phase of leg 2} \\ \text{LFS of leg 3,} & \text{Stance phase of leg 3} \end{pmatrix}$$

With this number set, a foot strike and foot-off pattern of the three legs is uniquely determined. Because these numbers are measurable quantitatively, we are

able to express the tripedal gait in a quantitative manner with this number set. We propose to name this number set “a gait matrix”.

In addition, by counting the number of the foot in the stance phase, we can determine the periods of single-limb support, double-limb support, and triple-limb support. This can also be expressed quantitatively.

3 Results

Tripedal gait with foot strike and foot-off pattern as described in Fig. 6 can be expressed as:

$$\begin{pmatrix} 1.2 \text{ s,} & 60 \% \\ 50 \%, & 60 \% \\ 10 \%, & 30 \% \end{pmatrix}$$

This gait matrix is unique to the gait pattern of Fig. 6. The single-, double-, and triple-limb support times are 50 %, 50 %, and 0 % of the gait cycle, respectively.

As another example, Fig. 7 is a kind of gait pattern that is taught to patients by rehabilitation doctors or physical therapists. Gait matrix of this pattern is:

$$\begin{pmatrix} 1.2 \text{ s,} & 60 \% \\ 50 \%, & 60 \% \\ 0 \%, & 60 \% \end{pmatrix}$$

The single-, double-, and triple-limb support times are 40 %, 40 %, and 20 % of the gait cycle, respectively.

Fig. 6 Example of calculating gait matrix

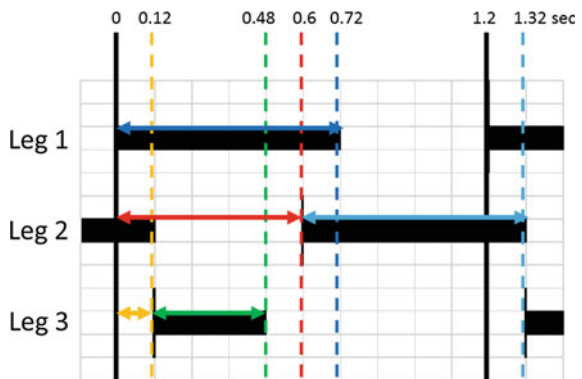
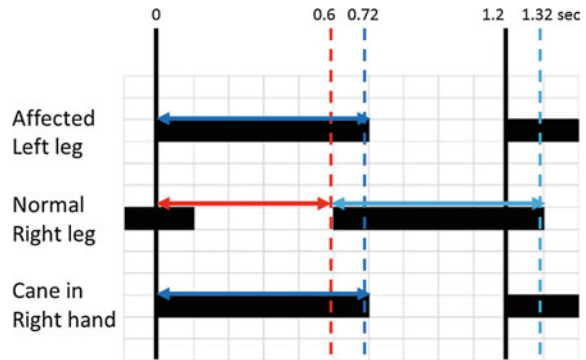


Fig. 7 A kind of gait pattern that is taught to patients in rehabilitation



4 Discussion

We proposed this method to express the temporal parameters of tripodism. With a set of six numbers, we could determine a gait pattern uniquely.

Previously, Hildebrand expressed the symmetrical gait of horses using two parameters: the stance phase of the hindlimb and lag between forefoot foot strike and hindfoot foot strike [8]. He assumed that the stance phase of the fore- and hind-foot are equal. He also assumed that the right and left side are equally reciprocal. Therefore, he only needed two parameters. However, in tripodism, the stance phase may vary between the legs, and right and left may not be symmetrical. This is why we needed six parameters to define the gait.

Acknowledgements This work was supported in part by the Japan Society for the Promotion of Science KAKENHI (# 15K01361).

References

1. Rose, J., Gamble, J.G.: Human Walking. Lippincott Williams & Wilkins, Philadelphia (2006)
2. Perry, J., Burnfield, J.M.: Gait analysis: Normal and Pathological Function. Slack Incorporated, New Jersey (1992)
3. Mayo Foundation for Medical Education and Research: ‘Tips for choosing and using canes’. <http://www.mayoclinic.org/healthy-lifestyle/healthy-aging/multimedia/canes/sls-20077060>
4. Jayakaran, P., DeSouza, L., Cossar, J., Gilhooly, K.: Influence of a walking aid on temporal and spatial parameters of gait in healthy adults. *PM & R: The Journal of Injury, Function, and Rehabilitation* **6**, 796–801 (2014)
5. Tyson, S.F., Rogerson, L.: Assistive walking devices in nonambulant patients undergoing rehabilitation after stroke: the effects on functional mobility, walking impairments, and patients’ opinion. *Arch. Phys. Med. Rehabil.* **90**, 475–479 (2009)
6. Allet, L., Leemann, B., Guyen, E., Murphy, L., Monnin, D., Herrmann, F.R., Schnider, A.: Effect of different walking aids on walking capacity of patients with poststroke hemiparesis. *Arch. Phys. Med. Rehabil.* **90**, 1408–1413 (2009)

7. Ajemian, S., Thon, D., Clare, P., Kaul, L., Zernicke, R.F., Loitz-Ramage, B.: Cane-assisted gait biomechanics and electromyography after total hip arthroplasty. *Arch. Phys. Med. Rehabil.* **85**, 1966–1971 (2004)
8. Hildebrand, M.: Symmetrical gaits of horses. *Science* **150**, 701–708 (1965)

Proposal of a Stance Postural Control Model with Vestibular and Proprioceptive Somatosensory Sensory Input

Ping Jiang, Shouhei Shirafuji, Ryosuke Chiba, Kaoru Takakusaki and Jun Ota

Abstract Maintenance of upright stance is one of the basic requirements in human daily life. Stance postural control is achieved based on multisensory inputs such as visual, vestibular and proprioceptive somatosensory inputs. In this paper, we proposed a stance postural control model including a neural controller with feed-forward inputs (muscle stiffness regulation) and sensory feedback of vestibular and proprioceptive somatosensory sensation. Through the optimization, variables of neural controller were designed to keep a musculoskeletal model standing during a 5 s forward dynamics simulation. From the results, we found that when both vestibular and proprioceptive somatosensory sensory input are available, low muscle stiffness is enough to maintain the balance of a musculoskeletal model in a stance posture. However, when vestibular sensory input get lost, higher muscle stiffness will be desired to keep the musculoskeletal model standing.

Keywords Stance postural control · Musculoskeletal model · Forward dynamics simulation · Multisensory feedback information · Embodied-brain systems science

1 Introduction

Maintenance of upright stance is one of the basic requirements in human daily life. Stance postural control is complicated because center nervous system coordi-

P. Jiang (✉)

Department of Precision Engineering, School of Engineering,
The University of Tokyo, Tokyo, Japan
e-mail: jiang@race.u-tokyo.ac.jp

R. Chiba · K. Takakusaki

Research Center for Brain Function and Medical Engineering,
Asahikawa Medical University, Asahikawa, Japan

S. Shirafuji · J. Ota

Research into Artifacts, Center for Engineering (RACE),
The University of Tokyo, Kashiwa, Japan

© Springer International Publishing AG 2017

W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_4

nates numerous muscles based on multisensory inputs such as visual, vestibular and proprioceptive somatosensory inputs. Loss or degradation of the sensory inputs will cause the motor dysfunction. Our rehabilitation therapy is still far from effective for our limited understandings on human postural control. To make more insights into postural control, modeling postural control mechanism in consideration of sensory inputs and a musculoskeletal model with muscles is significant.

Physiological studies discussed influence of multisensory inputs on the postural stability. Generally, human beings have three sort of sensory inputs: visual, vestibular and somatosensory input. Hwang et al. [1] elucidated that when one of the sensory inputs was inhabited, the weight of others would increase. Peterka [2] conducted postural response experiment on normal and vestibular loss subjects, respectively. When the platform tilt (disturbance) is smaller than 2° (deg.). Somatosensory input played a more important role than vestibular input while the role vestibular input would increase with the increase of disturbance. Vestibular loss subjects generally could not stand when platform tilt was larger than 4° , implying that somatosensory input is important during quiet standing. Chiba et al. [3] proposed that there was an internal model which regulate the whole body stiffness when one or two of the sensory inputs were inhabited. These experimental studies qualitatively evaluated the role of different sensory inputs, but the nature of postural control mechanism still remains to clarify.

Modeling postural control mechanism also advanced our understandings on the nature of postural control. Many researchers only considered somatosensory input. They used joint angle and angular velocity as continuous [4] or intermittent [5] feedback information for a PD controller to keep an inverted pendulum standing. Different from obtaining joint feedback information directly in [4, 5], van der Kooij [6] used Kalman filter and a predictor to estimate the joint angle and angular velocity based on measurement on multisensory inputs. However, human nervous system actuates muscles rather than joint torque. These studies didn't reveal how human coordinates muscles based on multisensory inputs. Jo et al. [7] proposed a cerebellar feedback control model with feedback of joint angle and angular velocity for keeping a inverted pendulum with 9 muscles standing on a transferred platform. The simulated results of muscle activation patterns qualitatively fit the experimental data. Yet they didn't consider visual and vestibular sensory inputs and evaluate the postural stability during quiet standing.

In this research, we aim to propose a stance postural control model in consideration of vestibular and somatosensory sensory input. A musculoskeletal model with muscles instead of inverted pendulum was adopted. In our previous works [8, 9], we only adopted proprioceptive sensory input. In this paper, we further adopted vestibular sensory input. We investigated the contribution of different muscle stiffness under different sensory input conditions: vestibular input inclusion and exclusion during quiet standing.

2 Problem Statement

Specifically, the problem to be addressed in this paper is to keep a musculoskeletal model with 70 muscles standing when neurological time delay is 120 ms. The neural controller actuate muscles based on the feedback information of muscle length and lengthening velocity as well as body center of mass (CoM) displacement and velocity in the horizontal direction.

2.1 Musculoskeletal Model

To approximate human complicated musculoskeletal system, a musculoskeletal model with muscles is indispensable. There are many musculoskeletal models with precise anatomical data available in OpenSim (SimTK.org), an open source biological simulator. We used Gait2392 model [10] and ToyLanding model [11] in OpenSim 3.3 (SimTK.org). These two models were used in running and jumping researches to reproduce the complicated motion so that the number of muscles and joints are supposed to be sufficient to represent quiet standing.

The specific structure was shown in Fig. 1, the musculoskeletal model was composed of 8 segments, 7 joints and 70 muscular-tendon actuators. Each joint has only one rotation DoF for flexion since we only considered motion in the sagittal plane. In addition, instead of adding torques to all joints directly in previous studies, we utilized 70 Millard2012 [12] muscular-tendon actuators generating linear tension force to actuate the model.

Moreover, movable arrangement of each joint has been defined and realized by adding a coordinate limit force, force added when joint angle displacement exceeds joint limits, to the model.

2.2 Neurological Time Delay

As shown in Fig. 2, generally there are three types of neurological time delay during postural control, which are: feedback (τ_{fb}), transmission (τ_{trans}) and activation

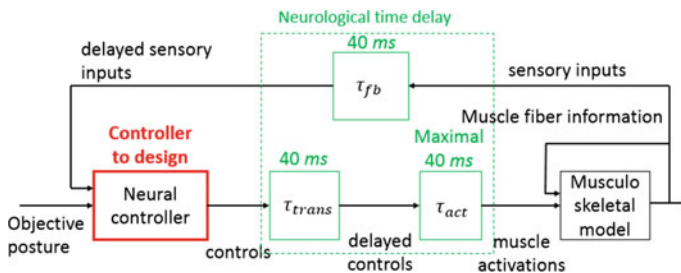


Fig. 1 Problem statement

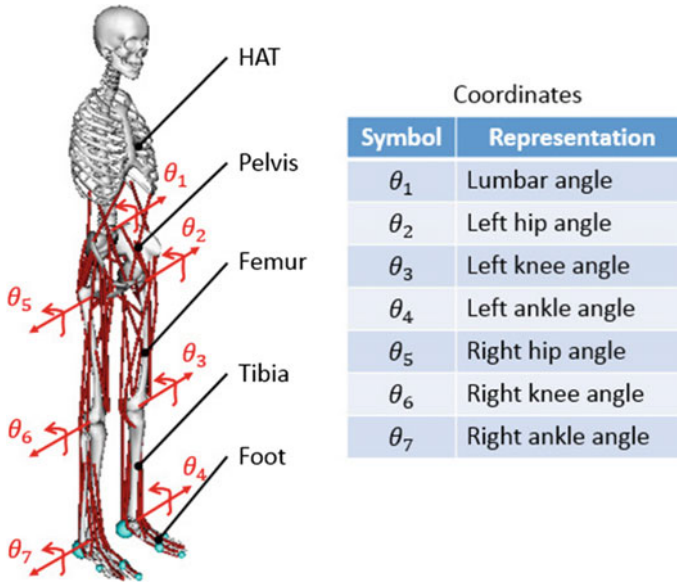


Fig. 2 Problem statement. HAT: head, arm and trunk

dynamics (τ_{act}) neurological time delay. The total of them is 120 ms, including 40 ms τ_{fb} , 40 ms τ_{trans} and a maximum 40 ms τ_{act} [13]. Feedback and transmission neurological time delay are pure 40 ms time delay while activation dynamics delay included activation time delay (t_{act}) and deactivation time delay (t_{deact}) as shown in Eq. 2. t_{act} and t_{deact} were set to 20 and 40 ms according to previous studies [14–16].

$$\dot{a}(t) = \frac{u(t - \tau_{trans}) - a(t)}{D(a(t), u(t - \tau_{trans}))} \quad (1)$$

$$D(a(t), u(t - \tau_{trans})) = \begin{cases} t_{act}(0.5 + 1.5a(t)) & u(t - \tau_{trans}) > a(t) \\ t_{deact}/(0.5 + 1.5a(t)) & u(t - \tau_{trans}) \leq a(t) \end{cases} \quad (2)$$

where u is the total output of the neural controller; a is muscle activation; t_{act} and t_{deact} are the activation and deactivation dynamics time delay, respectively.

2.3 Sensory Inputs

Generally, human beings have three kind of sensory inputs: visual, vestibular and somatosensory sensory inputs. In experimental studies, visual sensory input was inhibited by eye occlusion and without occlusion, or adjusting the tilt angle of a rotational screen relative to human body orientation. This kind of inhabitation is difficult

to model mathematically. To make problem simpler, we only considered vestibular and somatosensory sensory input in this paper. Somatosensory sensory input includes proprioceptive and tactile somatosensory input. The tactile somatosensory input mainly depends on the material of the ground or light touch from environment. We only considered proprioceptive sensory input in this paper for the assumption that musculoskeletal model stands on a normal ground without any external touch.

2.3.1 Vestibular Sensory Input

The vestibular receptors are located in human head. Canal and otolith transducer signals are fused in the brain, yielding a vestibular sensor that provides 3D information of head angular velocity, attitude, and translational acceleration in space [17]. Lockhart et al. [18] showed delayed kinematics of the horizontal CoM could be used in a simple feedback law to generated model muscle-activation patterns for standing on a transferred platform. Paterka showed the role vestibular sensory input will increase with the increase of external disturbance. Therefore, we assumed that the vestibular sensory input, as feedback information to muscles, is CoM position and velocity in horizontal direction.

2.3.2 Proprioceptive Sensory Input

Proprioceptive Sensory receptor in this paper is muscle receptors called muscle spindles. Muscle spindle measures the muscle length during the motion task. In this paper, proprioceptive sensory input was modeled as muscle length and lengthening velocity.

3 Proposed Neural Controller

Our proposed neural controller consists of both feed-forward and feedback control, as shown in Fig. 3. Feedback control includes proprioceptive and vestibular feedback control.

3.1 Feed-Forward Control

Feed-forward (FF) control is a set of activations necessary to keep standing in a stable posture as shown in Eq. 3. For standing by the similar posture, human might have different muscle activation patterns. FF could be regarded as regulation of human body stiffness.

$$u_{ff,i} = c_i \quad (3)$$

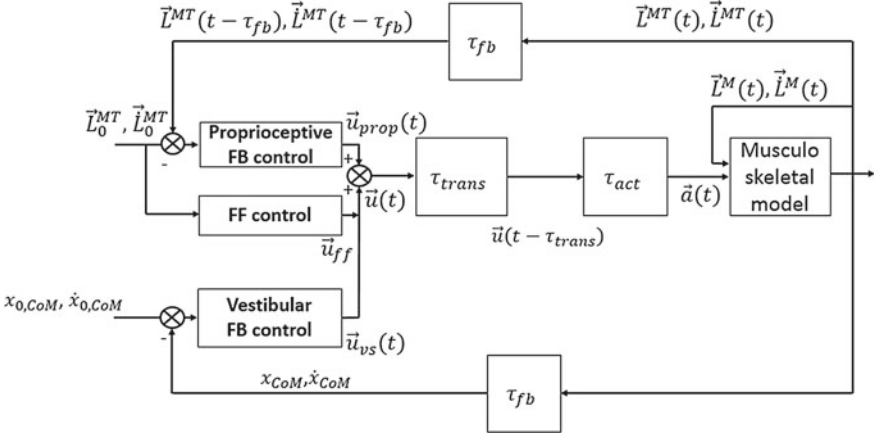


Fig. 3 Proposed neural controller. \vec{L}^{MT} , \vec{L}^M , $\vec{\dot{L}}^{MT}$, $\vec{\dot{L}}^M$ are muscular-tendon and muscle length and lengthening velocity, respectively. x_{CoM} and \dot{x}_{CoM} are horizontal displacement and velocity of center of mass. \vec{a} is muscle activation. \vec{L}_0^{MT} and $\vec{\dot{L}}_0^{MT}$ are objective muscular-tendon length and velocity. $x_{0,CoM}$ and $\dot{x}_{0,CoM}$ are objective horizontal displacement and velocity of center of mass. \vec{u} , \vec{u}_{ff} , \vec{u}_{prop} and \vec{u}_{vs} are total, feed-forward (FF), proprioceptive (PROP) and vestibular (VS) control outputs, respectively

where c_i is a constant activation for the i th muscle to keep musculoskeletal model standing by the objective posture.

3.2 Proprioceptive Feedback Control

Proprioceptive feedback (FB) control is a simple PD control based on the feedback information of muscular-tendon length and lengthening velocity, as shown in Eq. 4. In this paper, nine types of pairs of feedback gains were considered as shown in Eq. 5. Feedback gain for each muscle depends on the joint its path crosses. e.g. muscles whose paths crossing the lumbar joint and have the function to flex the lumbar will belong to lumbar flexor and have feedback gain $[k_{p_l_fl}, k_{d_l_fl}]$.

$$u_{prop,i}(t) = k_{prop_p,i} \left(\frac{L_i^{MT}(t) - L_{0,i}^{MT}}{L_{0,i}^{MT}} \right) + k_{prop_d,i} \left(\frac{\dot{L}_i^{MT}(t) - \dot{L}_{0,i}^{MT}}{V^{max}} \right) \quad (4)$$

where V^{max} is max muscle lengthening velocity. k_{prop_p} and k_{prop_d} are proprioceptive feedback gains.

$$[k_{prop_p,i}, k_{prop_d,i}] = \begin{cases} [k_{p_l_ex}, k_{d_l_ex}] & \text{lumbar extensor} \\ [k_{p_h_ex}, k_{d_h_ex}] & \text{hip extensor} \\ [k_{p_k_ex}, k_{d_k_ex}] & \text{knee extensor} \\ [k_{p_a_ex}, k_{d_a_ex}] & \text{ankle extensor} \\ [k_{p_l_fl}, k_{d_l_fl}] & \text{lumbar flexor} \\ [k_{p_h_fl}, k_{d_h_fl}] & \text{hip flexor} \\ [k_{p_k_fl}, k_{d_k_fl}] & \text{knee flexor} \\ [k_{p_a_fl}, k_{d_a_fl}] & \text{ankle flexor} \\ [k_{p_bi}, k_{d_bi}] & \text{biarticular muscle} \end{cases} \quad (5)$$

where $[k_{prop_p,i}, k_{prop_d,i}]$ is feedback gain of i th muscle.

3.3 Vestibular Feedback Control

Vestibular feedback control is also a PD control based on the feedback information of CoM horizontal position and velocity, as shown in Fig. 4. We divided all the muscles into two groups: muscles moving CoM forward (MCF) and muscles moving CoM backward (MCB). MCF was activated only when current CoM position is behind objective one. On the contrary, MCB was activated only when current CoM position is ahead of objective one. Muscles in each group was further divided into several groups based on their function and location as shown in Eqs. 6–10. Similarly to proprioceptive feedback control, muscles in the same group have the same gains.

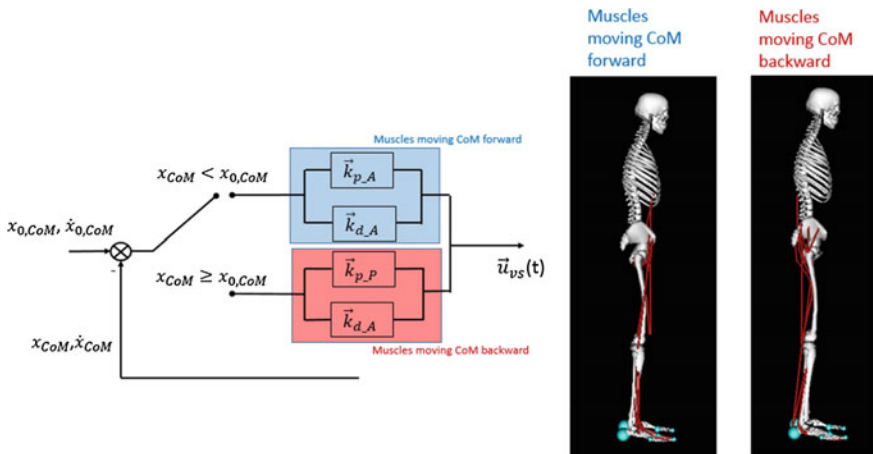


Fig. 4 Vestibular feedback control

$$\vec{u}_{vs}(t) = \vec{k}_{vs_p} x_{CoM} + \vec{k}_{vs_d} \dot{x}_{CoM} \quad (6)$$

$$[\vec{k}_{vs_p}, \vec{k}_{vs_d}] = \begin{cases} [\vec{k}_{p_A}, \vec{k}_{d_A}] & \text{MCF and } x_{CoM} < x_{0,CoM} \\ [0, 0] & \text{else} \end{cases} \quad (7)$$

$$[\vec{k}_{vs_p}, \vec{k}_{vs_d}] = \begin{cases} [\vec{k}_{p_P}, \vec{k}_{d_P}] & \text{MCB and } x_{CoM} > x_{0,CoM} \\ [0, 0] & \text{else} \end{cases} \quad (8)$$

$$[k_{p_A,i}, k_{d_A,i}] = \begin{cases} [k_{p_l_fl}^A, k_{d_l_fl}^A] & \text{lumbar flexor} \\ [k_{p_h_fl}^A, k_{d_h_fl}^A] & \text{hip flexor} \\ [k_{p_k_ex}^A, k_{d_k_ex}^A] & \text{knee extensor} \\ [k_{p_a_fl}^A, k_{d_a_fl}^A] & \text{ankle flexor} \\ [k_{p_bih_fl}^A, k_{d_bih_fl}^A] & \text{biarticular hip flexor} \end{cases} \quad (9)$$

where $[k_{p_A,i}, k_{d_A,i}]$ is feedback gain of i th MCF muscle.

$$[k_{p_P,j}, k_{d_P,j}] = \begin{cases} [k_{p_l_ex}^P, k_{d_l_ex}^P] & \text{lumbar extensor} \\ [k_{p_h_ex}^P, k_{d_h_ex}^P] & \text{hip extensor} \\ [k_{p_k_fl}^P, k_{d_k_fl}^P] & \text{knee flexor} \\ [k_{p_a_ex}^P, k_{d_a_ex}^P] & \text{ankle extensor} \\ [k_{p_bih_ex}^P, k_{d_bih_ex}^P] & \text{biarticular hip extensor} \\ [k_{p_bia_ex}^P, k_{d_bia_ex}^P] & \text{biarticular ankle extensor} \end{cases} \quad (10)$$

where $[k_{p_P,j}, k_{d_P,j}]$ is feedback gain of j th MCB muscle.

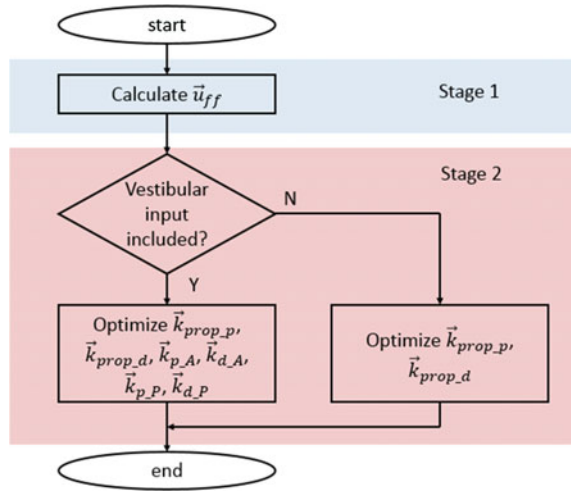
4 Parameter Design

There are 110 parameters necessary to be designed, including 70 \vec{u}_{ff} , 18 $[\vec{k}_{prop_p}, \vec{k}_{prop_d}]$, 10 $[\vec{k}_{p_A}, \vec{k}_{d_A}]$ and 12 $[\vec{k}_{p_P}, \vec{k}_{d_P}]$. It is very difficult to design all the parameters at the same time so that we designed parameters by two stages as shown in Fig. 5.

In the first stage, \vec{u}_{ff} was calculated by only searching the gains of proprioceptive feedback control able to keep musculoskeletal model standing for 5 s when neurological time delay is 0 ms. Please refer to our previous work [8] for the details of calculation.

In the second stage, based on calculated \vec{u}_{ff} in the first stage, we used Covariance Matrix Adaptation Evolution Strategy (CMAES) optimizer to optimize the feedback

Fig. 5 Parameter design



gains under two sensory input conditions: vestibular inclusion and exclusion, respectively. For vestibular exclusion condition, all vestibular feedback gains are 0. Details of optimization and objective function were described in our previous work [9].

5 Results

We defined muscle stiffness by calculating the norm of \vec{u}_{ff} . Among 402 calculated $||\vec{u}_{ff}||$, we selected three $||\vec{u}_{ff}||$ ($||\vec{u}_{ff}|| = 0.11, 0.15, 0.89$) to conduct optimization under the vestibular input inclusion and exclusion condition, respectively. We empirically selected these three $||\vec{u}_{ff}||$ based on assumption that $||\vec{u}_{ff}|| = 0.11$ and 0.15 represented low stiffness while $||\vec{u}_{ff}|| = 0.89$ represented rather high stiffness.

Table 1 Optimized gains for vestibular input excluded

$ \vec{u}_{ff} $	$k_{p_l_ex}$	$k_{d_l_ex}$	$k_{p_h_ex}$	$k_{d_h_ex}$	$k_{p_k_ex}$	$k_{d_k_ex}$	$k_{p_a_ex}$	$k_{d_a_ex}$	$k_{p_l_fl}$
0.11	0.004	0.005	0.065	0.011	0.023	0.052	0.011	0.057	0.029
0.15	0.031	0.034	0.03	0.001	0.003	0.001	0.007	0.007	0.032
0.89	0.049	0.057	0.085	0.029	0.182	0.05	0.059	0.05	0.147
$ \vec{u}_{ff} $	$k_{d_l_fl}$	$k_{p_h_fl}$	$k_{d_h_fl}$	$k_{p_k_fl}$	$k_{d_k_fl}$	$k_{p_a_fl}$	$k_{d_a_fl}$	k_{p_bi}	k_{d_bi}
0.11	0.008	0.022	0.001	0.002	0.003	0.023	0.003	0.014	0.023
0.15	0.021	0.027	0.01	0.053	0.035	0.023	0.015	0.053	0.006
0.89	0.002	0.008	0.009	0.082	0.114	0.122	0.071	0.108	0.007

Table 2 Optimized gains for vestibular input included

$\ \vec{u}_{ff}\ $	$k_{p,l,ex}$	$k_{d,l,ex}$	$k_{p,h,ex}$	$k_{d,h,ex}$	$k_{p,k,ex}$	$k_{d,k,ex}$	$k_{p,a,ex}$	$k_{d,a,ex}$	$k_{p,l,fl}$		
0.11	0.024	0.012	0.002	0.016	0.037	0.034	0.033	0.012	0.018		
0.15	0.033	0.001	0.031	0.024	0.032	0.005	0.019	0.015	0.001		
0.89	0.052	0.033	0.038	0.023	0.026	0.034	0.023	0.014	0.034		
$\ \vec{u}_{ff}\ $	$k_{d,l,fl}$	$k_{p,h,fl}$	$k_{d,h,fl}$	$k_{p,k,fl}$	$k_{d,k,fl}$	$k_{p,a,fl}$	$k_{d,a,fl}$	$k_{p,bi}$	$k_{d,bi}$		
0.11	0.002	0.031	0.006	0.006	0.01	0.02	0.009	0.037	0.002		
0.15	0.023	0.003	0.001	0.022	0.011	0.007	0.012	0.019	0.028		
0.89	0.042	0.01	0.007	0.049	0.009	0.034	0.04	0.012	0.018		
$\ \vec{u}_{ff}\ $	$k_{p,l,fl}^A$	$k_{d,l,fl}^A$	$k_{p,h,fl}^A$	$k_{d,h,fl}^A$	$k_{p,k,ex}^A$	$k_{d,k,ex}^A$	$k_{p,a,fl}^A$	$k_{d,a,fl}^A$	$k_{p,bi}^A$	$k_{d,bi}^A$	$k_{d,bi}^A$
0.11	0.002	0.013	0.015	0.024	0.017	0.027	0.004	0.027	0.007	0.005	
0.15	0.015	0.024	0.007	0.003	0.013	0.016	0.025	0.013	0.018	0.004	
0.89	0.015	0.013	0.01	0.02	0.01	0.031	0.017	0.034	0.007	0.016	
$\ \vec{u}_{ff}\ $	$k_{p,l,ex}^P$	$k_{d,l,ex}^P$	$k_{p,h,ex}^P$	$k_{d,h,ex}^P$	$k_{p,k,fl}^P$	$k_{d,k,fl}^P$	$k_{p,a,ex}^P$	$k_{d,a,ex}^P$	$k_{p,bi}^P$	$k_{d,bi}^P$	$k_{d,bi}^P$
0.11	0.003	0.009	0.016	0.008	0.014	0.01	0.016	0.02	0.006	0.003	0.003
0.15	0.019	0.018	0.021	0.002	0.005	0.001	0.017	0.018	0.013	0.006	0.001
0.89	0.013	0.007	0.003	0.007	0.003	0.006	0.026	0.011	0.013	0.004	0.004

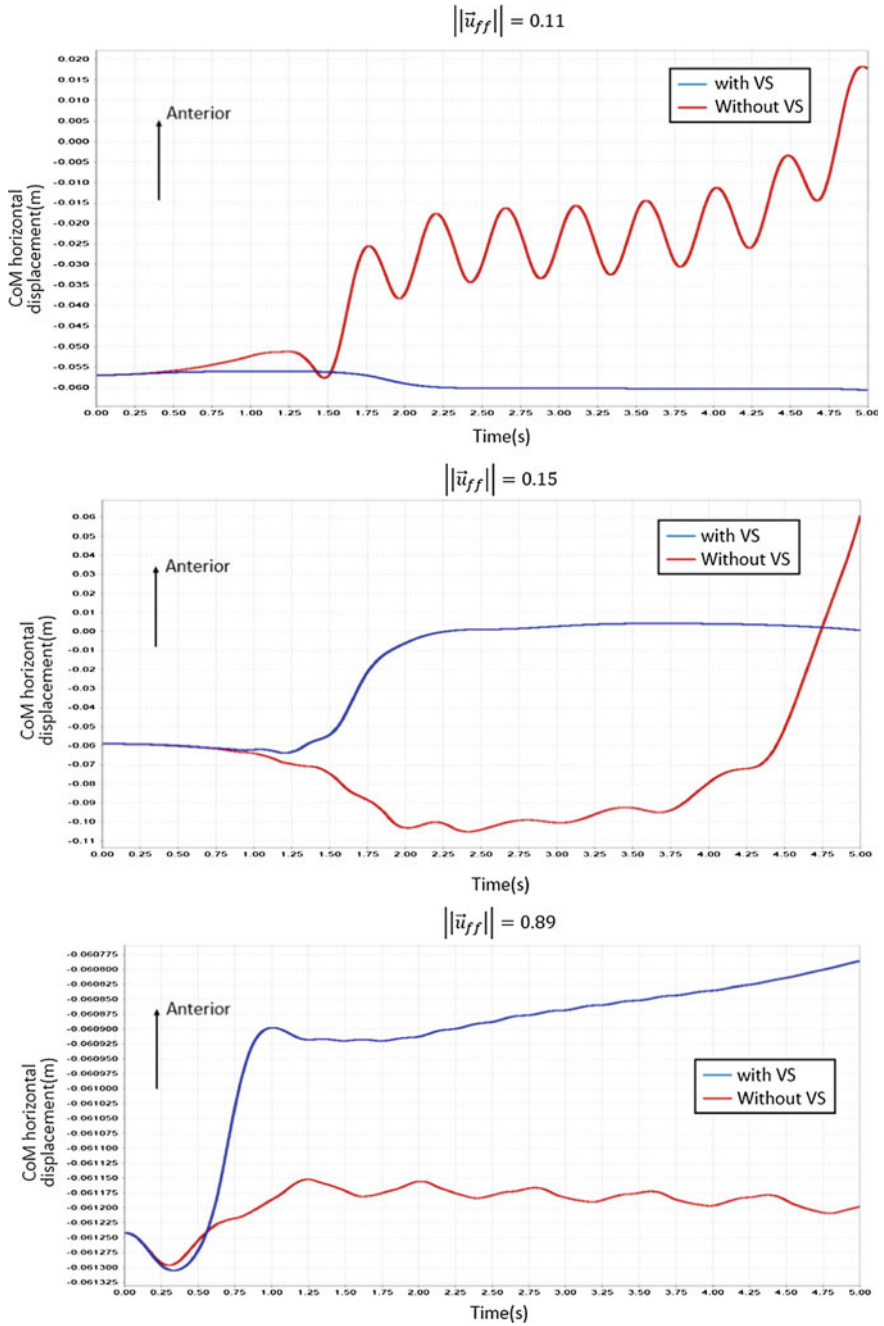


Fig. 6 CoM horizontal displacement. Versus: vestibular input

Table 1 showed the optimized gains when vestibular input was excluded. Table 2 showed the optimized gains when vestibular input was included.

Figure 6 showed the displacement of CoM in the horizontal direction.

6 Discussion

In this research, we aim to propose a stance postural control model in consideration of vestibular and somatosensory sensory input to keep a musculoskeletal model standing when neurological time delay was 120 ms. From the results, we found that the optimizer could find the solution of gains for both sensory input conditions (Vestibular inclusion and exclusion) to keep a musculoskeletal standing for 5 s. However, from the plot of CoM horizontal displacement, we found that when the vestibular sensory input was included, musculoskeletal model could stand stably by a low muscle stiffness ($\|\vec{u}_{ff}\|$ was 0.11 and 0.15). However, when vestibular input got excluded, rather high muscle stiffness ($\|\vec{u}_{ff}\| = 0.89$) was desired. This result is consistent with physiology knowledge that individuals may stiffen their lower legs during upright standing to cope with the deterioration in sensory input.

7 Conclusion

We proposed a stance postural control model in consideration of vestibular and somatosensory sensory input. We optimized parameters when vestibular input was included and excluded respectively to keep musculoskeletal model standing. We found that muscle stiffness might contribution to the compensation of sensory input loss.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number 26120004 and 26120006.

References

1. Hwang, S., Agada, P., Kiemel, T., Jeka, J.J.: Dynamic reweighting of three modalities for sensor fusion. *PloS one* **9**(1) (2014)
2. Peterka, R.: Sensorimotor integration in human postural control. *J. Neurophys.* **88**(3), 1097–1118 (2002)
3. Chiba, R., Ogawa, H., Takakusaki, K., Asama, H., Ota, J.: Muscle activities changing model by difference in sensory inputs on human posture control. In: *Intelligent Autonomous Systems 12*, pp. 479–491. Springer (2013)
4. Masani, K., Vette, A.H., Popovic, M.R.: Controlling balance during quiet standing: proportional and derivative controller generates preceding motor command to body sway position observed in experiments. *Gait & posture* **23**(2), 164–172 (2006)

5. Asai, Y., Tasaka, Y., Nomura, K., Nomura, T., Casadio, M., Morasso, P.: A model of postural control in quiet standing: robust compensation of delay-induced instability using intermittent activation of feedback control. *PLoS One* **4**(7), e6169–e6169 (2009)
6. van der Kooij, H., Jacobs, R., Koopman, B., Grootenboer, H.: A multisensory integration model of human stance control. *Biol. Cybern.* **80**(5), 299–308 (1999)
7. Jo, S., Massaquoi, S.G.: A model of cerebellum stabilized and scheduled hybrid long-loop control of upright balance. *Biol. Cybern.* **91**(3), 188–202 (2004)
8. Jiang, P., Chiba, R., Takakusaki, K., Ota, J.: Stance postural control of a musculoskeletal model able to compensate neurological time delay. In: 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1130–1135, Dec 2014
9. Jiang, P., Chiba, R., Takakusaki, K., Ota, J.: Generation of biped stance motion in consideration of neurological time delay through forward dynamics simulation. In: 2015 International Symposium on Micro-NanoMechatronics and Human Science, pp. 205–208, Nov 2015
10. Au, C.: Gait 2392 and 2354 models (2013)
11. Hicks, J.: Simulation-based design to prevent ankle injuries (2014)
12. Millard, M., Uchida, T., Seth, A., Delp, S.L.: Flexing computational muscle: modeling and simulation of musculotendon dynamics. *J. Biomech. Eng.* **135**(2), 021005 (2013)
13. Masani, K., Popovic, M.R., Nakazawa, K., Kouzaki, M., Nozaki, D.: Importance of body sway velocity information in controlling ankle extensor activities during quiet stance. *J. Neurophys.* **90**(6), 3774–3782 (2003)
14. Dorn, T.W., Wang, J.M., Hicks, J.L., Delp, S.L.: Predictive simulation generates human adaptations during loaded and inclined walking. *PLoS ONE* **10**(4), e0121407, Apr 2015
15. Winters, J.M.: An improved muscle-reflex actuator for use in large-scale neuromusculoskeletal models. *Ann. Biomed. Eng.* **23**(4), 359–374 (1995)
16. Jacobs, D.A.: First-order activation dynamics (2015)
17. Mergner, T.: A neurological view on reactive human stance control. *Ann. Rev. Control* **34**(2), 177–198 (2010)
18. Lockhart, D.B., Ting, L.H.: Optimal sensorimotor transformations for balance. *Nature Neurosci.* **10**(10), 1329–1336 (2007)

Simultaneous Localization, Mapping and Self-body Shape Estimation by a Mobile Robot

Akira Taniguchi, Lv WanPeng, Tadahiro Taniguchi, Toshiaki Takano, Yoshinobu Hagiwara and Shiro Yano

Abstract This paper describes a new method for estimating the body shape of a mobile robot by using sensory-motor information. In many biological systems, it is important to be able to estimate body shapes to allow it to appropriately behave in a complex environment. Humans and other animals can form their body image and determine actions based on their recognized body shape. However, conventional mobile robots have not had the ability to estimate body shape, and instead, developers have provided body shape information to the robots. In this paper, we describe a new method that enables a robot to obtain only subjective information, e.g., motor commands and distance sensor information, and automatically estimate its self-body shape. We call the method simultaneous localization, mapping, and self-body shape estimation (SLAM-SBE). The method is based on Bayesian statistics. In particular, the method is obtained by extending the simultaneous localization and mapping (SLAM) method. Experimental results show that a mobile robot can obtain a self-body shape image represented by an occupancy grid by using only its sensory-motor information (i.e., without any objective measurement of its body).

A. Taniguchi (✉) · L. WanPeng · T. Taniguchi (✉) · T. Takano (✉) · Y. Hagiwara (✉)
Ritsumeikan University, 1-1-1 Noji Higashi, Kusatsu, Shiga 525-8577, Japan
e-mail: a.taniguchi@em.ci.ritsumeik.ac.jp

T. Taniguchi
e-mail: taniguchi@em.ci.ritsumeik.ac.jp

T. Takano
e-mail: takano@em.ci.ritsumeik.ac.jp

Y. Hagiwara
e-mail: yhagiwara@em.ci.ritsumeik.ac.jp

L. WanPeng
e-mail: ro@em.ci.ritsumeik.ac.jp

S. Yano (✉)
Tokyo University of Agriculture and Technology,
2-24-16 Nakamachi, Koganei, Tokyo 184-8588, Japan
e-mail: syano@cc.tuat.ac.jp

Keywords Self-body shape estimation • SLAM • Bayesian filter • Sensory-motor information

1 Introduction

Animals, especially higher mammals such as human beings, must adapt to their environment and control their bodies accordingly to be able to survive. To control their bodies and to behave appropriately in environments, recognizing objects and situations, i.e., things outside biological systems, and recognizing their own bodies are important because real-world behavior emerges from triadic interactions between the brain, body, and environment (see Fig. 1).

When an animal or a robot attempts to recognize its own body, the problem is that the cognition of the animal or robot is self-enclosed. Therefore, an animal has no knowledge regarding its environment or body before interactions. An animal must gain knowledge regarding its environment and body through triadic interactions. To cite a very simple case, if a person does not know the size of his/her body, he/she cannot determine whether he/she could pass through an aperture.

There are many studies regarding self-awareness of animals' bodies [1–5]. Self-awareness of an animal's body includes several elements, i.e., body schema, body

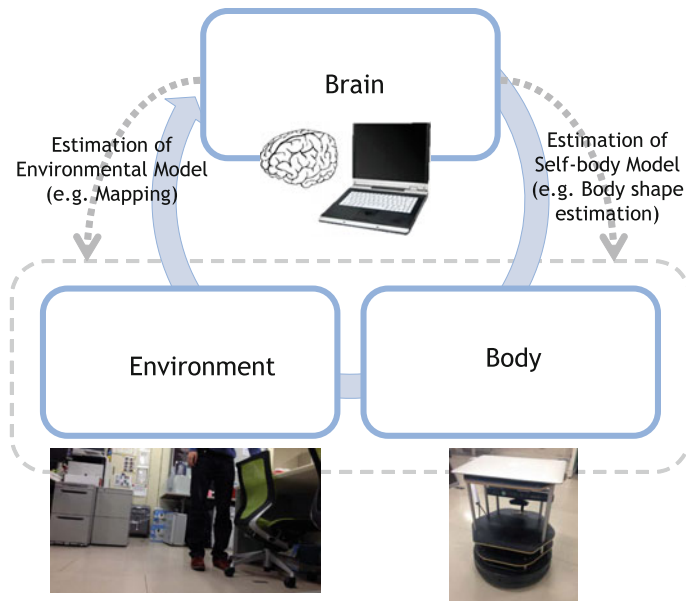


Fig. 1 Diagram of triadic interaction between the brain, body, and environment. The brain must construct an environmental model as well as a self-body model

image, body map, sense of self-ownership, sense of self-agency, and body shape. In this paper, we focus on body shape, which relates to body image.

As we grow, the shapes of our body gradually change. Therefore, our self-body image must adapt to a changing body. Additionally, when a human uses a tool, he/she extends his/her body schema and assimilates the tool with the body image represented as a part of his/her own body in his/her brain. This process is a very important brain function for our daily life. For example, as a human learns how to drive a car, he/she develops a sense regarding the body of the car. The driver must know the width and length of the vehicle during driving practice. Iriki et al. showed that the representation of a macaque's body schema was also extended in its brain when it used a tool [1, 2]. These cases are examples of fast adaptation of body schemas.

Not only the usual adaptation of body schema but also unusual phenomena, such as phantom limb and rubber hand illusion, are gaining the attention of researchers [3]. It is known that people who are suffering from phantom limb after losing a part of their body experience a disjunct between sensation and structure.

Not only mammals, but also other animals, can dynamically adapt to their changing bodies. For example, Sonoda et al. showed that hermit crabs, *Coenobita rugosus*, which change their shells frequently, can modify their behavior based on the shape of their shells [4, 5].

A biological system estimating its body shape without any objective information, e.g., a visual image from outside, or measuring the length of its body is not a trivial process. A biological system must estimate its self-body shape by using only subjective sensory-motor information, e.g., a visual image obtained from its eyes or action commands output to its muscle system, as its cognitive system is self-enclosed and cannot obtain information observed from outside its body. To provide a computational and robotic model, in which a robot can automatically estimate its body shape from subjective sensory-motor information alone, is important for further understanding of animals' self-body shape estimation process. Forming knowledge regarding self-body shape is also important for mobile robots because a robot requires body-shape information when it plans its trajectory of movement.

There are several previous studies regarding self-body estimation by a robot. Nabeshima et al. provided a robotic model that reproduces the process that a manipulated object, i.e., a tool, can become incorporated into the body schema [6]. The study treated the modification of body schema in tool-use that was observed in Iriki et al.'s experiment [1]. Olsson et al. proposed a method for reconstructing the layout of sensors and actuators by using only an information flow obtained from unknown sensors and actuators [7]. Sugiura et al. solved a similar problem by using the time difference of arrival of signals and multidimensional scaling [8]. Yoshikawa et al. proposed a cross-modal map learning method for body schema acquisition [9]. Mori and Kuniyoshi provided a constructive model of a human fetus that forms behaviors and understanding regarding its body through general movements in the womb of its mother [10]. They used subjective sensory-motor information alone, but the method did not enable a mobile robot to estimate its body shape so as to help the robot plan its trajectory of movement. They provided topological relationships and

the proximity of sensors and actuators, or representations that were difficult to use for further motion planning and execution.¹ Sturn et al. proposed a method by which a robot can learn body schema through self-perception in an unsupervised manner. However, this method requires an objective observation of the robot's posture, which cannot be obtained by an animal [11]. As far as we know, a computational and robotic model that can estimate its self-body shape by using only subjective sensory-motor information has not yet been proposed. In this paper, we propose a computational and robotic model that can automatically estimate body shape.

When a robot cannot obtain objective information regarding its body shape, it will not be able to accurately measure its body shape. However, if a robot can construct an environmental model, e.g., build an environmental map, from its sensory-motor information, the robot might be able to estimate its body shape in relation to the estimated environmental model. The goal of our approach is the simultaneous estimation of an environmental model, i.e., a map of the environment, and a self-body model (namely, body shape).

In the past two decades, simultaneous localization and mapping (SLAM) was created, developed, and widely used [12]. SLAM simultaneously estimates a robot's position and posture and creates an environmental map by using a Bayesian and Monte Carlo-based approach. In particular, the simultaneous use of a particle filter and a grid map provides a comprehensive and powerful framework for developers to build mobile robots. However, in SLAM, it is usually assumed that a robot's position is a point that does not have any volume in the state space, although a robot has a certain volume in the real world.

In this context, we describe a new method that enables a robot to obtain subjective information, e.g., motor commands and distance sensor information, to automatically estimate its self-body shape, by extending SLAM. Our method simultaneously estimates not only a map and the robot's location but also the robot's body shape. We call this method simultaneous localization, mapping, and self-body shape estimation (SLAM-SBE). The method is based on Bayesian statistics. Experimental results show that a mobile robot can obtain an image of its body shape represented by an occupancy grid by using its sensory-motor information without any objective measurement of its body.

The remainder of this paper is organized as follows. Section 2 describes SLAM, which is the basis of our proposed method and is used for estimating an environmental model for a robot, i.e., a map. Section 3 presents SLAM-SBE, which is our proposed method. Section 4 demonstrates experiments using an off-the-shelf mobile robot called Turtlebot2 and shows the validity of our method. Section 5 concludes this paper and describes future directions of study.

¹We admit that the question of whether a biological system should have a self-body model independent of particular behaviors is open to discussion in embodied cognitive science and related fields.

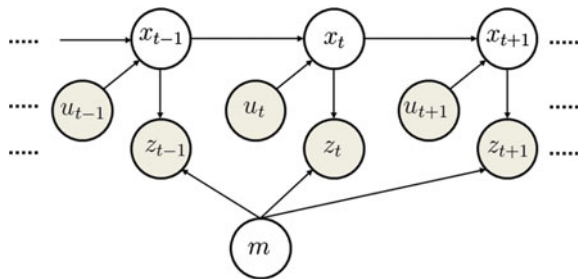
2 SLAM

SLAM is a sophisticated statistical method for localization and mapping of a mobile robot [12]. It is important for a mobile robot to estimate its position and posture. Usually, positional information cannot be directly and accurately obtained. Even if the robot uses a global positioning system (GPS), the measured value inevitably contains measurement errors. Here, we assume that a robot is equipped with distance sensors, which are widely used in mobile robotics. The localization problem usually performs estimation of the mobile robot’s position and posture by using sensory-motor information and a pre-existing map. However, the map itself is usually unknown before the robot moves around in its environment. To build a map, a robot must estimate its position and posture with some degree of accuracy. Therefore, localization and mapping problems should be solved simultaneously. SLAM simultaneously estimates a robot’s position and posture and creates an environmental map based on an integrative Bayesian and Monte Carlo-based approach.

Figure 2 shows a graphical model of SLAM, where x_t is the position and posture of the robot, u_t is the motor command output by the robot, and z_t is an observation obtained by sensors at time t . Map information is represented by m . This graphical model shows that SLAM assumes that the system dynamics follow the Markov decision process (MDP). By iteratively estimating x_t and m , the robot can estimate its position and posture and create a map.

Simultaneous use of Monte Carlo localization (MCL) and an occupancy grid provides a comprehensive algorithm to SLAM. The algorithm of the procedure called at every time step in SLAM is outlined in Algorithm 1. The procedure **sample_motion_model** generates a sample based on forward dynamics assumed by the system. The procedure **measurement_model** yields the likelihood of the observation z_t based on its estimated position and posture $x_t^{[k]}$ and map $m_{t-1}^{[k]}$. Here, m_t indicates m estimated at the t -th time step. Both of the functions are introduced in MCL. The procedure **updated_occupancy_grid** updates the occupancy grid representing a map on the basis of the robot’s position and posture $x_t^{[k]}$ and sensory observation z_t . For further details, please refer to a relevant textbook [12].

Fig. 2 Graphical model of SLAM



Algorithm 1 SLAM(X_{t-1}, u_t, z_t) (Grid-based Fast SLAM)

```

 $\bar{X}_t = X_t = \emptyset$ 
for  $k = 1$  to  $M$  do
   $x_t^{[k]} = \text{sample\_motion\_model}(u_t, x_{t-1}^{[k]})$ 
   $\omega_t^{[k]} = \text{measurement\_model}(z_t, x_t^{[k]}, m_{t-1}^{[k]})$ 
   $m_t^{[k]} = \text{updated\_occupancy\_grid}(z_t, x_t^{[k]}, m_{t-1}^{[k]})$ 
   $\bar{X}_t = \bar{X}_t + \langle x_t^{[k]}, \omega_t^{[k]}, m_t^{[k]} \rangle$ 
end for
for  $k = 1$  to  $M$  do
  draw  $i$  with probability  $\propto \omega_t^{[i]}$ 
  add  $\langle x_t^{[i]}, m_t^{[i]} \rangle$  to  $X_t$ 
end for
return  $X_t$ 

```

3 SLAM-SBE

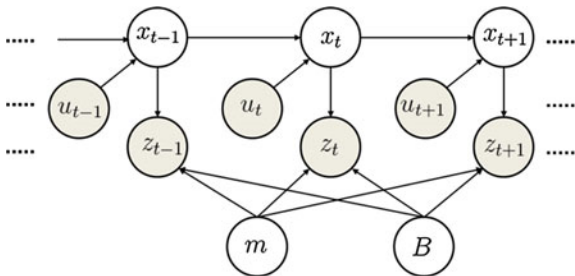
SLAM-SBE assumes that map information and self-body shape information possessed by a robot are both represented by occupancy grids. The robot has an occupancy grid B that represents its body shape.

The graphical model assumed in SLAM-SBE is shown in Fig. 3. The difference between the graphical model of SLAM (Fig. 2) and that of SLAM-SBE (Fig. 3) is that SLAM-SBE has a node for self-body information B . We assume that obstacles are not usually observed inside the robot's body. In this paper, we describe an algorithm that infers the self-body shape B in a recursive manner. The method of calculating the probability that the i -th cell of an occupancy grid represents a robot's body is as follows. In the occupancy grid, the probability of occupation of each cell is independent. Therefore, we describe an update formula for the i -th cell.

We assume that $B_i = 1$ indicates that the i -th cell of the occupancy grid for the self-body shape is a part of the body, where B_i is an occupancy grid for the self-body shape at time t . In contrast, $B_i = 0$ indicates that the i -th cell does not belong to the robot's body.

If the robot's position and posture x_t is determined, the occupancy grid of the self-body shape can be translated, rotated, and overlaid on the occupancy grid map of the

Fig. 3 Graphical model of SLAM-SBE



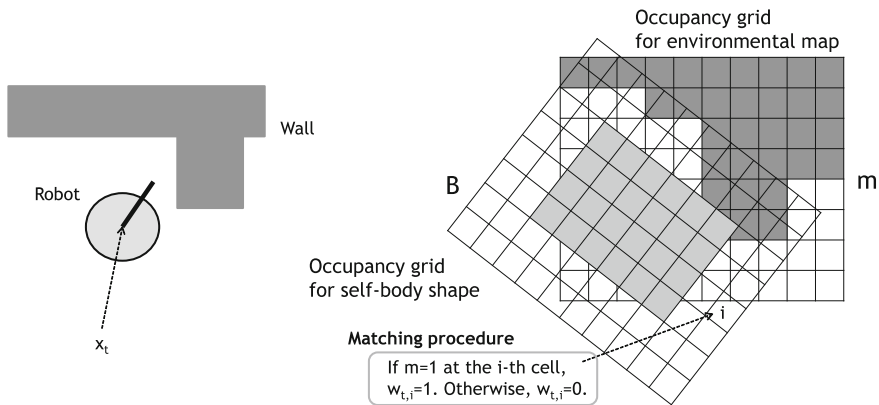


Fig. 4 Illustrative figure of overlay of two occupancy grids representing self-body shape and environmental map. Using the matching procedure for comparing cells for B and m , $w_{t,i}$ can be calculated

Algorithm 2 Bayesian filter-based SLAM – $SBE(X_{t-1}, u_t, z_t, G_{t-1}, \tilde{G}_{t-1})$

$X_t \leftarrow \text{SLAM}(X_{t-1}, u_t, z_t)$

for $i = 1$ to I **do**

 Calculate $P(x_t, m | z_{1:t}, u_{1:t})$ using X_t

 // Approximately, draw $(x_t^*, m^*) \sim X_t$, and use it as a deterministic probability distribution, i.e.,

$P(x_t, m | z_{1:t}, u_{1:t}) = \delta((x_t, m), (x_t^*, m^*))$.

 Calculate $P(w_{t,i} | z_{1:t}, u_{1:t})$ using transformation shown in Fig. 4 using $P(x_{1:t}, m | z_{1:t}, u_{1:t})$

 Update $G_{t,i}$ using (9)

 Update $\tilde{G}_{t,i}$ using (10)

end for

return X_t, G_t, \tilde{G}_t

environment, as shown in Fig. 4. Using this matching procedure shown in Fig. 4, it can be determined whether or not the i -th cell of the grid of the self-body shape has an obstacle in the occupancy grid for an environmental map, i.e., $w_{t,i} = 1$. In Fig. 4, cells estimated to contain an obstacle or body part are shown in gray.

We define $w_{t,i}$ as the occupancy of an obstacle in the i -th cell of the occupancy grid for the self-body shape B . If $w_{t,i} = 1$, the i -th cell in B is overlaid on an obstacle in the occupancy grid of the environmental map m , i.e., the gray cell in m . We assume that the conditional probabilities of the two variables are given as follows:

$$p(w_{t,i} = 1 | B_i = 1) = \epsilon \quad (1)$$

$$p(w_{t,i} = 0 | B_i = 1) = 1 - \epsilon \quad (2)$$

$$p(w_{t,i} = 1 | B_i = 0) = \gamma \quad (3)$$

$$p(w_{t,i} = 0 | B_i = 0) = 1 - \gamma \quad (4)$$

The probability of occupancy of the i -th cell in the grid for self-body shape is described as $P(B_i|z_{1:t}, u_{1:t})$, which can be calculated by adopting Bayes' rule and marginalizing $w_{1:t,i}$ as follows:

$$P(B_i|z_{1:t}, u_{1:t}) = \sum_{\forall x_{1:t}, m} P(B_i|x_{1:t}, m)P(x_{1:t}, m|z_{1:t}, u_{1:t}) \quad (5)$$

$$= \sum_{\forall w_{1:t,i}} P(B_i|w_{1:t,i})P(w_{1:t,i}|z_{1:t}, u_{1:t}) \quad (6)$$

$$\propto \sum_{\forall w_{1:t,i}} \left(\prod_{k=1}^t P(w_{k,i}|B_i) \right) P(B_i) \prod_{k=1}^t P(w_{k,i}|z_{1:k}, u_{1:k}) \quad (7)$$

$$= \sum_{\forall w_{t,i}} P(w_{t,i}|B_i)P(w_{t,i}|z_{1:t}, u_{1:t})P(B_i|z_{1:t-1}, u_{1:t-1}). \quad (8)$$

If we define two parameters $G_{t,i}$ and $\bar{G}_{t,i}$ as follows:

$$G_{t,i} = \sum_{w_{t,i} \in \{0,1\}} P(w_{t,i} | B_i = 1)P(w_{t,i}|z_{1:t}, u_{1:t})G_{t-1,i} \quad (9)$$

$$\bar{G}_{t,i} = \sum_{w_{t,i} \in \{0,1\}} P(w_{t,i} | B_i = 0)P(w_{t,i}|z_{1:t}, u_{1:t})\bar{G}_{t-1,i}, \quad (10)$$

we can obtain $P(B_i|z_{1:t}, u_{1:t})$ as follows:

$$P(B_i = 1|z_{1:t}, u_{1:t}) = \frac{G_{t,i}}{G_{t,i} + \bar{G}_{t,i}}. \quad (11)$$

The above formula describes a Bayesian filter for B . As a result, the Bayesian filter-based SLAM-SBE algorithm is derived, as shown in Algorithm 2. Using (11), at each step for all cells in the occupancy grid of self-body shape B , we can recursively update B . Note that a particle filter-based SLAM-SBE can be derived based on the same concept (see Appendix).

4 Experiment

To validate the proposed method, we conducted an experiment by using an off-the-shelf mobile robot.

Fig. 5 Turtlebot2 in condition 1



4.1 Conditions

In this experiment, we used an off-the-shelf mobile robot called Turtlebot2.² An overview of the robot is shown in Fig. 5. Turtlebot2 consists of a base robot with two wheels, an RGB-D camera sensor (Kinect), and a laptop computer. We used ROS (Robot Operating System) [13] Hydro as middleware for controlling the Turtlebot2. We used AMCL and gmapping packages of ROS for SLAM.

We prepared two robots with different body shapes, as shown in the Figure. In condition 2, additional top boards were attached, as shown in Fig. 6. We set the length and width of each cell in the occupancy grid as 5 cm. The parameters for self-body shape estimation are set to $\epsilon = 0.01$ and $\gamma = 0.4$.

In this experiment, the mobile robot was controlled by an experimenter using a PlayStation3 controller.³ We had the mobile robot move around a corridor and estimate its position and posture, map, and self-body shape. Body shape estimation was conducted after the environmental map converged. At each step, the particle with the highest likelihood was selected from among X_t , and body shape information B was updated using position and posture information x_t obtained by the particle. A predetermined forward model for actuators and a measurement model for the sensors of the Turtlebot2 were used.

²Turtlebot2: <http://www.turtlebot.com/>.

³<https://www.playstation.com/en-us/explore/accessories/dualshock-3-ps3/>.

Fig. 6 Turtlebot2 in condition 2



4.2 Results

Figures 7, 8 and 9 show the estimated occupancy grids for self-body shape B at each step in condition 1. The gray-scale color shows the probability for the occupancy grid, i.e., black indicates that the probability is 1, and white indicates a probability of 0. The obtained occupancy grid for the environmental map is shown in Fig. 10. In this experiment, the corridor has a very simple shape. Therefore, the robot could successfully estimate its environmental map.

Figure 11 shows the estimated occupancy grid for self-body shape B in condition 2. The comparison between Figs. 9 and 11 shows that the robot could estimate its body shape using B . Figures 12 and 13 show rough comparisons between the estimated body shape and the real body shape in condition 1 and 2, respectively.

Fig. 7 Occupancy grid for self-body shape B at time step 20 (condition 1)

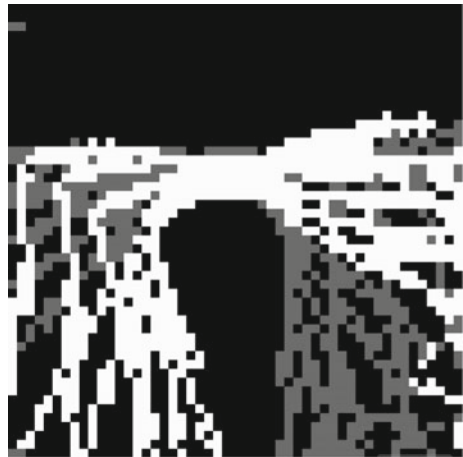


Fig. 8 Occupancy grid for self-body shape B at time step 40 (condition 1)



Fig. 9 Occupancy grid for self-body shape B at time step 72 (condition 1)



Here, we assume that the body shape is a continuum, and omit separated nodes during post processing.

The results of the quantitative evaluation of body shape estimation are shown in Table 1. By treating the body shape estimation problem as an information retrieval task that attempts to estimate whether or not each cell contains a body part, the precision, recall, and F-measure were calculated. To calculate these values, original pictures of the mobile robots were matched to the estimated body shapes and manually compared. The results show that SLAM-SBE could relatively accurately estimate the shape of the robot's body. However, the error in length in condition 2 was relatively large. The robot's only depth sensor is facing forward. Therefore, the localization error in the lateral direction was relatively large, which might affect the estimation error of the robot's body shape. In condition 1, the body shape was estimated to be slightly larger than the actual body. This might result because the robot

Fig. 10 Occupancy grid for environmental map estimated in the experiment

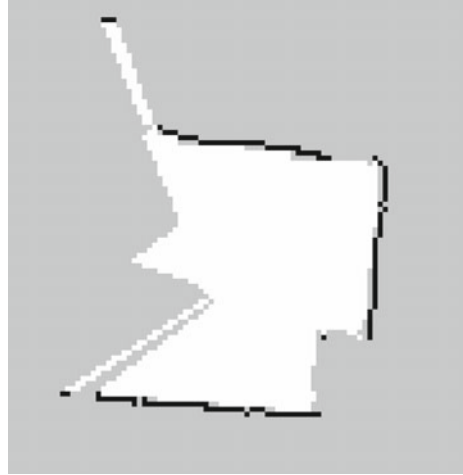


Fig. 11 Occupancy grid for self-body shape B using a long mobile robot (condition 2)

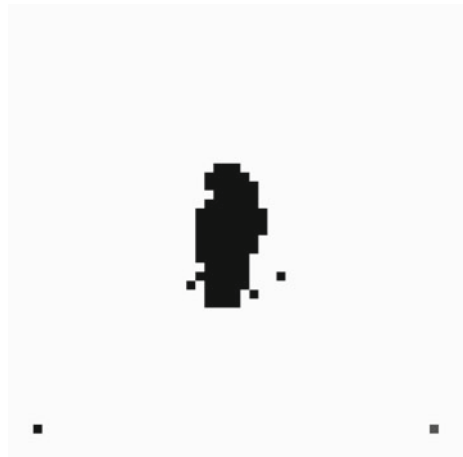
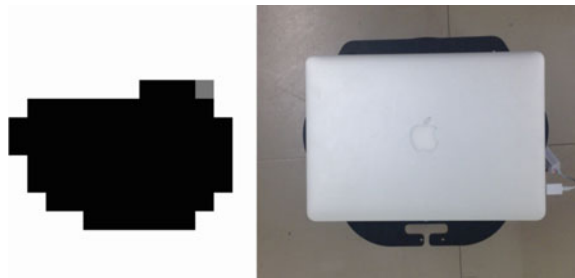


Fig. 12 Comparison between the estimated body shape and the real body shape in condition 1



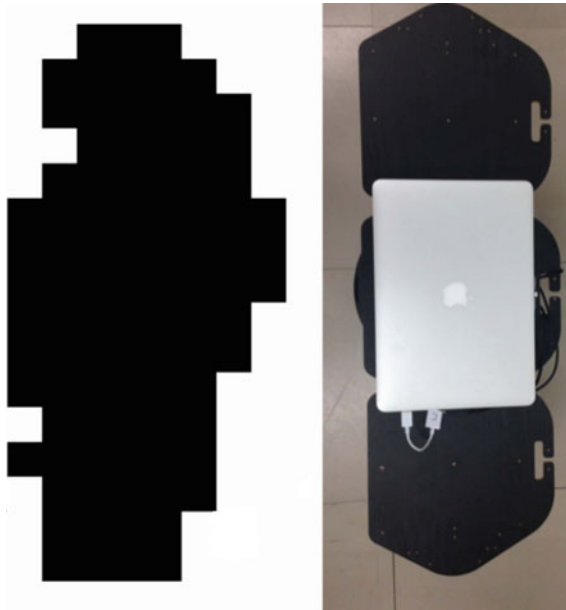


Fig. 13 Comparison between the estimated body shape and the real body shape in condition 2

Table 1 Quantitative evaluation of estimated body shape

	Condition 1		Condition 2	
	Length (cm)	Width (cm)	Length (cm)	Width (cm)
Ground truth	35	35	100	35
Estimated size	60	40	80	40
Precision	0.80		0.97	
Recall	1.00		0.61	
F-measure	0.89		0.75	

was controlled so as to avoid collisions with walls. In addition, the Kinect sensor could measure the distance only between 1.2 and 3.5 m; thus, the distance to objects that are located within 1.2m could not be correctly measured. This phenomenon might degrade the performance of body shape estimation.

5 Conclusion and Future Directions

This paper describes a new method for estimating the body shape of a mobile robot by using only the robot’s sensory-motor information. We call the method SLAM-SBE. The method was developed by extending the SLAM method. Experimental

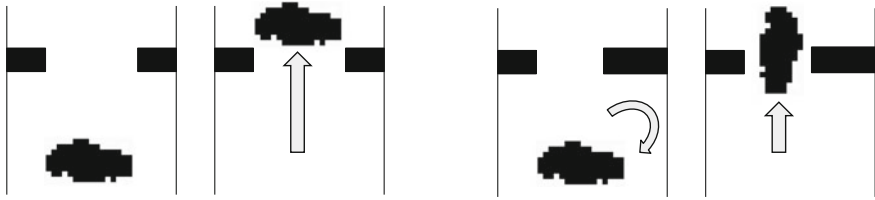


Fig. 14 Example of generation of adaptive behavior based on estimated self-body shape

results showed that a mobile robot could estimate its body shape, represented by an occupancy grid, using only its sensory-motor information (i.e., without any objective measurement of its body).

In this paper, we considered only the estimation problem. However, the estimated body shape can be used by the mobile robot in a variety of ways. For example, when the robot finds a narrow aperture, the robot must determine how to pass through the aperture. If a robot knows its body shape, the robot can plan its movement by using various planning methods, e.g., the A* algorithm or a Monte Carlo tree search, and automatically generate adaptive behaviors depending on its estimated body shape, as shown in Fig. 14.

The accuracy of self-body shape estimation was still limited. To reduce the estimation errors and to increase the performance of the estimation, there are many ways we can employ, e.g., introducing more accurate distance sensors like URG sensors, and using more fine grid map. Our proposed framework, i.e., SLAM-SBE, is so general that it can employ various kind of sensors and maps.

In this paper, we assume that the mobile robot moves in a 2D space. However, SLAM-SBE is a general framework, and it can easily be extended and applied to a robot moving in a 3D space. In the formulation of SLAM-SBE, the body shape B is assumed to be constant. This is the same assumption that is made for the environmental map in SLAM. To extend our model to one that is applicable to a system with multiple links, e.g., humanoid robots, humans, and many mammals, a deformable map representation should be included in SLAM. Extending SLAM-SBE and obtaining a method that can estimate the body shape of a multi-joint system is one of our future directions. Additionally, in the experiment, we assumed that the robot has a forward model for actuators and a measurement model for sensors, as assumed in many SLAM problems. However, it is quite likely that many biological systems do not know the models before evaluating the triadic interaction between the brain, body, and environment. Therefore, a method that can estimate body shapes and environmental maps without a priori forward and measurement models needs to be developed.

Acknowledgements This research was partially supported by a Grant-in-Aid for Scientific Research on Innovative Areas 2015–2017 (15H01670) funded by the Ministry of Education, Culture, Sports, Science, and Technology, Japan.

Algorithm 3 Particle filter-based SLAM – SBE(X_{t-1}, u_t, z_t)

```

 $\bar{X}_t = X_t = \emptyset$ 
for  $k = 1$  to  $M$  do
   $x_t^{[k]} = \text{sample\_motion\_model}(u_t, x_{t-1}^{[k]})$ 
   $\omega_t^{[k]} = \text{measurement\_model}(z_t, x_t^{[k]}, m_{t-1}^{[k]}, B_{t-1}^{[k]})$ 
   $m_t^{[k]} = \text{updated\_occupancy\_grid\_for\_map}(z_t, x_t^{[k]}, m_{t-1}^{[k]}, B_{t-1}^{[k]})$ 
   $B_t^{[k]} = \text{updated\_occupancy\_grid\_for\_body}(z_t, x_t^{[k]}, m_{t-1}^{[k]}, B_{t-1}^{[k]})$ 
   $\bar{X}_t = \bar{X}_t + \langle x_t^{[k]}, \omega_t^{[k]}, m_t^{[k]}, B_t^{[k]} \rangle$ 
end for
for  $k = 1$  to  $M$  do
  draw  $i$  with probability  $\propto \omega_t^{[i]}$ 
  add  $\langle x_t^{[i]}, m_t^{[i]}, B_t^{[i]} \rangle$  to  $X_t$ 
end for
return  $X_t$ 

```

Appendix

The recursive property of the update formula shown in (9) and (10) enables us to derive particle-filter-based SLAM-SBE in the same way as conventional MCL and SLAM. The outline of particle-filter-based SLAM-SBE is presented in Algorithm 3 for reference. We exclude precise descriptions of each function in Algorithm 3. They can be easily derived based on the definitions of SLAM and the derived Bayesian filter.

References

1. Iriki, A., Tanaka, M., Iwamura, Y.: Coding of modified body schema during tool use by macaque postcentral neurones. *Neuroreport* **7**(14), 2325–2330 (1996)
2. Maravita, A., Iriki, A.: Tools for the body (schema). *Trends Cogn. Sci.* **8**(2), 79–86 (2004)
3. Ramachandran, V.S., Blakeslee, S.: *Phantoms in the Brain: Human Nature and the Architecture of the Mind*. Fourth Estate (1998)
4. Sonoda, K., Moriyama, T., Asakura, A., Furuyama, N., Gunji, Y.P.: Can hermit crabs perceive affordance for aperture crossing? In: *Proceedings of European Conference on Complex Systems*, pp. 553–557 (2012)
5. Sonoda, K., Asakura, A., Minoura, M., Elwood, R.W., Gunji, Y.-P.: Hermit crabs perceive the extent of their virtual bodies. *Biol. Lett.* **8**(4), 495–497 (2012)
6. Nabeshima, C., Kuniyoshi, Y., Lungarella, M.: Adaptive body schema for robotic tool-use. *Adv. Robot.* **20**(10), 1105–1126 (2006)
7. Olsson, L., Nehaniv, C., Polani, D.: From unknown sensors and actuators to actions grounded in sensorimotor perceptions. *Connection Sci.* **18**(2), 121–144 (2006)
8. Sugiura, K., Matsubara, D., Katai, O.: Construction of robotic body schema by extracting temporal information from sensory inputs. In: *SICE-ICASE International Joint Conference*, pp. 302–307 (2006)
9. Yoshikawa, Y., Kawanishi, H., Asada, M., Hosoda, K.: Body scheme acquisition by cross modal map learning among tactile, visual, and proprioceptive spaces. In: *International Workshop on Epigenetic Robotics*, pp. 181–184 (2002)

10. Mori, H., Kuniyoshi, Y.: A human fetus development simulation: self-organization of behaviours through tactile sensation. In: IEEE 9th International Conference on Development and Learning (ICDL), pp. 82–87 (2010)
11. Sturm, J., Plagemann, C., Burgard, W.: Unsupervised body scheme learning through self-perception. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3328–3333 (2008)
12. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2005)
13. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source robot operating system. In: ICRA workshop on open source software, vol. 3, p. 5 (2009)

Objective Measurement of Dynamic Balance Function by the Simultaneous Measurement of the Center of Gravity (COG) and Center of Pressure (COP)

Masahiko Mukaino, Fumihiko Matsuda, Ryoma Sassa, Kei Ohtsuka, Nobuhiro Kumazawa, Kazuhiro Tsuchiyama, Shigeo Tanabe and Eiichi Saitoh

Abstract Although a posturography is commonly used for objective evaluation of static balance function, dynamic balance function is usually evaluated only with clinical scales. Simplified objective measurement systems for the evaluation of dynamic balance function need to be developed. In this preliminary study, we attempted to develop an index for the objective measurement of dynamic balance function from COP-COG data. The subjects comprised nine hemiparetic post-stroke patients and five healthy subjects. The simultaneous measurements of COG and COP were performed using a three-dimensional motion analysis system (Kinema-tracer, KisseiComtec, Japan) combined with force plate analysis. As indices for evaluating dynamic balance function, the latency of COP passing COG after heel contact (LCP) and the averaged $|COPI - COG|$ subtraction value during stance phase (ASV) were calculated. For evaluating validity of the measurement, the Berg Balance Scale, a frequently used clinical balance scale, was used. The results

M. Mukaino (✉) · E. Saitoh

Department of Rehabilitation Medicine I, School of Medicine,
Fujita Health University, 1-98 Dengakugakubo, Kutsukake-Cho,
Toyoake, Aichi, Japan
e-mail: mmukaino@fujita-hu.ac.jp

F. Matsuda · K. Ohtsuka (✉) · S. Tanabe

Faculty of Rehabilitation, School of Health Sciences,
Fujita Health University, 1-98 Dengakugakubo,
Kutsukake-Cho, Toyoake, Aichi, Japan
e-mail: ohtsuka@fujita-hu.ac.jp

R. Sassa · K. Tsuchiyama

Department of Rehabilitation, Fujita Health University Hospital,
1-98 Dengakugakubo, Kutsukake-Cho, Toyoake, Aichi, Japan

N. Kumazawa

Graduate School of Health Sciences, Fujita Health University,
1-98 Dengakugakubo, Kutsukake-Cho, Toyoake, Aichi, Japan

© Springer International Publishing AG 2017

W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_6

showed significant differences (0.13 ± 0.02 vs. 0.29 ± 0.23 s) between the healthy subjects and patients in LCP, and large, yet insignificant, differences (4.3 ± 0.5 vs. 2.7 ± 2.0 cm) in ASV. The ASV was strongly correlated with BBS. A strong correlation was observed between COG acceleration and ASV, except in one patient, who had a severe balance disorder. These results may encourage further investigation into the feasibility of COP-COG measurements for balance measurement.

Keywords Motion analysis • Balance disorder • Dynamic balance function

1 Introduction

Impaired dynamic balance function is an important risk factor for falls. Many clinical batteries, such as the Berg Balance Scale (BBS) [1] and the Timed Up and Go Test [2], have been developed for evaluating static and dynamic balance function and are frequently used in rehabilitation clinics. Although those scales are valid for clinical evaluation, more detailed objective method would be needed for further understanding of the mechanism of the balance disorder. For the further objective measurement of balance function, posturography, which measures the trajectory of the center of pressure (COP), is used [3]. However, because the trajectory of COP is considerably affected by bodily movements, posturography is usually only used for the measurement of static balance function. For the measurement of dynamic balance function, computerized dynamic posturography (CDP), which provides standardized external perturbation for patients, has to be used [4].

In contrast to frequent usage of posturography for evaluating static balance function, CDP is not frequently performed for the evaluation of dynamic balance function. The high cost of the CDP system and the time required for the measurements may reduce CDP usage in clinics.

More simplified measurement systems for the evaluation of dynamic balance function need to be developed. Ideally, these should evaluate not only reactive postural responses but also the balance response during spontaneous movement, which is generally referred to as anticipatory postural adjustment [5].

Dynamic balance function is often referred to as the stability of control of center of gravity (COG) acceleration [6, 7].

As the horizontal acceleration of COG is mainly regulated by the COP-COG relationship [8], the simultaneous measurement of COG and COP movements may reflect the ability to control balance by COP. In this preliminary study, we attempted to develop an index for the objective measurement of dynamic balance function from COP-COG data. The underlying mechanisms were also discussed.

2 Materials and Methods

2.1 Subjects

The subjects comprised 9 hemiparetic post-stroke patients and 5 healthy subjects. The inclusion criteria for the patients were as follows: (1) able to stand and step 5 times and (2) not dependent on either the assistance of another person or assisting devices (e.g., handrail, canes, or orthotics). The patients included 6 males and 3 females (6 patients with right hemiplegia and 3 patients with left hemiplegia) with a mean age of 57 ± 8 years. The mean total score of the Stroke Impairment Assessment Set (SIAS) motor test was 9 ± 1 .

2.2 Measurements

The simultaneous measurement of COG and COP was performed using a three-dimensional motion analysis system (Kinematracer, KisseiComtec, Japan) combined with force plate analysis. For the calculation of COG, markers were attached to 10 landmark sites on the body bilaterally: the acromion processes, the hip joints (one third of the distance between the greater trochanter and the anterior superior iliac spine), the knee joints (midpoint of the lateral epicondyle of the femur), the lateral malleoli, and the fifth metatarsal heads. The virtual COG was calculated using a software that estimated the equation of Ehara and Yamamoto [9]. Briefly, the centers of seven segments (trunk, bilateral thighs, bilateral lower thighs, and bilateral feet) were defined from the markers. The mass ratios of each segment were hypothesized to be as follows: trunk 0.66, thigh 0.1, lower thigh 0.05, and foot 0.02. COG was then calculated as the composition center of the seven segments.

Subjects were instructed to step 5 times on the force plate. Before and after the task, subjects remained standing for 5 s. During stepping, subjects were instructed to raise the foot as high as possible and to keep both feet within shoulder width to reduce compensation by the foot position. To investigate the validity of the index, the BBS [1], which is a frequently used scale for evaluating balance function, was employed for comparative measurement. BBS consists of the several balance tests and the total score ranged 0–66, where the cut off value of balance disorder is 44. All of the healthy subjects showed the score of 66. This study was approved by the ethical committee of our institution.

2.3 Data Analysis

The lateral displacements of COG and COP were compared. The acceleration of COG in the medial direction (braking followed by acceleration in the other

direction) is necessary for maintaining balance during stepping. When COG moves laterally, the leg in the direction of movement is the main source of power for “braking.” Therefore, the data between the bilateral heel contact (HC) were used for the present analysis.

Acceleration in the medial direction is produced when COP is placed laterally to COG. Therefore, the latency of COP passing COG after HC (LCP) and the averaged $|COPI - COG|$ subtraction value (ASV) were calculated. The acceleration value of COG was also calculated.

All statistical analyses were conducted in JMP11 (SAS Institute Inc. Cary, NC, USA). Descriptive statistics (mean and standard deviation) were computed. Differences between the stroke and control groups were investigated using the Mann–Whitney U-test. For multiple testing, the Bonferroni correction was applied. A p-value of less than 0.05 was considered to be statistically significant.

3 Results

Significant differences between the patients and healthy subjects were observed in LCP. Furthermore, although not significant, there was a large difference in ASV between the two groups (Fig. 1a, b). Patients with high BBS tended to show high ASV; a similar tendency was seen in healthy subjects. In contrast, the patients with

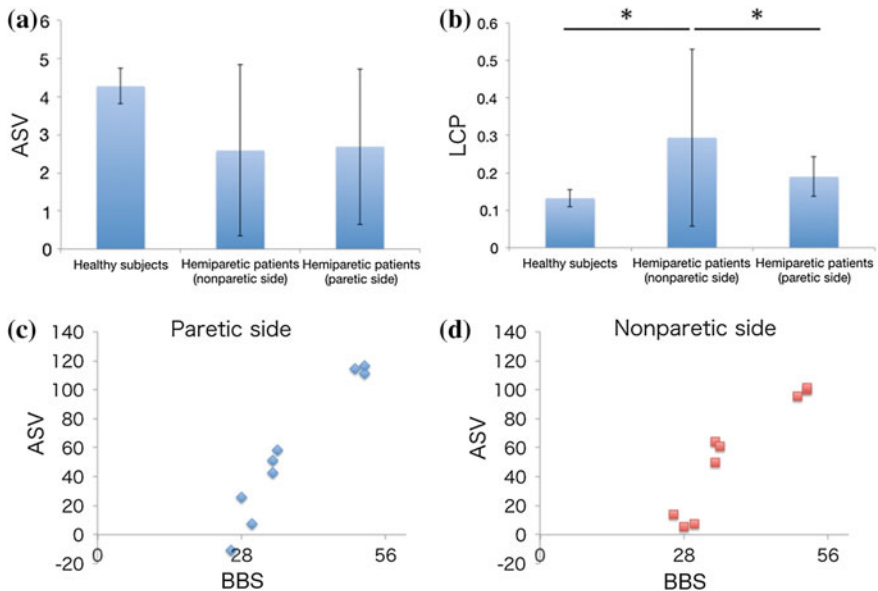


Fig. 1 a, b Comparison of ASV and LCP between hemiparetic patients and healthy subjects. c, d Relationship between BBS and ASV in the stroke patients

Table 1 The correlation coefficient of COG acceleration and |COPI-COG| value, score of BBS and step width

	CC	BBS	Step width
Pt 1	-0.49	26	21
Pt 2	-0.85	28	20.4
Pt 3	-0.84	30	23.2
Pt 4	-0.94	34	25.4
Pt 5	-0.93	34	22.6
Pt 6	-0.94	35	18.3
Pt 7	-0.96	50	21.1
Pt 8	-0.91	52	22.3
Pt 9	-0.90	52	21.6
Hs 1	-0.91		25.3
Hs 2	-0.86		15.6
Hs 3	-0.90		18.2
Hs 4	-0.92		19.2
Hs 5	-0.93		21.9

Pt: Stroke patients; Hs: Healthy subjects

CC: Correlation coefficient of COG acceleration and |COPI-COG| value BBS: Berg Balance Scale

low BBS scores tended to show extremely low ASV (Fig. 1c, d). ASV and COG acceleration closely correlated with each other, except in one patient who presented a low BBS score (Table 1).

4 Discussion

In this preliminary study, significant differences in LCP and ASV were observed between patients and healthy subjects. Furthermore, although the sample size was small, ASV showed a strong tendency for correlation with BBS. These results may encourage further investigation into the feasibility of COP-COG measurements for balance measurement.

The acceleration pattern of COG is frequently reported to be closely related with dynamic stability [6, 7, 10]. For a stable pattern of acceleration, the ability to control COP is important because COG acceleration is closely related to the COP-COG relationship [11, 12]. In this study a strong correlation was observed between COG acceleration and ASV in most of the subjects.

However, one patient with poor balance ability showed a low correlation coefficient (CC) between COG acceleration and ASV. Moreover, two patients with low BBS scores showed relatively low CCs (0.85 and 0.84). One healthy subject with a considerably small step width presented with a relatively low CC (Table 1). These results are possibly related to compensatory movements. When considering the reactive postural response against external perturbation, control of foot pressure

as well as compensatory movement of the trunk or steps is observed. Therefore, ASV possibly reflects only one of the several factors of balance function, particularly in patients with balance disorders. Considering that patients with severe hemiparesis often employ various compensatory strategies for standing or walking [13], this system of balance ability should be evaluated with various compensatory strategies.

Nonetheless, these results show the feasibility of using the COP-COG relationship for the evaluation of dynamic balance function and encourage further study.

5 Conclusion

In this study, the feasibility of a simplified index that showed dynamic balance function from the COP-COG relationship was investigated. The present preliminary results encourage the confirmation of these results and the need for further studies, including investigations for compensatory movements, to demonstrate the holistic system of balance maintenance in human dynamic movement.

References

1. Berg, K.O., Maki, B.E., Williams, J.I., Holliday, P.J., Wood-Dauphinee, S.L.: Clinical and laboratory measures of postural balance in an elderly population. *Arch. Phys. Med. Rehabil.* **73**, 1073–1080 (1992)
2. Podsiadlo, D., Richardson, S.: The timed “Up & Go”: a test of basic functional mobility for frail elderly persons. *J. Am. Geriatr. Soc.* **39**, 142–148 (1991)
3. Visser, J.E., Carpenter, M.G., van der Kooij, H., Bloem, B.R.: The clinical utility of posturography. *Clin. Neurophysiol.* **119**, 2424–2436 (2008)
4. Nashner, L.M., Peters, J.F.: Dynamic posturography in the diagnosis and management of dizziness and balance disorders. *Neurol. Clin.* **8**, 331–349 (1990)
5. Lepers, R., Brenière, Y.: The role of anticipatory postural adjustments and gravity in gait initiation. *Exp. Brain Res.* **107**, 118–124 (1995)
6. Menz, H.B., Lord, S.R., Fitzpatrick, R.C.: Acceleration patterns of the head and pelvis when walking are associated with risk of falling in community-dwelling older people. *J. Gerontol. A Biol. Sci. Med. Sci.* **58**, M446–M452 (2003)
7. Toebes, M.J., Hoozemans, M.J., Furrer, R., Dekker, J., van Dieën, J.H.: Local dynamic stability and variability of gait are associated with fall history in elderly subjects. *Gait. Posture.* **36**, 527–531 (2012)
8. Winter, D.: Human balance and posture control during standing and walking. *Gait. Posture.* **3**, 193–214 (1995)
9. Ehara, Y., Yamamoto, S.: *Introduction to Body Dynamics-Analysis of Standing up Movement*. Ishiyaku Publishers, Tokyo (2001) (in Japanese)
10. Bisi, M.C., Riva, F., Stagni, R.: Measures of gait stability: performance on adults and toddlers at the beginning of independent walking. *J. Neuroeng. Rehabil.* **11**, 131 (2014)

11. Masani, K., Vette, A.H., Kouzaki, M., Kanehisa, H., Fukunaga, T., Popovic, M.R.: Larger center of pressure minus center of gravity in the elderly induces larger body acceleration during quiet standing. *Neurosci. Lett.* **422**, 202–206 (2007)
12. Brenière, Y., Cuong Do, M., Bouisset, S.: Are dynamic phenomena prior to stepping essential to walking? *J. Mot. Behav.* **19**, 62–76 (1987)
13. Balaban, B., Tok, F.: Gait disturbances in patients with stroke. *PM. R.* **6**, 635–642 (2014)

Development of a Master–Slave Finger Exoskeleton Driven by Pneumatic Artificial Muscles

Takuya Urino, Shuhei Ikemoto and Koh Hosoda

Abstract This paper presents a master–slave finger exoskeleton developed to allow subjects whose brain activity is being measured by functional magnetic resonance imaging (fMRI) to remotely perform tasks. The MRI environment requires the device to be free from metal components and strongly immobilized, which can reduce the device’s versatility and ease of setup. To overcome these limitations, we designed a finger exoskeleton using pneumatic artificial muscles, which can be made metal–free and used for not only actuators but also sensors. We also proposed a symmetric, bilateral control method for the device, and experimentally validated device performance and its control method.

Keywords Pneumatic artificial muscles · Master–slave · fMRI · Metalless

1 Introduction

The human hand has one of the most complex musculoskeletal structures in the human body and is very dexterous. Many fields of brain science measure brain activity when a subject grasps or manipulates an object [1]. Functional magnetic resonance imaging (fMRI), which can measure brain activity on an arbitrary cross-section at a high resolution, is widely used in this field. To measure brain activity while a subject performs a task during imaging, some fMRI-compatible devices have been developed [2–4]. These devices, however, are difficult to set up, often have metallic parts and must be strongly immobilized by fixing them to the floor or the fMRI equipment. These limitations have resulted in devices typically designed for

T. Urino (✉) · S. Ikemoto (✉) · K. Hosoda
Graduate School of Engineering Science, Osaka University, Osaka, Japan
e-mail: urino.takuya@arl.sys.es.osaka-u.ac.jp

S. Ikemoto
e-mail: ikemoto@sys.es.osaka-u.ac.jp

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_7

specific tasks. These experiments could be improved if a master–slave exoskeleton made from non-metal components could be designed to allow subjects to perform a wide variety of tasks.

Various devices that can be attached to human fingers while allowing joint movement have been developed for the purposes of rehabilitation and power assistance. In these designs, actuators and mechanisms were selected according to the specific purpose of the device. Two main types of actuator devices exist: electric motors [5–8] and pneumatic artificial muscles (PAMs) [9–12]. The mechanical structure of these devices can be further classified by the type of mechanism: independent drives of joints [5, 6] and interlocking or underactuated drives [7, 11]. These devices can already have subjects remotely perform tasks, but they consist of metal parts, electromagnetic motors, and/or sensors that emit electric signals rendering them unsuitable for use in the fMRI environment. The fMRI-incompatible parts are very important to the designs and cannot be easily replaced by non-metal components. Some attempts have been made to develop fMRI-compatible devices [13], however, these approaches do not use master–slave devices and have unresolved issues with sensing and control. To achieve better versatility and compatibility, we undertook the development of a master–slave exoskeleton device using fMRI-compatible, non-magnetic materials and proposed a bilateral control method for the device.

We chose to use PAMs as the actuators and the independent drive of a joint as the mechanism for developing our finger exoskeleton as shown in Fig. 1. Although PAMs typically contain small metal parts in the air joint, end cap and end cap fixture,

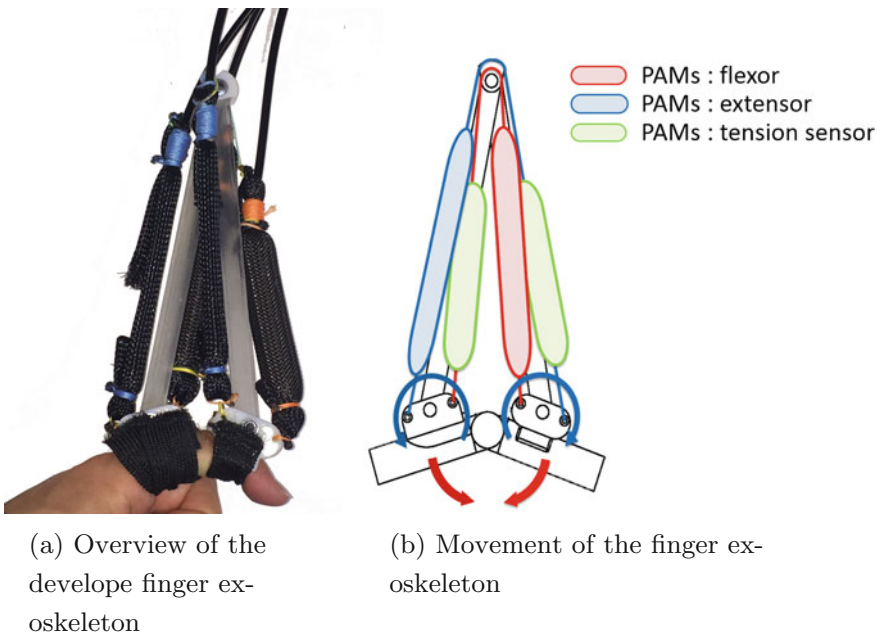


Fig. 1 Developed finger exoskeleton driven by metal-free PAMs

they are all easily replaceable with non-metal parts. However, conventional sensors (e.g., potentiometers, torquemeters, Inertial Measurement Units (IMUs)) transmit information via electric signals and as such are all inherently fMRI–incompatible. To address this problem, we used PAMs not only to drive the finger exoskeleton but also to sense the state in the form of air pressure that corresponds to the tension exerted on the PAM. The tension encoded as air pressure is measured outside of the room where the fMRI equipment is located.

The master and slave exoskeletons were designed in the same manner and were bilaterally controlled to relay movements from each other. In general, bilateral control methods are explained as to control both the master and the slave by referring to states of each other as the desired states. Symmetric bilateral control measures the angle of all joints in the master and the slave and uses this information in the angle feedback control. Neither joint angle sensors (such as potentiometers and rotary encoders), nor length sensors are fMRI–compatible. Therefore, we implemented a bilateral control method for the developed device using pressure as the sole measurement to maintain fMRI–compatibility.

Many different static models of PAMs have been proposed [14–17], in which two of the three model variables (pressure, tension, or length) must be known to determine the third. Our design primarily measures the pressure of the PAMs, and tension can also be measured in the form of pressure using PAMs as sensors. The pressure and tension information provided by the PAMs serves to implement a bilateral control method for the device driven by PAMs. Similarly, we proposed a direct teaching method based on pressure and tension information of the PAMs for complex musculoskeletal robots driven by PAMs [18, 19]. For the present research, we extend the direct teaching method to symmetric bilateral control by bilaterally applying the relationship which was used in direct teaching method and introducing several constraints characteristic to bilateral control.

The aim of this study was to develop an fMRI–compatible, master–slave finger exoskeleton and a bilateral control method for the device. Here we describe the hardware design, the symmetric bilateral control method and experimentally confirm the validity of the system by showing that the slave successfully tracks the master.

2 Construction of the Finger Exoskeleton

Table 1 presents the specifications. The device has one degree-of-freedom (DOF) when both ends are attached to two segments of a finger. We used the same structure to the master and slave exoskeletons. In application, for example, we assume that the master is attached to a human hand and the slave is attached to remote controlled hand. The exoskeleton can be used to drive many joints and Fig. 1a shows the arrangement for driving a proximal interphalangeal (PIP) joint. We adopt a 3D printer (Objet260, Stratasys Ltd.) to create the empirically designed linkage for the finger exoskeleton. The contraction length of a PAM strongly depends on the whole relaxing length, which was evaluated through iterative prototyping to test different

Table 1 Specifications of the developed finger exoskeleton

Degrees of freedom		1 [DOF]
Number of PAMs	Actuator	2
	Sensor	2
Weight of finger exoskeleton		120 [g]
Length of air tube		1.5 [m]
Maximum air pressure		0.6 [MPa]

geometries and confirm the strength of the structure. Figure 1b shows four PAMs: two PAMs (blue and red) actuate the mechanism, and two PAMs (green) sense the axial tensions. Each green PAM is connected to either the blue or red PAM in series to measure the tension of the actuating PAM. When the red PAM contracts and the blue PAM extends, the finger exoskeleton generates the torque depicted by the blue arrow in Fig. 1b. The bending torque of the joint is depicted by the red arrow. The blue and red PAMs operate as the extensor and flexor, respectively.

3 Bilateral Control Specialized for the Developed Device

The symmetric, bilateral control method specialized for the master–slave finger exoskeleton uses only the pressure information from the PAMs. This method is based on two relationships between the internal pressure and the axial tension of the PAMs. The first relationship is derived by introducing a constraint requiring the two corresponding PAMs in the master and slave exoskeletons to have the same lengths. This relationship is used to compute the pressures required to control the lengths of the PAMs. The second relationship is derived from the assumption that the amount of internal air is sufficiently large and invariant. This relationship is used to estimate the axial tensions from the pressures of the PAMs as sensors.

3.1 Basic Bilateral Control for the Master–Slave System

Bilateral control is generally used for master–slave systems. We explain bilateral control using the 1-DOF arm shown in Fig. 2 as a simplified model of the developed device. Here, θ and $\hat{\theta}$ indicate the joint angles of the master–slave 1-DOF robot arm. Variables with the caret ($\hat{\cdot}$) represent the slave state, while variables without the caret represent the master state. The variables P_I , P_O , F_I , F_O , and L_I and L_O indicate the internal pressures, axial tensions, and lengths of the extensor and flexor, respectively. Using symmetric bilateral control, the torque around the joint can be expressed as follows:

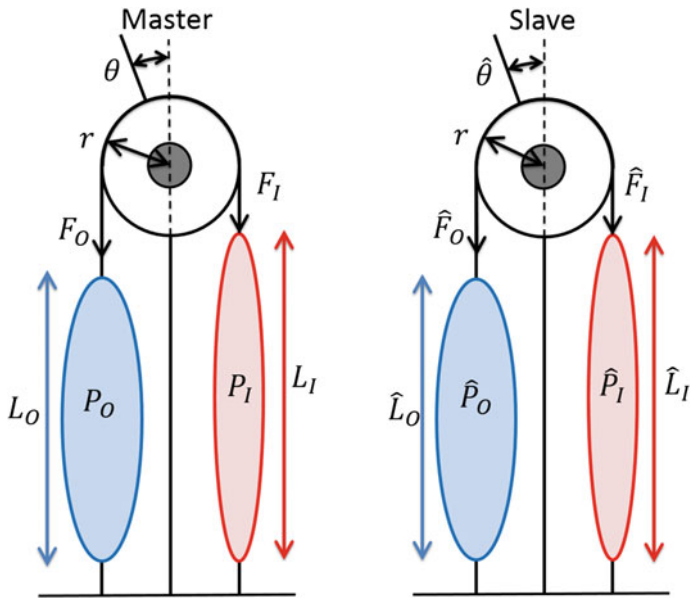


Fig. 2 Equivalent and simple model of the developed master–slave finger exoskeleton

$$\tau = K(\hat{\theta} - \theta), \hat{\tau} = K(\theta - \hat{\theta}). \quad (1)$$

where τ and K indicate the joint torque of the master and the positive feedback gain, respectively. According to Fig. 2, Eq. 1 can be rewritten using the lengths of the PAMs:

$$\tau = \frac{K}{r}(L_O - \hat{L}_O), \hat{\tau} = \frac{K}{r}(\hat{L}_I - L_I). \quad (2)$$

Because PAM lengths are difficult to measure in this application it is necessary to implicitly realize the length feedback by using the corresponding pressures and tensions.

3.2 Relationship Between Pressures and Tensions When the Master and Slave Take the Same Posture

Chou and Hannaford [15] proposed a simple static model of PAMs, which has three state variables and several intrinsic constant parameters. It can be expressed by the following equation:

$$F = \frac{P}{4\pi n^2}(3L^2 - b^2). \quad (3)$$

where F , P , and L are state variables that indicate the axial tension, internal pressure, and axial length of the PAMs, respectively. The variables n and b are intrinsic constant parameters representing the number of turns and the length of nylon thread constituting the sleeve of the PAMs, respectively. Ikemoto et al. [18] proposed a simple direct teaching method based on this model. According to them, muscle lengths should be the same between the teaching phase and reproduction phase, and the relationship between the axial tension and internal pressure is satisfied when the taught motion is successfully reproduced. In particular, the lengths of PAMs during the reproduction phase L and teaching phase \hat{L} can be derived from Eq. 3 as follows:

$$L = \sqrt{\frac{4\pi n^2 F}{3P} + \frac{b^2}{3}}, \hat{L} = \sqrt{\frac{4\pi n^2 \hat{F}}{3\hat{P}} + \frac{b^2}{3}}. \quad (4)$$

where the caret indicates variables during the teaching phase. Introducing the constraint that $L = \hat{L}$, which expresses the purpose of the direct teaching, the following relationship can be obtained:

$$\frac{P}{F} = \frac{\hat{P}}{\hat{F}}. \quad (5)$$

Our research group [18] previously used Eq. 5 to calculate the desired internal pressure from the measured current tension and the corresponding pressure and tension stored during the teaching phase. Applying Eq. 5 to both the master and slave symmetrically provides bilateral control of the master–slave exoskeleton. We can now show the correspondence between the two control methods proposed in Eqs. 2 and 1. To produce the desired pressure, a pressure feedback controller is needed and can be described as follows:

$$V = K_p(P^* - P). \quad (6)$$

where V , K_p , and P^* indicate the amount of the valve opening, the positive feedback gain, and the desired pressure. A positive V indicates that the valve is opened to increase the pressure of the connected PAM, and a negative V indicates the opposite. If we employ Eq. 5 to control a PAM, P^* is replaced, and Eq. 6 is transformed into:

$$V = K_p \left(\frac{F}{\hat{F}} \hat{P} - P \right). \quad (7)$$

Equation 7 can be transformed from Eqs. 3 and 4 as follows:

$$\begin{aligned}
V &= K_p \left(\frac{\frac{P}{4\pi n^2} (3L^2 - b^2)}{\frac{\hat{P}}{4\pi n^2} (3\hat{L}^2 - b^2)} \hat{P} - P \right) \\
&= \frac{3K_p P}{3\hat{L}^2 - b^2} (L^2 - \hat{L}^2) \\
&= \frac{3K_p P \hat{P}}{4\pi n^2 \hat{F}} \left(\sqrt{\frac{4\pi n^2 F}{3P} + \frac{b^2}{3}} + \sqrt{\frac{4\pi n^2 \hat{F}}{3\hat{P}} + \frac{b^2}{3}} \right) (L - \hat{L}). \tag{8}
\end{aligned}$$

The variables P , \hat{P} , F , and \hat{F} do not take negative values, which means the coefficient of the term $L - \hat{L}$ is always positive. Positive values of V increase the pressure of a PAM and make it contract. The control based on Eq. 5 was assumed to correspond to the length feedback control of a PAM, and the feedback gain varies according to the coefficient of the term $L - \hat{L}$. In [18, 19], despite the influence of the variable feedback gain, the control based on Eq. 5 was confirmed to work appropriately in experiments. Therefore, applying Eq. 5 to both the master and slave symmetrically will result in control comparable to that shown in Eq. 2.

3.3 Relationship Between Pressures and Tensions with a Constant Amount of Internal Air

To apply the control based on Eq. 5, the axial tension information must be obtained from the PAMs. Conventional tension sensors are not fMRI-compatible, however, axial tension can be estimated from the internal pressure of the PAMs by closing the valve to prevent the airflow.

Figure 3 shows the structure of a PAM. Variables D and θ_p represent the diameter of the PAM and the angle between the braided thread and long axis, respectively. In [15], the following equation was derived from Eq. 3:

$$F = \frac{b^2(3 \cos^2 \theta_p - 1)}{4\pi n^2} P. \tag{9}$$

In Eq. 9, we cannot measure variables and constants other than the pressure P . Therefore, we simply estimated the tension F by assuming that the coefficient of P is constant. Certain cases exist for which this assumption is feasible. Maximal contraction of the PAMs occurs when the internal pressures of the PAMs used as sensors are set as high as possible. In this case, the variation in θ_p , which represents the variation of the length L , is minimized because the stiffness of the PAM is maximized. Therefore, the assumption that the coefficient of P is constant is valid when the PAMs are fully supplied and the valve is closed to prevent the airflow (Fig. 4). We empirically con-

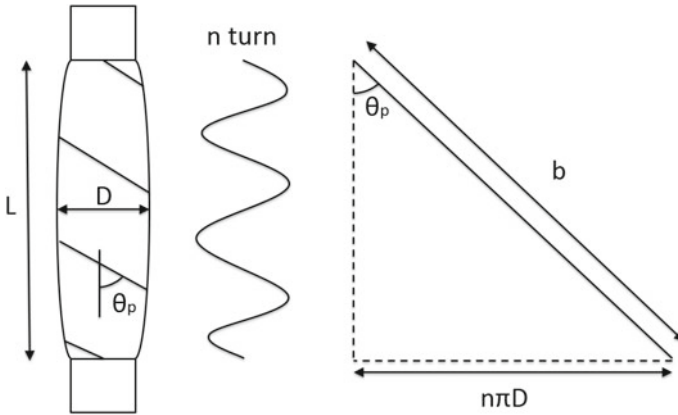
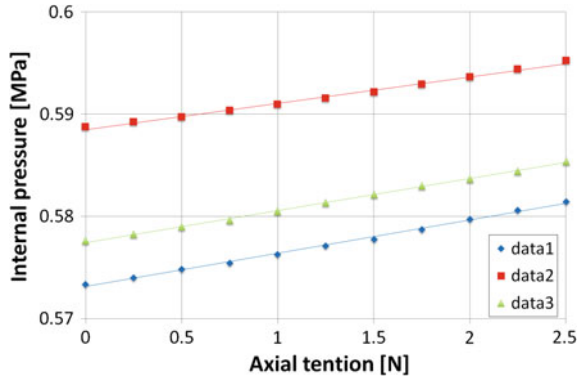


Fig. 3 Geometric structure of a PAM. The *left figure* shows the whole image. The *center figure* focuses on one of the nylon threads making up the PAM. The *right figure* shows the unfolded nylon thread

Fig. 4 Relationship between the internal pressure and axial tension when the PAMs are fully supplied and the valve is closed to prevent the airflow



firmly that θ_p is constant in this case and by extension, the coefficient $\frac{b^2(3 \cos^2 \theta_p - 1)}{4\pi n^2}$ is also constant and was approximated experimentally.

Figure 4 shows the relationship between the internal pressure and axial tension experimentally measured from three PAMs with the maximum regulated air pressure set to approximately 0.585 MPa. The relationship was approximated by a linear model in which the fully supplied sensing PAMs were connected in series with the extensor and flexor PAMs in the master and slave, as shown in Fig. 5. The variables P_{FI} , P_{FO} , \hat{P}_{FI} , and \hat{P}_{FO} are the internal pressures of the sensing PAMs, and correspond to the tension variables F_I , F_O , \hat{F}_I , and \hat{F}_O , respectively.

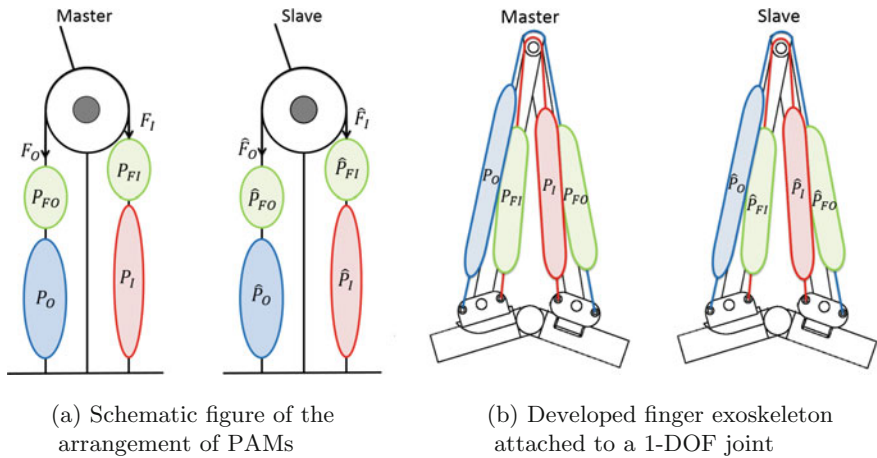


Fig. 5 Schematic figures of the developed finger exoskeleton employing PAMs measure the tensions

3.4 Derivation of the Proposed Method

We proposed a specialized, symmetric, bilateral control method for the master–slave finger exoskeleton. If the subject applies an external force f or \hat{f} to the master or slave, respectively, the following relationships are valid:

$$f = F_I - F_O, \hat{f} = \hat{F}_I - \hat{F}_O. \quad (10)$$

To describe a force applied to the master as if it were applied to the slave, the forces must observe the following relationship:

$$f = -\hat{f} \\ F_I - F_O = \hat{F}_O - \hat{F}_I. \quad (11)$$

We used this equation as a constraint condition to derive the symmetric bilateral control described in Eq. 5.

Using Eq. 5, we can derive the following equations to calculate the desired pressures of the PAMs actuating the master and slave:

$$P_I = \frac{F_I}{\hat{F}_I} \hat{P}_I, P_O = \frac{F_O}{\hat{F}_O} \hat{P}_O, \hat{P}_I = \frac{\hat{F}_I}{F_I} P_I, \hat{P}_O = \frac{\hat{F}_O}{F_O} P_O. \quad (12)$$

Introducing the constraint described in Eq. 11, the above pressures can be transformed as follows:

$$\begin{aligned} P_I &= \frac{F_O + \hat{F}_O - \hat{F}_I}{\hat{F}_I} \hat{P}_I, & P_O &= \frac{F_I + \hat{F}_I - \hat{F}_O}{\hat{F}_O} \hat{P}_O, \\ \hat{P}_I &= \frac{\hat{F}_O + F_O - F_I}{F_I} P_I, & \hat{P}_O &= \frac{\hat{F}_I + F_I - F_O}{F_O} P_O. \end{aligned} \quad (13)$$

Similarly, when forces are estimated from pressure, the desired pressures are described as follows:

$$\begin{aligned} P_I &= \frac{K_{FO} P_{FO} + \hat{K}_{FO} \hat{P}_{FO} - \hat{K}_{FI} \hat{P}_{FI}}{\hat{K}_{FI} \hat{P}_{FI}} \hat{P}_I, \\ P_O &= \frac{K_{FI} P_{FI} + \hat{K}_{FI} \hat{P}_{FI} - \hat{K}_{FO} \hat{P}_{FO}}{\hat{K}_{FO} \hat{P}_{FO}} \hat{P}_O, \\ \hat{P}_I &= \frac{\hat{K}_{FO} \hat{P}_{FO} + K_{FO} P_{FO} - K_{FI} P_{FI}}{K_{FI} P_{FI}} P_I, \\ \hat{P}_O &= \frac{\hat{K}_{FI} \hat{P}_{FI} + K_{FI} P_{FI} - K_{FO} P_{FO}}{K_{FO} P_{FO}} P_O. \end{aligned} \quad (14)$$

where the K variables are the gains of the sensing PAMs used to compute tension from pressure, with the specific muscles indicated by subscripts. In the proposed method, the desired pressures for all PAMs actuating the master or slave are computed according to Eq. 14 and used in each pressure feedback controller.

4 Experiment

4.1 Experimental Setup

The master–slave finger exoskeleton and the proposed symmetric, bilateral control method were validated experimentally. In the experiment, the master and slave were each attached to a 1-DOF joint that was equipped with a potentiometer to measure the angle. Figure 6 shows the overview of the developed system in which electro-pneumatic regulators (ITV Series, SMC Corporation) were used for PAMs actuating the mechanism, and pressure sensors (PSE series, SMC Corporation) were used for all other PAMs. A control board (Arduino Due) allowed the valves to be controlled by pulse width modulation (PWM) signals, and the sensors to be read by analog-to-digital (A/D) converters. The measured pressures were used to calculate the desired pressures based on Eq. 14, and the desired pressures were input to the valves.

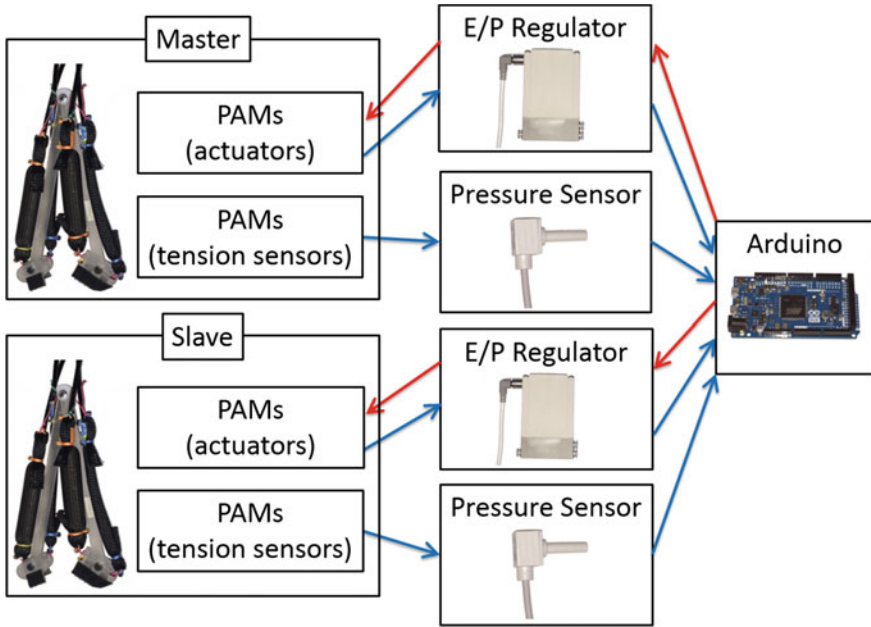
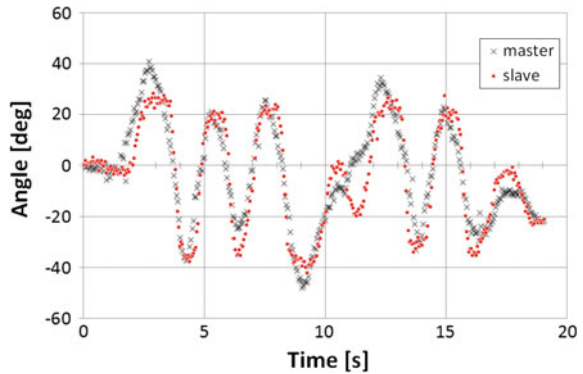


Fig. 6 Overview of the developed system

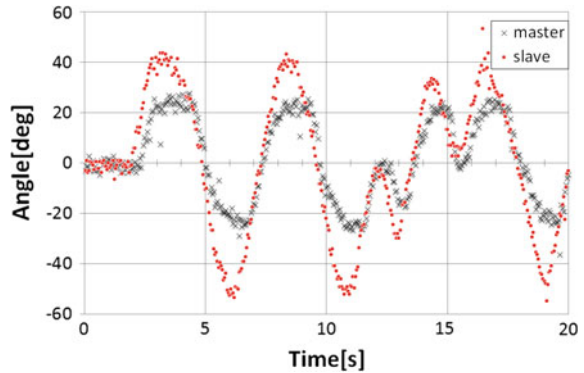
Fig. 7 Response when an external force was applied to the master side



4.2 Results

The initial joint angles of the master and slave were set to 0°. Figures 7 and 8 show the changes in the angles of the master and slave when the slave and master were manually moved. These graphs show that the angle of the master followed that of the slave and vice versa, which serves to validate the system. However, Fig. 8 shows that the movement of the master was smaller than that of the slave, although this does not appear in Fig. 7. We identified two potential causes for this issue. First, the angles

Fig. 8 Response when an external force was applied to the slave side



were not correctly measured with the potentiometers because we did not account for the linkages mounted on the 1-DOF joint, which received torques and shifted slightly during the movement. To prevent this problem in the future, the lengths of the linkages and routings of the tendons would need to be investigated. Second, the accuracy of the tension sensors using PAMs was problematic because the measurements were small and subject to noise as shown in Fig. 4. This problem may be prevented in future work if amplifiers and noise filters are used.

5 Conclusion and Future Works

We developed a master–slave finger exoskeleton driven by PAMs that can be used under a strong magnetic field owing to its lack of metallic components. Additionally, we proposed a specialized, symmetric, bilateral control method for the device that uses only the internal pressure information of PAMs. The system was successfully validated by experimental testing and we proposed potential solutions to remaining problems as future work.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number 26700025, 23220004, 24000012, 15H01665.

References

1. Craig, D., Takahashi, Der-Yeghian, L., Le, V., Motiwala, R.R., Cramer, S.C.: Robot-based hand motor therapy after stroke. *Brain* **131**(2), 425–437 (2008)
2. Khanicheh, A., Muto, A., Triantafyllou, C., Weinberg, B., Astrakas, L., Tzika, A., Mavroidis, C.: fMRI-compatible rehabilitation hand device. *J. Neuroeng. Rehabil.* **3**(1), 1 (2006)
3. Braadbaart, L., Buchan, G., Williams, J.H.G., Waite, G.D.: An fMRI compatible touchscreen to measure hand kinematics during a complex drawing task. *Br. J. Appl. Sci. Technol.* (2015)

4. Sergi, F., Erwin, A.C., O'Malley, M.K.: Interaction control capabilities of an MR-compatible compliant actuator for wrist sensorimotor protocols during fMRI. *IEEE/ASME Trans. Mechatron.* **20**(6), 2678–2690 (2015)
5. Wege, A., Hommel, G.: Development and control of a hand exoskeleton for rehabilitation of hand injuries. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3046–3051. IEEE (2005)
6. Fu, Y., Wang, P., Wang, S.: Development of a multi-DOF exoskeleton based machine for injured fingers. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1946–1951. IEEE (2008)
7. Yamaura, H., Matsushita, K., Kato, R., Yokoi, H.: Development of hand rehabilitation system using wire-driven link mechanism for paralysis patients. In: *IEEE International Conference on Robotics and Biomimetics*, pp. 209–214. IEEE (2009)
8. Hioki, M., Kawasaki, H., Sakaeda, H., Nishimoto, Y., Mouri, T.: Finger rehabilitation support system using a multifingered haptic interface controlled by a surface electromyogram. *J. Robot.* **2011** (2011)
9. Noritsugu, T., Yamamoto, H., Sasaki, D., Takaiwa, M.: Wearable power assist device for hand grasping using pneumatic artificial rubber muscle. In: *SICE Annual Conference*, vol. 1, p. 420. SICE (2004)
10. Sun, Z., Bao, G., Li, X., Lu, J.: Angle measurement and calibration of force feedback dataglove. In: *Proceedings of the JFPS International Symposium on Fluid Power*, vol. 2008, pp. 483–488. The Japan Fluid Power System Society (2008)
11. Polygerinos, P., Lyne, S., Wang, Z., Nicolini, L.F., Mosadegh, B., Whitesides, G.M., Walsh, C.J.: Towards a soft pneumatic glove for hand rehabilitation. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1512–1517. IEEE (2013)
12. Teramae, T., Noda, T., Morimoto, J.: Optimal control approach for pneumatic artificial muscle with using pressure-force conversion model. In: *IEEE International Conference on Robotics and Automation*, pp. 4792–4797. IEEE (2014)
13. Ikeda, T., Matsushita, A., Saotome, K., Hasegawa, Y., Sankai, Y.: Pilot study of floor-reactive-force generator mounted on MRI compatible lower-extremity motion simulator. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 311–316. IEEE (2012)
14. Tondur, B., Lopez, P.: Modeling and control of McKibben artificial muscle robot actuators. *IEEE Control Syst. Mag.* **20**(2), 15–38 (2000)
15. Chou, C.-P., Hannaford, B.: Measurement and modeling of McKibben pneumatic artificial muscles. *IEEE Trans. Robot. Autom.* **12**(1), 90–102 (1996)
16. Doumit, M., Fahim, A., Munro, M.: Analytical modeling and experimental validation of the braided pneumatic muscle. *IEEE Trans. Robot.* **25**(6), 1282–1291 (2009)
17. Kogiso, K., Sawano, K., Itto, T., Sugimoto, K.: Identification procedure for McKibben pneumatic artificial muscle systems. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3714–3721. IEEE (2012)
18. Ikemoto, S., Nishigori, Y., Hosoda, K.: Direct teaching method for musculoskeletal robots driven by pneumatic artificial muscles. In: *IEEE International Conference on Robotics and Automation*, pp. 3185–3191. IEEE (2012)
19. Ikemoto, S., Kayano, Y., Hosoda, K.: Active behavior of musculoskeletal robot arms driven by pneumatic artificial muscles to effectively receive human's direct teaching. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4612–4617. IEEE (2014)

Temporal Structure of Muscle Synergy of Human Stepping Leg During Sit-to-Walk Motion

Qi An, Hiroshi Yamakawa, Atsushi Yamashita and Hajime Asama

Abstract In daily lives, humans successfully transit their motions rather than performing separate movements. It has been widely acknowledged that there are four and five modules (called muscle synergy) in human sit-to-stand and walking motions, but it was still unclear how humans activate their redundant muscles to transit their movement from sitting to walking. Therefore this study hypothesize that human sit-to-stand can be explained from muscle synergies of sit-to-stand and walking motions, and we perform the experiment to verify it. Firstly, four and five muscle synergies were obtained from sit-to-stand and walking motion, and it has been tested whether these nine synergies are applicable to sit-to-walk motion. Results showed that sit-to-walk motion were successfully explained from nine synergies. Moreover, it was shown that humans adaptively changed the activation time of each synergies to delay body extension time and to generate necessary initial momentum for the walking motion.

Keywords Sit-to-walk · Muscle synergy · Motion analysis

1 Introduction

Aging society has become a serious issue these days, and many elderly people have been suffering from declined physical ability. This situation increases the social security cost and it also becomes a big burden to care givers. In order to solve these problems, it is important to firstly evaluate human body function and to develop a training methodology for the impaired mobility. However, it has not been clear how humans achieve their movement. Particularly it has been widely acknowledged that human

Q. An (✉) · H. Yamakawa · A. Yamashita · H. Asama
The University of Tokyo, Tokyo 1138656, Japan
e-mail: anqi@robot.t.u-tokyo.ac.jp
URL: <http://www.robot.t.u-tokyo.ac.jp/asamalab/en/>

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_8

body is a redundant system that the number of the muscles is much larger than that of the joints. To develop the efficient training protocol, it is necessary to know how humans control this redundant degrees of freedom.

To this end, our research group has employed the idea of muscle synergies. The muscle synergy was firstly proposed by Bernstein which stated that human body was controlled by the small number of modules (called synergies) rather than activating individual muscles [1]. In our previous study, we have been focusing on human sit-to-stand (STS) motion and we found that four muscle synergies could explain the motion [2]. Similarly it was suggested that human locomotion could be accounted mainly by five modules [3]. Another study investigated the muscle activation of the spinal cord injury patients and they found that muscle synergy structure was quite different from the healthy ones [4]. These previous findings implied that humans could utilize muscle synergies to achieve the motion and the movement would be disturbed with the impaired synergies. These findings also suggest the possibility which muscle synergies as an evaluation markers for the impaired function.

These studies, including our research, did investigate STS and walk motions in different conditions. Although these findings are limited to the single motion, humans usually have to combine the several motions or to transit one motion to another in daily lives rather than performing a individual movement. In particular, humans do not only stand up from a chair, but this movement will lead to the locomotion. Although previous studies suggested that four and five muscle synergies could be used for human STS and walking motions, it is unclear whether humans utilize the same module to transit their movements.

In fact, many medical and rehabilitation hospitals use this sit-to-walk (STW) motion in Timed Up and Go (TUG) test to evaluate body function of the elderly [5]. The test measures the time from when people stand up from a chair until they come back and sit down on the chair again after 6 m walk. Although this test only measured the time to perform the motions, it was known that the taken time is strongly co-related to the dynamic stability of the elderly. If we could clarify the necessary muscle activation structure in STW motion, it would provide more evidence for the TUG test.

There are a few studies which analyzed STW movements. One previous study analyzed body trajectory and reaction force during STW and they found that the young persons could transit the motions more smoothly than the elderly by generating more initial momentum [6]. Another research group defined four characteristic event also based on kinematics and reaction force [7, 8]. Other researchers investigated muscle activity, kinematics and reaction force to find activation order of each muscle and also revealed the relationship between the knee torque and hip flexion [9]. It has been also emphasized that there were merging of two tasks in STW motion rather than performing separate tasks [10]. These previous studies showed that STW was not achieved only by serial arrangement of two tasks but there was a fusion of two tasks. However, they mainly considered kinematic movement or center of pressure and they have not yet clarified how they controlled redundant muscles.

Therefore, this study analyzes human STW in terms of their muscle activity. In particular we hypothesize that modular organization (muscle synergy) of STS and walking can account for STW movement. Moreover, it will be investigated how these modules are adaptively coordinated in order to achieve STW motion.

2 Methods

2.1 Muscle Synergy Model

This study analyzes human motion based on the muscle synergy model. Firstly muscle synergy model is explained in detail. Muscle synergy model assumes that various human movements can be explained by the limited number of the modules. Particularly it supposes that these module are composed of synchronized muscle activation and humans control their weight to properly achieve different motion. In a mathematical expression, it is often expressed that muscle activation could be generated from linear summation of spatiotemporal patterns as in following equations,

$$\mathbf{M} = \mathbf{W}\mathbf{C}, \quad (1)$$

where matrix $\mathbf{M} \in \mathbb{R}^{n \times t_{\max}}$ indicates muscle activation. Each row of the matrix $\mathbf{m}_{i=1 \dots n}$ represents time series of n different muscle activation (Eq. 2). The matrix $\mathbf{W} \in \mathbb{R}^{n \times k}$ shows spatial patterns which indicates relative activation level of each muscle, and each column of the matrix $\mathbf{w}_{i=1 \dots k}$ represents separate modules (Eq. 3). The matrix $\mathbf{C} \in \mathbb{R}^{k \times t_{\max}}$ represents temporal patterns which is time-varying weighting coefficient of the muscle synergies, and the row of the matrix $\mathbf{c}_{i=1 \dots k}$ shows different temporal patterns (Eq. 4).

$$\mathbf{M} = \begin{pmatrix} \mathbf{m}_1(t) \\ \mathbf{m}_2(t) \\ \vdots \\ \mathbf{m}_n(t) \end{pmatrix} = \begin{pmatrix} m_1(1) & \cdots & m_1(t_{\max}) \\ \vdots & \ddots & \vdots \\ m_n(1) & \cdots & m_n(t_{\max}) \end{pmatrix}, \quad (2)$$

$$\mathbf{W} = (\mathbf{w}_1 \cdots \mathbf{w}_N) = \begin{pmatrix} w_{11} & \cdots & w_{1k} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nk} \end{pmatrix}, \quad (3)$$

$$\mathbf{C} = \begin{pmatrix} \mathbf{c}_1(t) \\ \mathbf{c}_2(t) \\ \vdots \\ \mathbf{c}_N(t) \end{pmatrix} = \begin{pmatrix} c_1(1) & \cdots & c_1(t_{\max}) \\ \vdots & \ddots & \vdots \\ c_k(1) & \cdots & c_k(t_{\max}) \end{pmatrix}. \quad (4)$$

Figure 1 shows the concept of the muscle synergy model; vertical bars in each square indicate spatial patterns (the left side of Fig. 1a), the solid line, the dashed

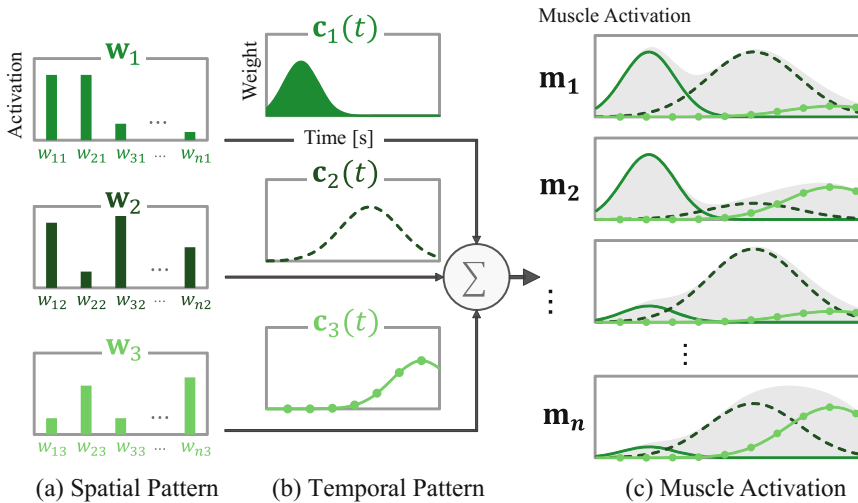


Fig. 1 Muscle Synergy model. **a** shows spatial patterns ($w_{1,2,3}$) which indicates relative excitation level of each muscle. **b** shows temporal patterns ($c_{1,2,3}$) to define time-varying weighting coefficient of corresponded muscle synergies. **c** shows time-varying activation for n muscles (gray part). Red, blue, and green dashed lines show generated activation from muscle synergies 1, 2, and 3 respectively

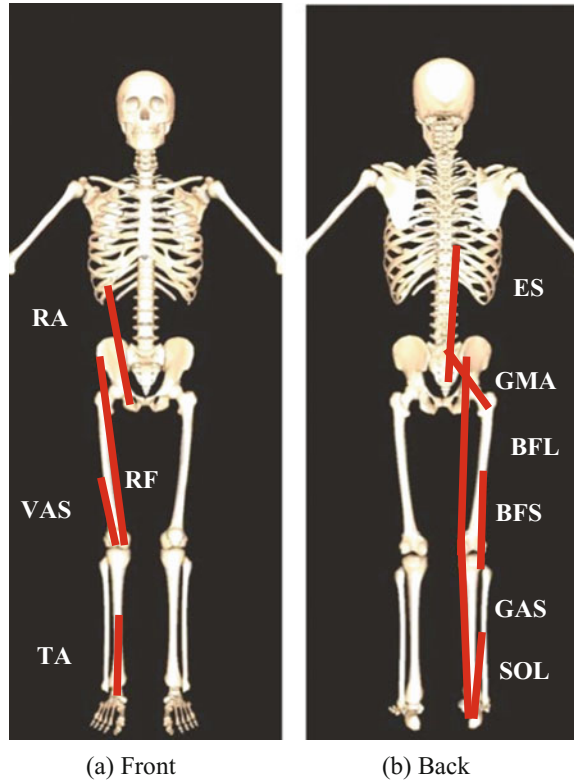
line, the line with circles represent temporal patterns corresponded to each spatial pattern (the middle side of Fig. 1b), and linear summation of these spatiotemporal patterns generates muscle activation (the right side of Fig. 1c). In order to calculate spatiotemporal patterns \mathbf{W} and \mathbf{C} of muscle synergies, non-negative matrix factorization (NNMF) [11] needs to be applied to muscle activation data \mathbf{M} . When using NNMF, the number of muscle synergies need to be determined beforehand, and it is decided as four for STS motion [2] and as five for walking motion [3] based on previous research. In this study, muscle activation ($\mathbf{M}^{\text{STS,Walk,STW}}$) during each condition is obtained in the measurement experiment.

In order to validate our hypothesis that muscle activity during STW could be explained from muscle synergies of STS and walking, we firstly extract spatiotemporal patterns (\mathbf{W}^{STS} , \mathbf{C}^{STS} , \mathbf{W}^{Walk} and \mathbf{C}^{Walk}) of muscle synergies from STS and walking motions respectively. Next, we use the extracted spatial patterns (\mathbf{W}^{STS} and \mathbf{W}^{Walk}) to investigate whether these patterns could explain muscle activation during STW motion well enough. This could be achieved by the optimization methodology [12] to find the optimal temporal patterns \mathbf{C}^{STW} to minimize following squared error z ,

$$z = |\mathbf{M}^{\text{STW}} - \mathbf{W}^{\text{STS,Walk}} \mathbf{C}^{\text{STW}}| \quad \text{when given } \mathbf{M}^{\text{STW}} \text{ and } \mathbf{W}^{\text{STW}}, \quad (5)$$

where \mathbf{M}^{STW} is muscle activation during STW motion and \mathbf{W}^{STW} is composed of \mathbf{W}^{STS} and \mathbf{W}^{Walk} . To evaluate how well the spatial patterns during STS and walking explain muscle activation of STW, coefficient of determination R^2 is calculated.

Fig. 2 Considered muscles. **a** and **b** respectively show measured muscles from front and back views. Ten muscles are considered which either flex or extend ankle, knee, hip and lumbar joints

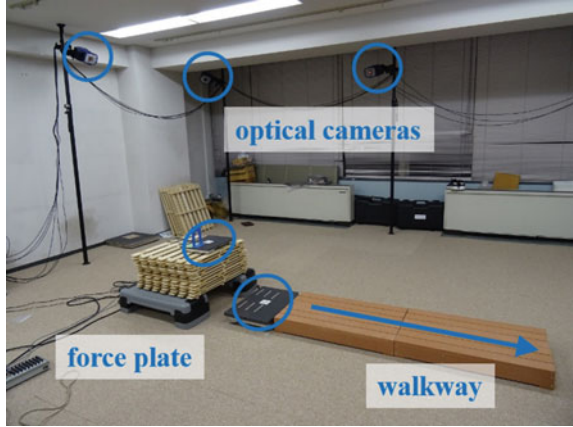


The most moved joints during STS, walking and STW are ankle, knee, hip and lumbar. Therefore, this study particularly investigates major ten muscles which could account for both flexion and extension movement of these joints. Considered muscles are as follows; tibialis anterior (TA), gastrocnemius (GAS), soleus (SOL), rectus femoris (RF), vastus lateralis (VAS), biceps femoris long head (BFL), biceps femoris short head (BFS), gluteus maximus (GMAS), rectus abdominis (RA), erector spine (ES). All the muscles are shown in Fig. 2.

2.2 Experiment

Measurement experiment was conducted to record body trajectory, reaction force and muscle activity during STS, normal walking and STW motions. Figure 3 shows our experimental setup.

Fig. 3 Experimental setup. Eight optical cameras were used to measure the body trajectory and two force plates were placed below the feet and hip to record reaction force. Wood blocks were put in front of the force plate to ensure that participants could perform STW motion continuously



2.2.1 Experimental Setup

Eight motion capture cameras (Raptor-H; Motion Analysis Corp.) were used to measure the body trajectory in 100 Hz. Reflective markers were attached to the participant body based on the Helen Hayes marker set. Reaction force from feet and hip was measured in 1,000 Hz by two forceplates (TF-4060 and TF-3040; TechGihan Co., Ltd.). Figure 3 shows our experimental setup. Muscle activation was measured in 1,000 Hz from surface electromyography sensor (DL-141 and DL-721; S&ME, Inc.). EMG sensors were attached to the right leg of the participants. In order to standardize the muscle activity among different trials and participants, it was normalized to 0–1 based on the maximum and minimum values in every trial. All the recorded data was filtered; body position data was filtered with the second order butter worth low pass filter with 5 Hz, reaction force data was low pass filtered with 20 Hz, and muscle activation data was filtered with the band pass filter of 60–200 Hz. Recording time for each trial was 10 s.

2.2.2 Experimental Condition

Three young male (23.7 ± 0.6 years) participated at our experiment. Firstly, participants were asked to stand up from the seated position in the comfortable speed. The chair height was adjusted to the knee height of each subject. Also they were told to cross their arms in front of their chest in order to avoid usage of the arms. Next, subjects were asked to perform STW motion. They were instructed to stand up and to transit continuously to the walking motion. The participants were asked to perform locomotion on the 1.8 m walkway as shown in Fig. 3. All of them was told to take the initial step from the right leg, and they stop walking until they took three steps. At last, participants were asked to perform walking motion from the standing posture.

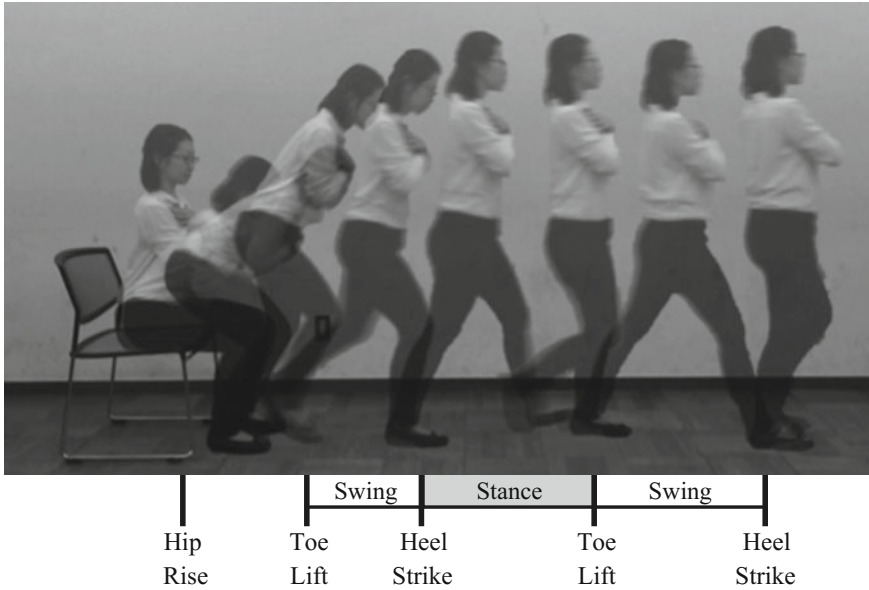


Fig. 4 Kinematic event during STW Motion. Above figure shows STW movement and the vertical lines show characteristic event such as hip rise, toe lift, heel strike. Squares show duration of swing and stance phase during walking

In this case, the subjects also started their motion from the right side and performed three steps. Fifteen trials were obtained from every condition.

Figure 4 shows kinematic movements during STW motion. The participants firstly rise their hip and they initiate the first step from the right leg. Some characteristic events are depicted in the vertical lines such as hip rise, toe lift and heel strike. White and gray squares between vertical lines show whether the leg is in swing or stance phases. These events could be obtained from measured kinematic and force data. The time of hip rise was determined when the vertical reaction force of the hip became less than 5.0 N. The toe lift timing was decided when the vertical velocity of the toe became positive. The heel strike time was decided when the vertical velocity of the heel became larger than -0.015 m/s after its deceleration.

Duration of all the trials were normalized in order to compare different conditions. Movement time of STS was decided based on characteristic kinematic movement. The start time of STS was when the participants started to bend their trunk. The end time for the motion could not be determined clearly since humans only stand straight after their standing-up. Therefore, we firstly calculate the time between the start time and the time when the participants reached the highest shoulder position. Next, the end time of STS was decided as it was 125 % of the period between the first bending and the straight standing. Duration time of walking condition was decided to the period from the first heel strike to the next heel strike. In the experiment, one cycle of walking motion was used from each trial.

3 Results

From measured muscle activation, four and five muscle synergies were extracted for STS and walking motions respectively. Using these extracted nine modules ($w_{1...4}^{STS}$ and $w_{1...5}^{Walk}$), the temporal patterns were obtained through the optimization methodology. The coefficient of determination R^2 was 0.88 ± 0.02 when muscle activation of STW was explained by modules of STS and walking.

Figure 5 shows spatiotemporal patterns during STS and STW motions. The left graphs (Fig. 5a) show mean and standard deviation of the spatial patterns during STS movement. In the middle (Fig. 5b), these stick pictures indicate corresponded movement. The arrows indicate the directions of the joints to be moved. The right graphs in Fig. 5c show the temporal patterns compared between STW and STS motions.

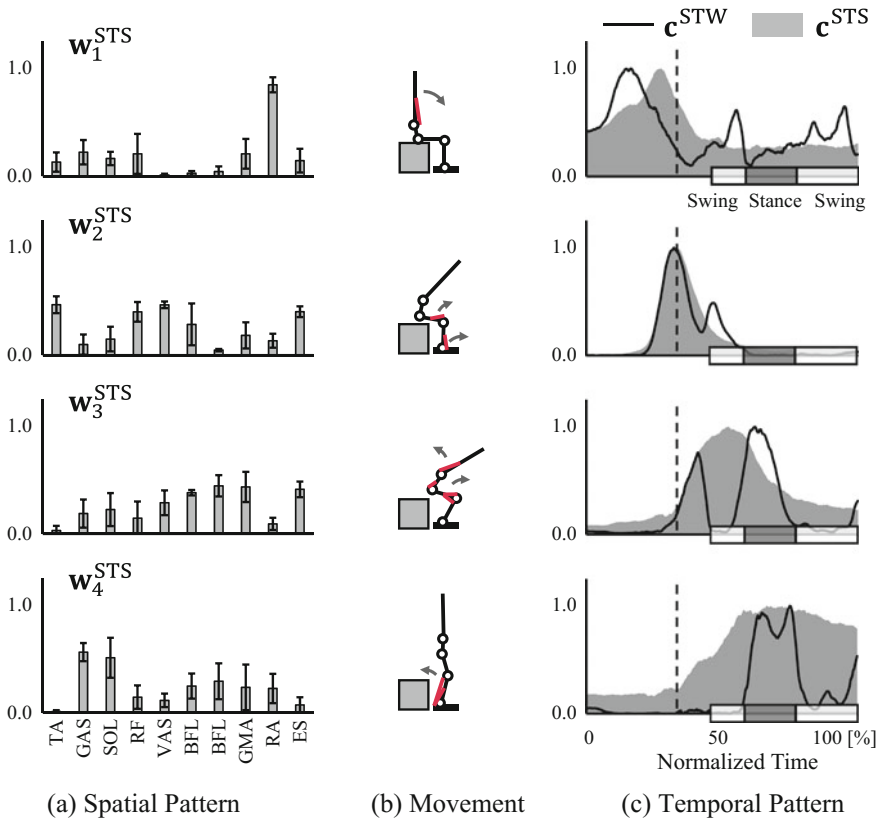


Fig. 5 Muscle synergy results in STS and STW motion. **a** Spatial patterns of extracted synergies are shown above. Four synergies are obtained from STS motion. **b** These stick pictures show corresponded movements to each spatial pattern. *Red lines* represent mainly activated muscle. **c** Above graphs show the temporal pattern of STW (*solid lines*) and STS motions (*colored area*)

Temporal patterns of STS are shown in colored filled area and ones of STW are shown in solid lines. The vertical lines represent the time of hip rise and squares on the horizontal axis represent duration of swing and stance phases. Mainly activated muscles in each spatial patterns had the same characteristics as the ones reported in the previous study [2]. The first module (w_1^{STS}) in STS motion activated RA to bend their trunk forward. The second module (w_2^{STS}) activated TA, VAS and RF to dorsiflex the ankle and extend the knee to rise from the chair. The third module (w_3^{STS})

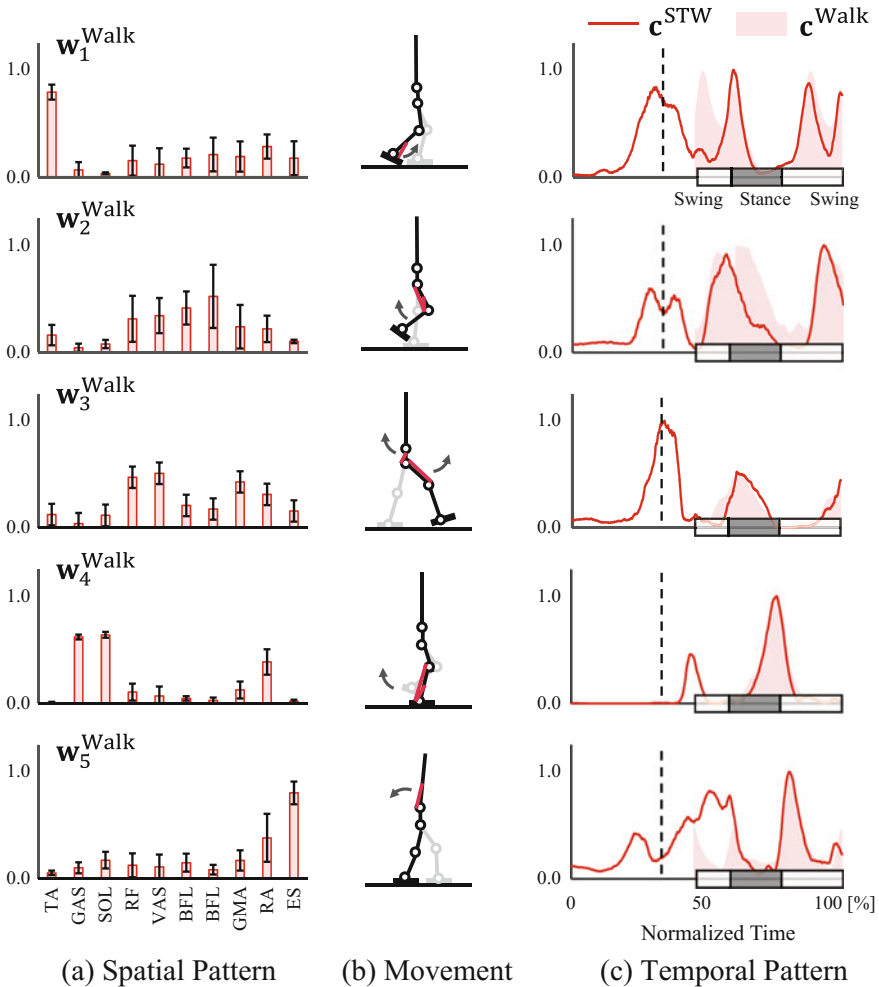


Fig. 6 Muscle synergy results in walking and STW motion. **a** Spatial patterns of extracted synergies are shown above. Five synergies are obtained from walking motion. **b** These stick pictures show corresponded movements to each spatial pattern. *Red lines* represent mainly activated muscle. **c** Above graphs show the temporal pattern of STW (*solid lines*) and walking motions (*colored area*)

activated VAS, RF, BFL, BFS and ES to extend the knee and the lumbar joints. The last module ($\mathbf{w}_4^{\text{STS}}$) activated SOL and GAS to plantarflex the ankle joint to stabilize their posture.

Similarly, Fig. 6 shows the spatial patterns in walking and STW motions (Fig. 6a), corresponded kinematics (Fig. 6b), and temporal patterns during walking and STW motions (Fig. 6c). Spatial patterns in walking motion also have the similar characteristics. The first module ($\mathbf{w}_1^{\text{Walk}}$) activated TA to plantarflex the ankle to control the feet during the swing phase. The second module ($\mathbf{w}_2^{\text{Walk}}$) activated BFL and BFS to pull the knee to avoid their shank to hit the ground. The third module ($\mathbf{w}_3^{\text{Walk}}$) activated VAS, RF and GMA to extend the knee and the hip to absorb impact force of heel strike. The fourth module ($\mathbf{w}_4^{\text{Walk}}$) activated SOL and GAS to dorsiflex the ankle to move the body forward. The last module ($\mathbf{w}_5^{\text{Walk}}$) activated ES to extend the lumber to control upper body.

4 Discussion

Our results showed that muscle synergies of STS and walking motions could explain the most part of the muscle activation during STW motion (88 %). This implied that humans did not need additional modules while they transited the motion from standing-up to walking. Although the same modules were applicable to the STW motion, their activation profiles (temporal patterns) were different from their initial motions (STS and walking).

The temporal pattern of the first module of STS ($\mathbf{w}_1^{\text{STS}}$) did not differ a lot for STW ($\mathbf{c}_1^{\text{STW}}$) from the initial temporal pattern of STS ($\mathbf{c}_1^{\text{STS}}$). On the contrary, other three modules of STS had significant difference compared to the temporal patterns of STS. Temporal pattern of the second module for STW ($\mathbf{w}_2^{\text{STS}}$) had one additional peak compared to the one of STS ($\mathbf{c}_2^{\text{STS}}$). Focusing on the second peak, it was activated at the time of toe lift, and this implied that the second module was also utilized for the different movement (lifting up their toe). The third module ($\mathbf{w}_3^{\text{STS}}$) had a similar activation profile until the participants lifted up their toe. However, the activation disappeared during the early swing phase but it would be re-activated toward the stance phase. Considering the original contribution of the third module (body extension), this phenomenon implied that humans did not fully extend their upper body, but they instead did so during the stance phase. Similarly, the fourth module ($\mathbf{w}_4^{\text{STS}}$) did not activate at all in STW at the time when it used to be activated in STS. Since, the last module contributed to posture stabilization by plantarflex the ankle joint, humans did not need this movement when they transit the motion from standing-up to walking. Instead, it was activated at the stance phase after the participants started walking. These results indicated that the same modules of STS could be utilized for STW motion, but one of the modules ($\mathbf{w}_2^{\text{STS}}$) were activated in different

ways to serve alternative movement of lifting up toes or some of them ($\mathbf{w}_3^{\text{STS}}$ and $\mathbf{w}_4^{\text{STS}}$) had the shifted activation peak.

The modules of walking were also utilized differently in STW motion. One of the findings was that in STW humans activated their modules very similarly to the one in walking motion from the time of the first heel strike. From Fig. 6c, it could be found that temporal patterns of STW ($\mathbf{C}_{1...5}^{\text{STW}}$) had very similar peak as the ones of walking ($\mathbf{w}_{1...5}^{\text{Walk}}$). Since walking motion required phasic activation of each module [3], this indicated that the participants initialized the walking program from the first step even in STW motion. Moreover three modules ($\mathbf{w}_{1,2,3}^{\text{Walk}}$) needed to be activated before starting lifting up the toe. In particular, they had a peak at the time of hip rise. This phenomenon implied that humans needed additional activation to activate TA, BFL, RF, VAS and GMA to generate more momentum to transit the motions.

Based on the above findings, it could also be implied that humans could utilize both modular organization of STS and walking at the same time. This results corresponded to the previous study [10] to suggest that there was merging of two task in transition phase. Moreover our study could reveal how humans utilized their individual modules of STS and walking motions in STW.

5 Conclusion

This study hypothesized that human transit the motion from sitting to walking could be accounted by the same muscle synergies of sit-to-stand and walking motions, and we performed the measurement experiment to verify it. Although the same muscle synergies were utilized, their activation profiles were different. One of the muscle synergies in sit-to-stand motion was also used for lifting up the toe besides rising the hip. In addition, two muscle synergies were activated in the latter time to shift the time of body extension and posture stabilization. On the other hand, humans could successfully start five modules of walking motion from the first step in order to initiate the motion. However, three modules were activated in the transit phase to generate initial momentum necessary for the walking motion.

Our future direction will be investigation of the elderly or the impaired persons, such as Parkinson disease. It is known that those who have Parkinson disease could not initiate the locomotion well. Also, it could be implied that those who had lower scores in timed up and go test cannot activate some of the modules correctly. Investigating these population will clarify the impaired body functions.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number 15K20956, 26120005, CASIO Science Promotion Foundation and JST RISTEX Service Science, Solutions and Foundation Integrated Research Program.

References

1. Bernstein, N.: *The Co-ordination and Regulation of Movement*. Pergamon, Oxford (1967)
2. An, Q., Ishikawa, Y., Aoi, S., Funato, T., Oka, H., Yamakawa, H., Yamashita, A., Asama, H.: Analysis of muscle synergy contribution on human standing-up motion using human neuromusculoskeletal model. In: *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA2015)*, pp. 5885–5890 (2015)
3. Ivanenko, Y.P., Poppele, R.E., Lacquaniti, F.: Five basic muscle activation patterns account for muscle activity during human locomotion. *J. Physiol.* **556**, 267–282 (2004)
4. Cheung, V.C.K., Turolla, A., Agostini, M., Silvoni, S., Bennis, C., Kasi, P., Paganoni, S., Bonato, P., Bizzi, E.: Muscle synergy patterns as physiological markers of motor cortical damage. *Proc. Natl. Acad. Sci. USA* **109**, 14652–14656 (2012)
5. Podsiadlo, D., Richardson, S.: The Timed Up & Go: a test of basic functional mobility for frail elderly persons. *J. Am. Geriatrics Soc.* **39**, 142–148 (1991)
6. Buckley, T., Pitsikoulis, C., Barthelemy, E., Hass, C.J.: Age impairs sit-to-walk motor performance. *J. Biomech.* **42**, 2318–2322 (2009)
7. Kerr, A., Durward, B., Kerr, K.M.: Defining phases for the sit-to-walk movement. *Clin. Biomech.* **19**, 385–390 (2004)
8. Kerr, A., Rafferty, D., Kerr, K.M., Durward, B.: Timing phases of the sit-to-walk movement: validity of a clinical test. *Gait Posture* **26**, 11–16 (2007)
9. Dehail, P., Bestaven, E., Muller, F., Mallet, A., Robert, B., Bourdel-Marchasson, I., Petit, J.: Kinematic and electromyographic analysis of rising from a chair during a sit-to-walk task in elderly subjects: role of strength. *Clin. Biomech.* **22**, 1096–1103 (2007)
10. Magnan, A., McFaden, B.J., St-Vincent, G.: Modification of the sit-to-stand task with the addition of gait initiation. *Gait Posture* **4**, 232–241 (1996)
11. Lee, D.D., Seun, H.S.: Learning the parts of objects by non-negative matrix factorization. *Nature* **401**, 788–791 (1999)
12. Clark, D.J., Ting, L.H., Zajac, F.E., Neptune, R.R., Kautz, S.A.: Merging of healthy motor modules predicts reduced locomotor performance and muscle coordination complexity post stroke. *J. Neurophysiol.* **103**, 844–857 (2010)

Part II

Field Robot

Fast and Accurate Crop and Weed Identification with Summarized Train Sets for Precision Agriculture

Ciro Potena, Daniele Nardi and Alberto Pretto

Abstract In this paper we present a perception system for agriculture robotics that enables an unmanned ground vehicle (UGV) equipped with a multi spectral camera to automatically perform the crop/weed detection and classification tasks in real-time. Our approach exploits a pipeline that includes two different convolutional neural networks (CNNs) applied to the input RGB+near infra-red (NIR) images. A lightweight CNN is used to perform a fast and robust, pixel-wise, binary image segmentation, in order to extract the pixels that represent projections of 3D points that belong to green vegetation. A deeper CNN is then used to classify the extracted pixels between the crop and weed classes. A further important contribution of this work is a novel unsupervised dataset summarization algorithm that automatically selects from a large dataset the most informative subsets that better describe the original one. This enables to streamline and speed-up the manual dataset labeling process, otherwise extremely time consuming, while preserving good classification performance. Experiments performed on different datasets taken from a real farm robot confirm the effectiveness of our approach.

Keywords Agriculture robotics · Classification · Segmentation · Convolutional neural networks

C. Potena · D. Nardi · A. Pretto (✉)
Department of Computer, Control and Management Engineering,
Sapienza University of Rome, Via Ariosto 25, 00185 Rome, Italy
e-mail: pretto@dis.uniroma1.it

D. Nardi
e-mail: nardi@dis.uniroma1.it

C. Potena
e-mail: potena@dis.uniroma1.it

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_9

1 Introduction

The application of autonomous robotics to precision agriculture is gaining a great attention in the research community, also thanks to the positive impacts that it may have in food security, sustainability and reduction of chemical treatments. In this work, we focus on applications that aim to reduce the amount of herbicides used to control weeds by means of autonomous robots that perform selective spraying or mechanical removal of accurately detected weeds. The robot in this case should autonomously detect and distinguish between crop and weeds inside the field, using only its own perception system.

In this work we present a robust and efficient weed identification system that leverages the effectiveness of convolutional neural networks (CNNs) in both the detection and classification steps. Our system takes as input 4-channels RGB+NIR images (e.g., Fig. 1b, c), provided by a multi spectral camera mounted on a farm robot (e.g., Fig. 1a) that autonomously monitors the crop and can apply selective weed treatments. The weed identification task includes, before plant classification, a plant detection step. Detection is generally a more challenging and time consuming task compared with classification, since it may require an exhaustive search in the whole image, with variable bounding boxes sizes. In the context of green plants the detection task can be simplified by exploiting the Normalized Difference Vegetation Index (NDVI) [20], extracted from the RGB+NIR images: NDVI enables to obtain

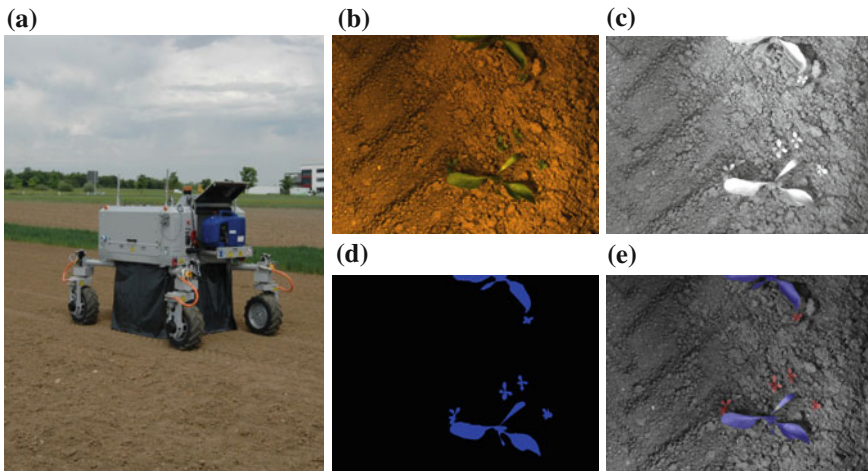


Fig. 1 **a** One of the BOSCH Bonirob employed to acquire the datasets used in the experiments; **b, c** An example of an RGB+NIR images couple provided by the multispectral camera mounted on the robot; **d** The output segmented image obtained using our vegetation detection algorithm: *blue* pixels represent projections of 3D points that belong to green vegetation; **e** The results of the proposed pixel-wise classification algorithm: pixels that belong to crop are highlighted in violet, pixels that belong to weeds are highlighted in *red*

a simple, fast and pixel-wise segmentation between green vegetation and soil (e.g., [9, 15]). Unfortunately, being threshold-based, this technique is not robust against illumination changes and different soil conditions: a careful tuning of the threshold and an outlier removal process are necessary to get a good segmentation [15]. To overcome these limitations, in this work we propose to combine the NDVI based segmentation with a trained lightweight CNN (that we call *sNet* in the following) that takes as input small patches of the RGB+NIR images. The idea is to use a very conservative threshold in order to select through the NDVI most of the true positive pixels (i.e., pixels that represent vegetation). The CNN is then used to validate each selected pixel, pruning most of the false positives (e.g., Fig. 1d). We will experimentally show that this hybrid technique outperforms the NDVI based segmentation, while preserving a good computational speed.

Pixels marked as vegetation in the segmentation step are then processed with a deeper 3-classes CNN (that we call *cNet* in the following) in order to recognize the category (crop, weeds or soil). Despite we are processing only pixels classified as vegetation in the previous step, we found that including also the class ‘soil’ in the *cNet* CNN helps to prune at no cost the remaining false positives not detected by the *sNet* CNN. In order to meet the real-time constraints required by our system, i.e. we need the classification results within one second from the image acquisition time,¹ we also propose to employ a *blob-wise* voting scheme, where blobs are connected regions extracted from the segmentation mask. We will experimentally show that: (a) our classification stage achieves state-of-the-art results; (b) the pipeline composed by the two sequential CNNs (*sNet* + *cNet*) obtains similar results if compared with a single *cNet*, but with a considerable gain in speed.

In the last part of this work we address a relatively new problem that we call *unsupervised dataset summarization*. It is well known that CNNs to be effective require large manual labeled training datasets [21]. Unfortunately, plant identification requires a challenging and extremely time consuming per-pixel labeling process. The proposed idea is to reduce the size of the dataset *before* the manual labeling stage, in order to streamline and speed-up the manual dataset labeling process while preserving good classification performance. We propose an algorithm that automatically selects a subset of K images that contain the most informative features over the N images of the whole dataset, $K \ll N$, in order to summarize in the best possible way the original dataset. The labeling process will then involve only these K images. Our features based subsets selection method is different from other max-relevance and min-redundancy feature selection methods (among others, [18, 25]) since it is *unsupervised*, i.e. it does not require the labels as input. We formulate the unsupervised dataset summarization problem as a combinatorial optimization problem, using as reward a submodular set function inspired by the coverage set functions used in text document summarization problems [14]. We will show that our dataset selection algorithm outperforms in all the experiments both the random dataset selection and the supervised manual selection strategies.

¹In our setup, one second represents a reasonable time constraint in order to enable the robot to actively remove the weeds as soon as they are detected.

2 Related Work

The problem of plant classification can be considered an instance of the so called *fine-grained visual classification* (FGVC) problem, where the purpose is for instance to distinguish between species of animals, models of cars, etc. FGVC problems are intrinsically difficult since the differences between similar categories (in our case, plant species) are often minimal, and only in recent works the researchers obtained noteworthy results (e.g., [17, 27]).

Early works in plant classification faced these problems using features extracted by co-occurrence matrices (CCM) from hue, saturation and intensity color space [22] or morphological and color features as input of a Fuzzy classifier [10]. Burks et al. [5] proposed to use CCM texture statistics as input variables for a backpropagation (BP) neural network for weed classification. Borregaard et al. [4] used two spectrometers covering both visible and near-infra-red, in order to record reflectance spectra in the wavelength range 660-1060 nm. The final crop-weed discrimination accuracy is in this case up to 90%. Feyaerts and van Gool [8] presented a performance of a classifier based on multispectral reflectance in order to distinguish the crop from weeds. The best classifier, based on neural networks, reached a classification rate of 80% for sugar beet plants and 91% for weeds.

More recently, Tellaeche et al. [24] proposed an automatic approach for detection and differential spraying of weeds. The captured images are segmented into cells, for each cell two area-based values are computed using crop, weed and soil coverage measurements, a Bayesian decision making framework is finally exploited to decide which cells have to be sprayed. Cells are also used in Aitkenhead et al. [2], where for each cell the classification is done by a pre-trained neural network. Hussin et al. [7] used shape and color features, the former extracted by the Scale Invariant Feature Transform (SIFT), the latter by the Grid Based Color Moment (GBCM). All the extracted features from the test images are then matched by Euclidean Distance with the ground truth, reaching an accuracy of 87,5%. In Haug et al. [9] a Random Forest (RF) classifier was proposed. It uses a large number of simple features extracted from a large overlapping neighborhood around sparse pixel positions. This approach achieves strong classification accuracies, due to its ability of discriminating also crops that are very similar to weeds. This approach has been improved in Lottes et al. [15] by extending the features set and including a relative plant arrangement prior that helps to obtain better classification results.

Other approaches rely on leaf classification and/or segmentation in order to detect the plant species. The leaf classification problem in complicated background has been addressed in Wang et al. [26], where leaf images are segmented using morphological operators, shape features are extracted and used in a moving center hypersphere classifier to infer plant species. Kumar et al. [12] presented an automatic plant identification application called Leafsnap: the proposed algorithm starts from segmented images of leaves and it exploits curvature features compared with a given database to extract the best match, while using a binary classifier on global image signatures as a validity test. Deformable leaf models and morphology descriptors

have been exploited in [6] to cover the variety of leaf shapes. Very recently Lee et al. [13] presented a leaf-based plant classification system that uses convolutional neural networks to automatically learn suitable visual features. Also Reyes et al. [19] used CNN for fine-grained plant classification: they used a deep CNN with the architecture proposed by Krishevsky et al. [11], first initialized to recognize 1000 categories of generic objects, then *fine-tuned* (i.e., specialized) for the specific task to recognize 1000 possible plant species.

The contribution of this work is a visual detection strategy that allows an UGV to autonomously detect crops and weeds in agricultural field environments. The proposed approach makes use of a multispectral camera as input and two CNNs to obtain accurate classification performance in different growth stages. A further contribution is our novel unsupervised dataset summarization algorithm that allows to reduce a large dataset into a smaller one with similar information properties. Selecting smaller training sets with this approach permits to boost up the labeling phase while preserving a good classification accuracy.

3 Vision-Based Plant Classification

3.1 Vegetation Detection

The goal of the vegetation detection task is to discriminate in the RGB+NIR images between pixels that represent projections of 3D points that belong to green vegetation and the other pixels. This process enables to simplify and speed up the subsequent plant detection and classification tasks.

Due to the photosynthesis, healthy green plants absorb more solar energy in the visible spectrum, causing a low reflectance level in the RGB channels. Similarly, the reflectance of the near-infra-red spectrum is affected by the same phenomena with opposite results and, as a direct consequence, with a low reflectance level in the NIR channel.

A well known indicator that is used to measure the reflectance properties of the plants is the Normalized Difference Vegetation Index (NDVI) [20], which is calculated as follows for each pixel (u, v) :

$$\mathcal{I}_{NDVI}(u, v) = \frac{\mathcal{I}_{NIR}(u, v) - \mathcal{I}_R(u, v)}{\mathcal{I}_{NIR}(u, v) + \mathcal{I}_R(u, v)} \quad (1)$$

where $\mathcal{I}_R(u, v)$ and $\mathcal{I}_{NIR}(u, v)$ stand for the spectral reflectance measurements taken from the R channel (visible red) and from the near-infrared channel, respectively. The vegetation detection task is typically solved by means of a thresholding operation on the NDVI image: a pixel (u, v) is classified as vegetation if $\mathcal{I}_{NDVI}(u, v) > th_V$, with th_V a fixed threshold. Unfortunately, a single threshold usually is not robust against

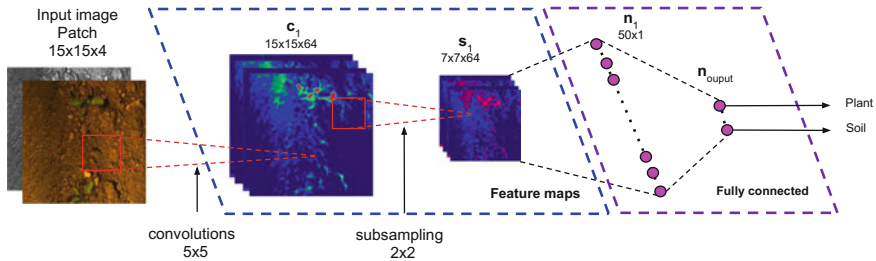


Fig. 2 Architecture of the *sNet* CNN

illumination changes and different soil conditions, even inside a single image. To address this problem, our idea is to combine the NDVI with a lightweight convolutional neural network. We first perform a thresholding operation on the NDVI using a conservative threshold, that allows to preserve most of the pixels that belong to vegetation. For each pixel classified as vegetation, we exploit a trained CNN applied to a 15×15 pixels 4 channels patch around the pixel. This network (*sNet*, Fig. 2) includes a single convolutional layer with rectified linear unit (ReLU) activation function, followed by a max pooling layer and a local response normalization step. We set both strides to 1 in the convolutional layer and both strides to 2 in the pooling layer, where a max pool operator is applied to 2×2 patches. The normalized neurons provided as output from the convolutional and pooling layers are used as inputs for a fully connected layer. The final neurons are then fully connected to the output labels ‘plant’ (i.e., vegetation) and ‘soil’ (i.e., not vegetation), that are normalized through a softmax layer. The architectural choices made for this CNN represent a good experimental trade-off between the sake of efficiency and the segmentation performance (see Sect. 5.2).

3.2 Pixel-Wise Crop/Weed Classification

The detection system described so far provides an accurate vegetation mask of the input image. Pixels that belong to vegetation need now to be classified between crop and weeds. In this plant classification task there are a lot of possible error sources, among others the similarity between plant species and the partial overlapping between different plants. In order for the network to learn more specific features that help to disambiguate in these challenging conditions, we move to a slightly deeper network with input patches of 61×61 pixel over the 4 RGB+NIR channels and, accordingly, an higher number of output neurons for every layer. The final network *cNet* (Fig. 3) includes two convolutional layers with ReLU activation function, each followed by a max pooling layer and a local normalization layer. As in the *sNet*, both the max pooling layers of *cNet* operate on 2×2 patches with strides of 2 pixels.

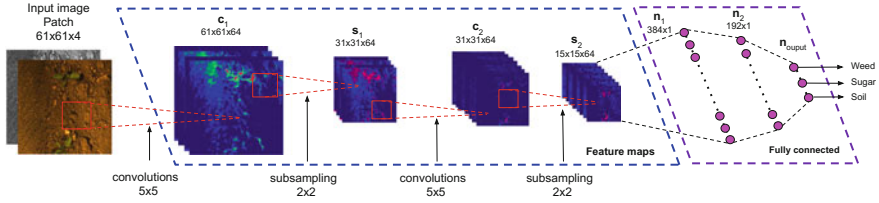


Fig. 3 Architecture of the *cNet* CNN

The normalized feature maps are then used as inputs for two fully connected layers before passing through a softmax activation function.

Algorithm 1: Blob-Wise Crop/Weed Detection and Classification

Data: The input image I and a conservative threshold th_{NDVI} for the NDVI.

Result: A set of classified blobs \mathbf{B}_c

```

/* Compute the vegetation mask  $I_v$  */
1  foreach  $(u, v) \in I$  do
2       $I_v \leftarrow$  'soil';
3      if  $I_{NDVI}(u, v) \geq th_{NDVI}$  then
4          if  $sNet(u, v) =$  'plant' then
5               $I_v \leftarrow$  'plant';
6          end if
7      end if
8  end foreach
9  Extract from  $I_v$  a set of blobs (i.e., connected regions)  $\mathbf{B} = \{b_1, \dots, b_n\}$  of pixel classified as
   'plant';
10  $\mathbf{B}_c \leftarrow \{\}$ ;
11 foreach  $b_i \in \mathbf{B}$  do
   /* We randomly sample a number of pixel from the blob,
   where  $s$  depends on the blob size  $|b_i|$  */
12 Sample  $s$  pixel  $(u, v)_j$  from the blob  $b_i$ ,  $s < |b_i|$ ;
13 Classify each pixel  $(u, v)_j$  using cNet;
14 if The majority of the  $s$  pixels have been classified as 'sugar' then
15      $\mathbf{B}_c \leftarrow \mathbf{B}_c \cup \{b_i, sugar\}$ ;
16 else if The majority of the  $s$  pixels have been classified as 'weed' then
17      $\mathbf{B}_c \leftarrow \mathbf{B}_c \cup \{b_i, weed\}$ ;
18 end if
19 end foreach

```

Despite the *cNet* processes only pixels classified as vegetation by *sNet* (i.e., pixel classified as “plant”), we still keep the class ‘soil’ as a possible output of *cNet*. We experimentally found that this helps pruning at no cost the remaining false positives not detected by the *sNet* CNN.

3.3 Blob-Wise Crop/Weed Classification

The plant detection and classification pipeline presented in the previous section provide state-of-the-art results, but it still suffers from some limitations: (a) A pure pixel-wise approach can lead to the detection of false positive plants composed by very few mis-classified pixels; (b) Differently from *sNet*, *cNet* does not meet the real-time constraints required by our system.

In order to address these problems, we propose to employ a blob-wise based voting scheme that speeds-up the processes while removing most of the small false positives plants. The pseudo-code of the proposed method is reported in Algorithm 1: we first compute the vegetation mask \mathcal{I}_v as described in Sect. 3.1 (lines 1–8), we extract all the connected regions whose pixels are classified as ‘*plant*’ (line 9) and, finally, we classify the blobs by applying *cNet* on a subset of pixels (lines 10–19): each pixel “votes” for a class, the majority decides the class of the whole blob, blobs classified as ‘*soil*’ are discarded.

4 Unsupervised Dataset Summarization

The CNNs described above should be trained using pixel-wise labeled datasets: unfortunately, pixel-wise data annotation is an extremely time consuming process, even if the user can exploit specific labeling tools that allow to quickly detect pixels belonging to vegetation by means of local thresholding operations based on NDVI. A first solution to this problem would be to extract and label only a subset of K images, randomly selected between the N images of the original dataset, $K \ll N$. Experimental evidence (Sect. 5) indicates that a randomly selected subset often does not well describe the original dataset, i.e. the subset provides a poor information “coverage” of the original dataset. Alternatively, the subset selection process could be done manually, by looking for a “good” subset of the sample images that well represent the original dataset: this strategy usually enables to obtain better classification results compared with randomly select subsets. We introduce here a simple but effective algorithm that enables to automatically select a subset of the training set that shows very good coverage properties over the original dataset. We call this problem *unsupervised dataset summarization*, where unsupervised means that the subset is extracted before the labeling process and summarization means that the subset must be very informative about the original dataset. This problem can be formulated as a special case of the *Knapsack Problem*, that given a set \mathbf{V} of N elements, each one with a given weight c_i , asks for the subset \mathbf{S}^* that maximize a *set* function $\mathcal{F} : 2^{\mathbf{V}} \rightarrow \mathbb{R}$ subject to a constraint that requires the total weight of the subset to be less or equal than a given threshold K :

$$\mathbf{S}^* = \underset{\mathbf{S} \subseteq \mathbf{V}}{\operatorname{argmax}} \mathcal{F}(\mathbf{S}) \text{ subject to } \sum_{i \in \mathbf{S}} c_i \leq K \quad (2)$$

The set function \mathcal{F} , also called *objective function*, measures the “quality” of a given subset. In our case the set \mathbf{V} is the original dataset that contains N images, the constraint is represented by an equality constraint where for each i we have $c_i = 1$, while the set function \mathcal{F} should tell us how well the subset \mathbf{S} summarizes the original dataset \mathbf{V} . It is well known that this class of problems is NP-hard, so the computation of the optimal solution \mathbf{S}^* is often not feasible. Despite that, a good approximated solution can be obtained if we provide a objective function \mathcal{F} that is *monotone submodular*. A set function \mathcal{F} is submodular if for each $\mathbf{A} \subseteq \mathbf{B} \subseteq \mathbf{V}$ and for some element $x \notin \mathbf{B}$, we have that:

$$\mathcal{F}(\mathbf{A} \cup x) - \mathcal{F}(\mathbf{A}) \geq \mathcal{F}(\mathbf{B} \cup x) - \mathcal{F}(\mathbf{B}) \quad (3)$$

A submodular set function is monotone if for each $\mathbf{A} \subseteq \mathbf{B}$ we have $\mathcal{F}(\mathbf{A}) \leq \mathcal{F}(\mathbf{B})$. Submodular functions have a very attractive property [16]: it can be proven that if \mathcal{F} is monotone submodular, then $\mathcal{F}(\hat{\mathbf{S}}) \geq \left(1 - \frac{1}{e}\right) \mathcal{F}(\mathbf{S}^*) \approx 0.632 \mathcal{F}(\mathbf{S}^*)$,² with $\hat{\mathbf{S}}$ an approximated solution computed using a greedy algorithm.

4.1 Subset Selection as a Document Summarization

Our method is inspired by the document summarization task that, given a set \mathbf{V} that contains all the sentences of a text document, searches for a subset of sentences $\mathbf{S} \subseteq \mathbf{V}$ that well represents the original document. Typically this task is subject to some constraints, such as the maximum number of words or the maximum number of sentences that compose the subset.

Let us consider a dataset acquired by a robot moving in the field as the original “document” \mathbf{V} , possibly composed by thousands of images. If we consider each image as a “sentence” of \mathbf{V} , each one composed by a set of “visual words” [23], we can reduce our problem of subset selection as a standard document summarization problem. Lin and Bilmes [14] faced the document summarization problem by proposing a class of submodular set functions that measure both the similarity of the subset \mathbf{S} to the document to be summarized (also called “coverage” of the original document) and the “diversity” of the sentences that compose the subset \mathbf{S} . Since our goal is to encourage subset \mathbf{S} that well describe \mathbf{V} , we employ as objective function a simple but in our case effective coverage set function:

$$\mathcal{L}(\mathbf{S}) = \sum_{i \in \mathbf{V}, j \in \mathbf{S}} w_{ij} \quad (4)$$

where $w_{ij} \geq 0$ represents a similarity between the image (i.e., “sentence”) i and the image j . $\mathcal{L}(\mathbf{S})$ is clearly monotone submodular.

²This is a lower bound: in most of the practical cases the approximated solution ensures much better results.

4.2 Bag-of-Visual-Words from the CNN

In the document summarization task sentences are usually represented using bag-of-terms vectors: in a similar way, we represent each image using bag-of-visual-words vectors [23]. Since the goal is to train a CNN (in our case, the CNN of Fig. 3) using a very informative subset of the original dataset, we would like to extract the visual words *directly* from the trained CNN. In a typical CNN architecture, the sequence of convolutional layers usually computes a n -dimensional vector f , used as input of a sequence of fully connected layers: the decision over the output classes depends only on f . Such a vector represents a descriptor, or *signature*, of the input image or patch. In our specific case, we apply the *cNet* of Fig. 3 to 61×61 possibly overlapping patches of the input image. After two convolutional + pooling layers (blue dotted box in Fig. 3) the patch is reduced to a 384-dimensional vector f . The idea is to represent an image as a collection of m visual words, derived from the vectors f_i , $i = 1, \dots, m$ provided by the CNN applied to m patches. If we denote with W the cardinality of our vocabulary through visual words, we can quantize the descriptors f into visual words exploiting the k-means clustering algorithm [3]. The bag-of-visual-words vector for a given image is simply the W -dimensional histogram that reports the number of times that each visual word α appears in the image.

We computed w_{ij} using the following cosine similarity:

$$w_{ij} = \frac{\sum_{\alpha \in \mathcal{S}_i} (h_{\alpha,i} \cdot h_{\alpha,j} \cdot ih_{\alpha}^2)}{\sqrt{\sum_{\alpha \in \mathcal{S}_i} (h_{\alpha,i}^2 \cdot ih_{\alpha}^2)} \sqrt{\sum_{\alpha \in \mathcal{S}_j} (h_{\alpha,j}^2 \cdot ih_{\alpha}^2)}} \quad (5)$$

where $h_{\alpha,i}$ and $h_{\alpha,j}$ are the number of times that the visual word α appears in the image, and ih_{α} is the inverse document frequency, that is calculated as the logarithm of the ratio of the number of images where α appears, over the total number of images N that compose the input dataset.

4.3 The Proposed Algorithm

The proposed method is not directly applicable: we tacitly assumed that the CNN is already able to provide valid results even if we are still training it (i.e., we are looking for a good subset of the original data set to be used for training). We solve this issue by pre-training the CNN using a general labeled auxiliary dataset or a randomly selected, manually labeled subset of the input dataset.

The pseudo-code of our Unsupervised Dataset Summarization technique is reported in Algorithm 2: we first compute the CNN descriptors from a set of patches (lines 1–4), we then extract the bag-of-visual-words vectors (lines 5–8) and finally we select the subset \mathcal{S} using a simple greedy algorithm that exploits the coverage set function reported in Eq. 4 and the similarity between images reported in Eq. 5 (lines 9–13).

5 Experimental Results

The experimental results presented this section are designed to show the accuracy of our classification system. They also confirm the performance reached by a CNN trained on a small and very representative dataset, build up by our unsupervised dataset summarization approach.

Algorithm 2: Unsupervised Dataset Summarization

Data: The input dataset \mathbf{V} with N images \mathcal{I} , the size W of the visual word vocabulary, the size K of the output subset

Result: The selected subset \mathbf{S}

```

1 foreach  $\mathcal{I} \in \mathbf{V}$  do
2   | Extract in a fixed grid a number of  $m$  patches;
3   | For each patch, compute the descriptor  $f$  provided as output of the convolutional layers
   |   of the pre-trained CNN;
4 end foreach
5 Quantize all the descriptors into  $W$  visual words using the k-means algorithm;
6 foreach  $\mathcal{I} \in \mathbf{V}$  do
7   | Compute the  $W$ -dimensional histogram that reports the numbers of times that each
   |   visual word  $\alpha$  appears in  $\mathcal{I}$ ;
8 end foreach
9  $\mathbf{S} \leftarrow \{\}$ ;
10 for  $k \leftarrow 1$  to  $K$  do
11   |  $\mathcal{I}^* \leftarrow \operatorname{argmax}_{\mathcal{I} \in \mathbf{V} \setminus \mathbf{S}} \mathcal{L}(\mathbf{S} \cup \{\mathcal{I}\})$ ;
12   |  $\mathbf{S} \leftarrow \mathbf{S} \cup \{\mathcal{I}^*\}$ ;
13 end for

```

5.1 Experimental Setup

We use two datasets, both collected from a BOSCH Bonirob farm robot (Fig. 1a) moving on a sugar beet field. Both the datasets are composed by a set of images taken by a 1296×966 pixels 4-channel JAI AD-130 camera mounted on the Bonirob. During the acquisition, the camera pointed downwards on the field and took images with a frequency of 1 Hz.

The first dataset (Dataset *A*) is composed by 700 images and it has been collected in the first growth stage of the plants, when both crop and weeds have not yet developed their complete morphological features. The second dataset (Dataset *B*) is composed by 900 images and it has been collected after 4 weeks: plants in this case are in an advanced growth stage. From each dataset we extract different subsets, each one manually labeled.

The performance of our classification approach have been measured by using two widely used metrics: the mean accuracy (MA, Eq. 6) and the mean average precision (MaP, Eq. 7):

$$MA = \frac{1}{N} \sum_{n=1}^N \frac{T_{pos} + T_{neg}}{T_{pos} + F_{pos} + T_{neg} + F_{neg}} \quad (6)$$

$$MaP = \frac{1}{Q} \sum_{q=1}^Q AP(q) \quad (7)$$

where T_{pos} and F_{pos} are the numbers of true and false positives, T_{neg} and F_{neg} are the numbers of true and false negatives, and $AP(q)$ is the average precision.

We implemented and trained the proposed CNNs *cNet* and *cNet* using the open source library TensorFlow [1].

5.2 Vegetation Detection

The first set of experiments is designed to show the performance of our vegetation detection approach that makes use of a conservative NDVI segmentation as initial pixel segmentation (see Sect. 3.1).

We train different networks in terms of amount and sizes of convolutional and fully connected layers by using the same training set taken from the dataset *A*. The results are shown in Table 1. We achieve the best mean average precision and accuracy (96.8 % and 91.3 %, respectively) with the biggest networks, composed by two convolutional and two fully connected layers. Nevertheless, our choice is to use the *sNet1c10-1f20*, being it a perfect trade-off between average time and accuracy. We compared the performance of this network with the standard NDVI based vegetation detection algorithm for some fixed thresholds (Table 2): the results of *sNet* are remarkable since it outperforms NDVI in all cases while it does not depend on any threshold.

Table 1 Vegetation detection results for different *sNet* networks. The network names follow the convention: *sNet* < *x* > *c* < *y* > - < *z* > *f* < *w* >, where *x*: number of convolutional layers, *y*: size of output feature maps, *z*: number of fully connected layers, *w*: size of the fully connected layers

Net type	MA (%)	MaP (%)	Average time (s)
sNet1c10-1f20	96.7	91.2	0.43
sNet1c5-1f10	96.6	90.8	0.34
sNet1c20-1f40	96.7	91.2	0.45
sNet2c10-2f20	96.8	91.3	1.05
sNet2c5-2f10	96.8	91.3	0.98
sNet2c20-2f40	96.8	91.3	1.23

Table 2 Comparison between the NDVI threshold based vegetation detection and the *sNet1c10-1f20*

Net type	<i>sNet</i>	NDVI ₁₆₀	NDVI ₁₇₀	NDVI ₁₈₀	NDVI ₁₉₀	NDVI ₂₀₀
Mean Accuracy	96.7 %	90.2 %	95.6 %	96.4 %	95.2 %	92.3 %

5.3 Crop/Weed Classification

In order to show the classification accuracy of our pipeline, we perform experiments for both the pixel-wise and blob-wise approaches. The results of a comparison among different networks in the case of pixel-wise classification are reported in Table 3(a). As described in Sect. 3, we use a combination of a *sNet* followed by a *cNet*. We report the average timing results for sample steps of 1 (i.e., the *cNet* is applied to

Table 3 Classification results for different *cNet* networks. The network names follow the convention: *cNet* $\langle x \rangle < c \langle y \rangle - \langle z \rangle f \langle w \rangle$, where x : number of convolutional layers, y : size of output feature maps, z : number of fully connected layers, w : size of the fully connected layers

(a) Pixel-wise and 3×3 grid classification						
Net type	MA (%)	MA _{3×3} (%)	MaP (%)	Map _{3×3} (%)	Time (s)	Time _{3×3} (s)
scNet2c64-2f192	92.3	93.3	96.2	95.6	200	23
sNet1c10-1f20 + cNet2c64-2f192	91.7	91.8	96.1	94.3	25–37	2.8–3.4
sNet1c10-1f20 + cNet2c32-2f100	90.8	90.7	95.2	94.1	22–35	2.5–3.2
sNet1c10-1f20 + cNet2c96-2f384	91.7	91.7	97.2	94.5	28–40	2.6–3.3
sNet1c10-1f20 + cNet3c64-3f192	91.8	91.8	97.4	95.7	33–48	3.6–4.5
sNet1c10-1f20 + cNet3c96-3f384	92	91.9	97.4	94.9	35–50	3.7–4.9
(b) Blob-wise classification						
Net type	MA	MaP	Time (s)			
scNet2c64-2f192	92.3	96.2	23			
sNet1c10-1f20 + cNet2c64-2f192	97.1	98.3	0.99			
sNet1c10-1f20 + cNet2c32-2f100	95.6	97.8	0.93			
sNet1c10-1f20 + cNet2c96-2f384	97.2	98.3	1.02			
sNet1c10-1f20 + cNet3c64-3f192	98	98.3	1.74			
sNet1c10-1f20 + cNet3c96-3f384	98	98.7	2.01			

each active pixel) and 3 pixels (i.e., the *cNet* is applied on a grid with spacing 3 by 3 pixels). The best trade-off in terms of accuracy, precision and computational time is obtained by the combination *sNet1c10-1f20* + *cNet2c64-2f192*, where the *cNet* is composed by four layers, equally divided into two convolutional and two fully connected layers. This network reaches a MaP of 96.1 % with a lower computational time with respect to the others. We also compare the combinations of *sNet* and *cNet* with the *cNet* network used alone. In this case the *cNet* has to be applied to the whole image, and the complete image classification is done in 23 seconds without any significant increase in precision. Examples of pixel-wise and grid classification are shown in Fig. 4a, b.

In Table 3(b) we report the classification performance obtained using our blob-wise classification algorithm (Sect. 4.3). The results are remarkable since the reported statistics refer only to the image pixels classified as vegetation by the *sNet*. We obtain these results without employing any plant position prior. Moreover, the timing results

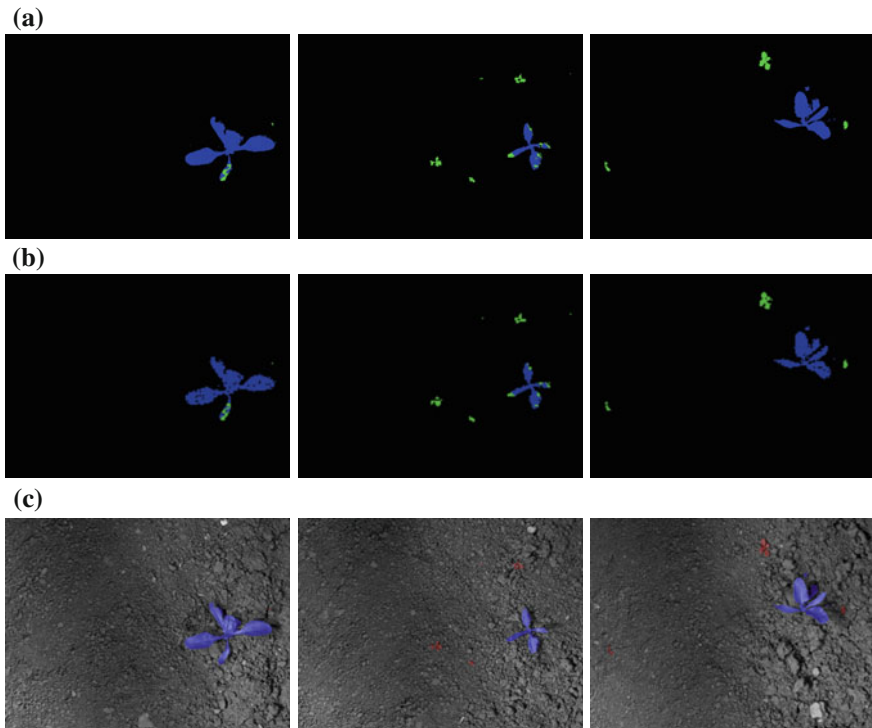


Fig. 4 **a, b** Pixel-wise and 3×3 grid based classification mask outputs from the *sNet1cm1fm* + *cNet2cm2fm* network: in *black*, *green* and *blue* are represented, respectively, pixels that belong to soil, weed and crop; **c** Final blob-wise classification outputs from *sNet1cm1fm* + *cNet2cm2fm* network: pixels that belong to crop are highlighted in violet, pixel that belong to weeds are highlighted in *red*

meet the real-time constraints required by our system. Some qualitative results are reported in Fig. 4c.

5.4 Unsupervised Dataset Summarization

We finally evaluated the performance of our unsupervised dataset summarization algorithm. We compared the pixel-wise classification results using a CNN similar to the *cNet* depicted in Fig. 3, with a descriptor size of 384 entries (i.e. the size of the first fully connected layer). We used subsets of $K = 50$ images for both dataset *A* and dataset *B*: we trained the CNN using K randomly chosen images taken only from the dataset *A* (*cNetRandomA* in Table 4) and K randomly chosen images taken only from the dataset *B* (*cNetRandomB* in Table 4). We repeated the training steps using K images manually chosen from the dataset *A* (*cNetManualA* in Table 4) and K images manually chosen from the dataset *B* (*cNetManualB* in Table 4): in both cases, we looked for subsets that well represent the original dataset. We finally trained the CNN using the automatically selected subsets obtained by applying Algorithm 2 (*cNetUdsA* and *cNetUdsB*), where we used *cNetRandomA* and *cNetRandomB* as pre-trained CNNs, respectively, and a vocabulary of $W = 4096$ visual words. We also performed a cross-validation of the trained CNNs, evaluating a CNN trained with the dataset *A* with a validation set extracted from dataset *B*, and vice versa. As shown in Table 4, the network trained by using subsets selected by our unsupervised dataset summarization algorithm outperform in all the evaluations the network trained with the manually and the randomly chosen training sets, in both datasets *A* and *B*. The relatively poor classification results (59.4%) obtained with a CNN tested with a subset of the dataset *B* and trained using samples taken from dataset *A* are due to the fact that the dataset *A* includes only plants that are in their first growth stage, thus without their complete morphological features.

Globally, our results are comparable with the ones recently reported in [15], obtained using the same datasets but, differently from [15], we *do not* exploit any row arrangement. We expect to obtain even better results by integrating also this type of information.

Table 4 Pixel-wise classification performance comparison for both datasets *A* and *B* for a *cNet* trained with different trainings sets

TrainSet & Dataset	MaP A (%)	MaP B (%)
<i>cNetRandomA</i>	94.5	57.7
<i>cNetManualA</i>	95.4	57.9
<i>cNetUdsA</i>	96.1	59.4
<i>cNetRandomB</i>	78.1	97.5
<i>cNetManualB</i>	79.1	98.6
<i>cNetUdsB</i>	82.3	99.4

6 Conclusions

In this work we addressed the problem of plant detection and crop/weed classification through a multi-spectral camera mounted on a ground robot. We leverage the effectiveness of the convolutional neural networks by proposing the following contributions: (a) A parameterless vegetation detection approach that outperforms conventional methods based on the Normalized Difference Vegetation Index (NDVI); (b) A fast classification pipeline that achieves state-of-the-art results by exploiting a sequence of a lightweight CNN followed by a deeper CNN that votes on connected vegetation blobs; (c) A dataset summarization algorithm that enables to streamline and speed-up the manual dataset labeling process while preserving good classification performance. The latter represents the main contribution of this paper. We reported detailed validations of each contribution, where we used real datasets taken from a farm robot moving in a sugar beet field. The results confirm the effectiveness of the proposed solutions.

Acknowledgments We thank Cyrill Stachniss and Philipp Lottes for providing us with the datasets used in this paper.

This work has been supported by the European Commission under the grant number H2020-ICT-644227-FLOURISH.

References

1. Abadi, M., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015). <http://tensorflow.org/>
2. Aitkenhead, M., Dalgetty, I., Mullins, C., McDonald, A., Strachan, N.: Weed and crop discrimination using image analysis and artificial intelligence methods. *Comput. Electron. Agric.* **39**(3), 157–171 (2003)
3. Bishop, C.M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc. (2006)
4. Borregaard, T., Nielsen, H., Nrgaard, L., Have, H.: Cropweed discrimination by line imaging spectroscopy. *J. Agric. Eng. Res.* **75**(4), 389–400 (2000)
5. Burks, T.F., Shearer, S.A., Gates, R.S., Donohue, K.D.: Backpropagation neural network design and evaluation for classifying weed species using color image texture. *Trans. ASAE* **43**(4), 1029–1037 (2000)
6. Cerutti, G., Tougne, L., Mille, J., Vacavant, A., Coquin, D.: A model-based approach for compound leaves understanding and identification. In: *IEEE International Conference on Image Processing*, pp. 1471–1475 (2013)
7. Che Hussin, N., Jamil, N., Nordin, S., Awang, K.: Plant species identification by using scale invariant feature transform (SIFT) and grid based colour moment (GBCM). In: *Proceedings of the IEEE Conference on Open Systems (ICOS)*, pp. 226–230 (2013)
8. Feyaerts, F., Gool, L.V.: Multi-spectral vision system for weed detection. *Pattern Recognit. Lett.* **22**(6–7), 667–674 (2001)
9. Haug, S., Michaels, A., Biber, P., Ostermann, J.: Plant classification system for crop /weed discrimination without segmentation. In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)* (2014)
10. Hemming, J., Rath, T.: PA-precision agriculture: computer-vision-based weed identification under field conditions using controlled lighting. *J. Agric. Eng. Res.* **78**(3), 233–243 (2001)

11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of a Advances in Neural Information Processing Systems (NIPS), pp. 1106–1114 (2012)
12. Kumar, N., Belhumeur, P.N., Biswas, A., Jacobs, D.W., Kress, W.J., Lopez, I., Soares, J.V.B.: Leafsnap: A computer vision system for automatic plant species identification. In: The 12th European Conference on Computer Vision (ECCV), pp. 502–516 (2012)
13. Lee, S.H., Chan, C.S., Wilkin, P., Remagnino, P.: Deep-plant: Plant identification with convolutional neural networks. In: Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 452–456 (2015)
14. Lin, H., Bilmes, J.: A class of submodular functions for document summarization. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. vol. 37, pp. 510–520 (2011)
15. Lottes, P., Hoeflerlin, M., Sander, S., Mütter, M., Schulze Lammers, P., Stachniss, C.: an effective classification system for separating sugar beets and weeds for precision farming applications. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2016)
16. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions-i. *Math. Program.* **14**(1), 265–294 (1978)
17. Parkhi, O.M., Vedaldi, A., Jawahar, C.V., Zisserman, A.: Cats and dogs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3498–3505 (2012)
18. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(8), 1226–1238 (2005)
19. Reyes, A.K., Caicedo, J.C., Camargo, J.E.: Fine-tuning deep convolutional networks for plant recognition. In: Working Notes of Conference and Labs of the Evaluation forum (CLEF) (2015)
20. Rouse, Jr., J.W., Haas, R.H., Schell, J.A., Deering, D.W.: Monitoring vegetation systems in the great plains with ERTS. In: Proceedings of the 3rd Earth Resource Technology Satellite (ERTS) Symposium. vol. 1 (1974)
21. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2014)
22. Shearer, S.A., Holmes, R.G.: Plant identification using color co-occurrence matrices. *Trans. ASAE* **33**(6), 2037–2044 (1990)
23. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their location in images. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2005)
24. Tellaeche, A., Burgos-Artizzu, X.P., Pajares, G., Ribeiro, A.: A vision-based method for weeds identification through the bayesian decision theory. *Pattern Recognit.* **41**(2), 521–530 (2008)
25. Vinh, L.T., Lee, S., Park, Y., d’Auriol, B.J.: A novel feature selection method based on normalized mutual information. *Appl. Intell.* **37**(1), 100–120 (2011)
26. Wang, X.F., shuang Huang, D., xiang Du, J., Xu, H., Heutte, L.: Classification of plant leaf images with complicated background. *Appl. Math. Comput.* **205**(2), 916–926 (2008)
27. Yao, B., Khosla, A., Fei-Fei, L.: Combining randomization and discrimination for fine-grained image categorization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1577–1584 (2011)

Supervised Autonomy for Exploration and Mobile Manipulation in Rough Terrain

Max Schwarz, Sebastian Schüller, Christian Lenz,
David Droeschel and Sven Behnke

Abstract Planetary exploration scenarios illustrate the need for robots that are capable to operate in unknown environments without direct human interaction. Motivated by the DLR SpaceBot Cup 2015, where robots should explore a Mars-like environment, find and transport objects, take a soil sample, and perform assembly tasks, we developed autonomous capabilities for our mobile manipulation robot Momaro. The robot perceives and maps previously unknown, uneven terrain using a 3D laser scanner. We assess drivability and plan navigation for the omnidirectional drive. Using its four legs, Momaro adapts to the slope of the terrain. It perceives objects with cameras, estimates their pose, and manipulates them with its two arms autonomously. For specifying missions, monitoring mission progress, and on-the-fly reconfiguration, we developed suitable operator interfaces. With the developed system, our team NimbRo Explorer solved all tasks of the DLR SpaceBot Camp 2015.

Keywords Supervised autonomy · Anthropomorphic robot · Mobile manipulation · Robotic challenges · Planetary exploration

1 Introduction

In planetary exploration scenarios, robots are needed that are capable of operating autonomously in unknown environments and highly unstructured and unpredictable situations. To address this need, the German Aerospace Center (DLR) held the DLR

M. Schwarz (✉) · S. Schüller · C. Lenz · D. Droeschel (✉) · S. Behnke (✉)
Institute for Computer Science VI, Autonomous Intelligent Systems, University of Bonn,
Friedrich-Ebert-Allee 144, 53113 Bonn, Germany
e-mail: max.schwarz@uni-bonn.de

D. Droeschel
e-mail: droeschel@ais.uni-bonn.de

S. Behnke
e-mail: behnke@cs.uni-bonn.de

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_10



Fig. 1 The mobile manipulation robot Momaro taking a soil sample

SpaceBot Camp 2015.¹ The robots needed to tackle the following tasks: (i) find and identify three previously known objects in a planetary-like environment (cup, battery, and base station); (ii) take a soil sample of a previously known spot (optional); (iii) pick up and deliver the cup and the battery to the base station; and (iv) assemble all objects.

All tasks had to be completed in 60 min as autonomously as possible, including perception, manipulation and navigation in difficult terrain. A coarse height map with 50 cm resolution of the environment was known prior to the run. No line-of-sight between the robot and the crew was allowed and communication between the robot and the operators was restricted by a round trip latency of 4 s and scheduled blackouts.

To address these tasks, we used the mobile manipulation robot Momaro (see Fig. 1), which is configured and monitored from a ground station. Momaro is equipped with four articulated compliant legs that end in pairs of directly driven, steerable wheels. This flexible locomotion base allows it to drive on suitable terrain and to make steps when required to overcome obstacles. Momaro has an anthropomorphic upper body with two 7 degrees of freedom (DOF) arms that end in dexterous grippers. Through adjustable base-height and attitude and a yaw joint in the spine, our robot has a work space equal to the one of an adult person.

Momaro is equipped with a 3D laser scanner, multiple color cameras, an RGB-D camera, and a fast onboard computer. The robot communicates to a relay at the landing site via WiFi and is powered by a rechargeable LiPo battery.

¹<http://www.dlr.de/rd/desktopdefault.aspx/tabid-8101/>.

The developed system was tested successfully at the DLR SpaceBot Camp 2015. In this paper, we report on the robust perception, state estimation, navigation, and manipulation methods that we developed for exploration and mobile manipulation in rough terrain with supervised autonomy, i.e. autonomous robot operation under supervision of a human operator crew, which can configure and monitor the operation on a high level.

2 Related Work

The need for mobile manipulation has been addressed with the development of a variety of mobile manipulation systems, consisting of robotic arms installed on mobile bases with the mobility provided by wheels, tracks, or leg mechanisms. Several research groups use purely wheeled locomotion for their robots, e.g. [1, 2]. In previous work, we developed NimbRo Explorer [3], a six-wheeled robot equipped with a 7 DOF arm designed for mobile manipulation in rough terrain encountered in planetary exploration scenarios. Compared to wheeled robots, legged robots are more complex to design, build, and control, but they have obvious mobility advantages when operating in unstructured terrains and environments, see e.g. [4, 5]. Some groups have started investigating mobile robot designs which combine the advantages of both legged and wheeled locomotion, using different coupling mechanisms between the wheels and legs, e.g. [6, 7].

In 2013, DLR held a very similar SpaceBot competition which encouraged several robotic developments [8]. Heppner et al. [9] describe one of the participating systems, the six-legged walking robot LAURON V. LAURON is able to overcome challenging terrain, although its six legs limit the locomotion speed in comparison to wheeled robots. Sünderhauf et al. [10] developed a cooperative team of two wheeled robots, which had good driving capabilities, but failed due to communication issues. Schwendner et al. [11] developed the six-wheeled Artemis rover able to passively cope with considerable terrain slopes (up to 45°). In contrast, Momaro employs active balancing strategies (see Sect. 5.3).

In our previous work [3], we describe the Explorer robot used in the 2013 competition and its local navigation system [12]. Compared to the 2013 system, we improve on (i) capabilities of the mechanical design (e.g. execution of stepping motions and bimanual manipulation); (ii) degree of autonomy (autonomous execution of full missions, including assembly tasks at the base station); (iii) situational awareness of the operator crew; and (iv) robustness of network communication.

The local navigation approach has moved from a hybrid laser-scanner-and-RGB-D system on three levels to a laser scanner-only system on two levels—allowing operation in regions where current RGB-D sensors fail to measure distance (e.g. in direct sunlight).

In contrast to many other robots, Momaro can drive omnidirectionally, which simplifies navigation in restricted spaces and allows us to make small lateral positional corrections faster. Furthermore, our robot is equipped with six limbs, two of which

are exclusively used for manipulation. The use of four legs for locomotion provides a large and flexible support polygon when the robot is performing mobile manipulation tasks. The Momaro system demonstrated multiple complex tasks under teleoperation in the DARPA Robotics Challenge [13, 14].

Supervised autonomy has been proposed by Cheng et al. [15], who shift basic autonomous functions like collision avoidance from the supervisor back to the robot, while offering high-level interfaces to configure the functions remotely. In contrast to human-in-the-loop control, supervised autonomy is more suited for the large latencies of space communication. Gillett et al. [16] use supervised autonomy for an unmanned satellite servicing system that must perform satellite capture autonomously. The survey by Pedersen et al. [17] highlights the trend in space robotics towards more autonomous functions, but also points out that space exploration will always have a human component, if only as consumers of the data produced by the robotic system. In this manner, supervised autonomy is also the limit of sensible autonomy in space exploration.

3 Mapping and Localization

For autonomous navigation during a mission, our system continuously builds a map of the environment and localizes within this map. To this end, 3D scans of the environment are aggregated in a robot-centric local multiresolution map. The 6D sensor motion is estimated by registering the 3D scan to the map using our efficient surfel-based registration method [18]. In order to obtain an allocentric map of the environment—and to localize in it—individual local maps are aligned to each other using the same surfel-based registration method. A pose graph that connects the maps of neighboring key poses is optimized globally. The architecture of our perception and mapping system is outlined in Fig. 2.

3.1 Preprocessing and 3D Scan Assembly

The raw measurements from the laser scanner are subject to spurious measurements at occluded transitions between two objects. These so-called *jump edges* are filtered by comparing the angle of neighboring measurements. After filtering for jump edges, we assemble a 3D scan from the 2D scans of a complete rotation of the scanner. Since the sensor is moving during acquisition, we undistort the individual 2D scans in two steps.

First, measurements of individual 2D scans are undistorted with regards to the rotation of the 2D laser scanner around the sensor rotation axis. Using spherical linear interpolation, the rotation between the acquisition of two scan lines is distributed over the measurements.

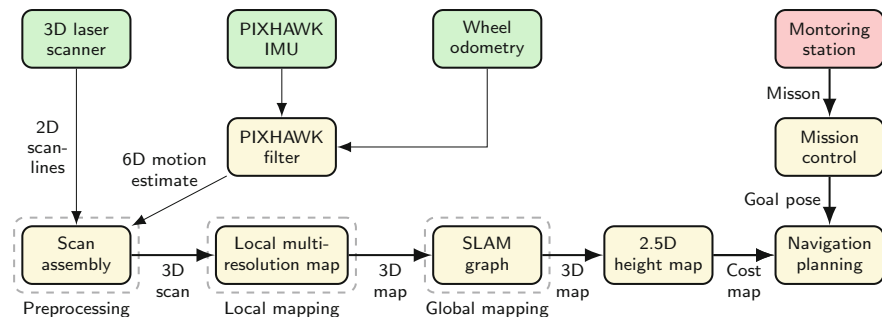


Fig. 2 Overview of mapping, localization, and navigation. Laser-range measurements are processed (Sect. 3.1). Scans are registered with and stored in a local multiresolution map (Sect. 3.2). Keyframe views of local maps are registered against each other in a SLAM graph (Sect. 3.3). A 2.5D height map is used to assess drivability. A 2D grid-based approach is used for planning (Sect. 5)

Second, the motion of the robot during acquisition of a full 3D scan is compensated. Due to Momaro’s flexible legs, it is not sufficient to simply use wheel odometry to compensate for the robot motion. Instead we estimate the full 6D state with the Pixhawk IMU attached to Momaro’s head. Here we calculate a 3D attitude estimate from accelerometers and gyroscopes to compensate for rotational motions of the robot. Afterwards, we filter the wheel odometry with measured linear acceleration to compensate for linear motions. The resulting 6D state estimate includes otherwise unobservable motions due to external forces like rough terrain, contacts with the environment, wind, etc. It is used to assemble the individual 2D scans of each rotation to a 3D scan.

3.2 Local Mapping

Distance measurements from the laser-range sensor are accumulated in a 3D multiresolution map with increasing cell sizes from the robot center. The representation consists of multiple robot-centered 3D grid-maps with different resolutions. On the finest resolution, we use a cell length of 0.25 m. Each grid-map is embedded in the next level with coarser resolution and doubled cell length. The stored points and grid structure are shown in Fig. 3.

We use a hybrid representation, storing 3D point measurements along with occupancy information in each cell. Similar to [19], we use a beam-based inverse sensor model and ray-casting to update the occupancy of a cell. For every measurement in the 3D scan, we update the occupancy information of cells on the ray between the sensor origin and the endpoint. Point measurements of consecutive 3D scans are stored in fixed-sized circular buffers, allowing for point-based data processing and facilitating efficient nearest-neighbor queries.

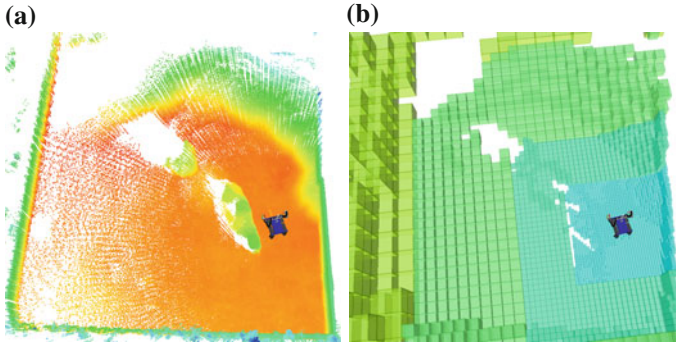


Fig. 3 Local multiresolution map. **a** 3D points stored in the map on the robot. Color encodes height. **b** Cell size increases with the distance from robot

After a full rotation of the laser, the newly acquired 3D scan is registered to the so far accumulated map to compensate for drift of the estimated motion. For aligning a 3D scan to the map, we use our surfel-based registration method [18] designed for this data structure. It leverages the multiresolution property of the map and gains efficiency by summarizing 3D points to surfels that are then used for registration. Measurements from the aligned 3D scan replace older measurements in the map and are used to update the occupancy information.

3.3 Allocentric Mapping

To estimate the motion of the robot, we incorporate IMU measurements, wheel odometry, and the local registration results. While these estimates allow us to control the robot and to track its pose over a short period of time, they are prone to drift and thus are not suitable for continuing localization. Furthermore, they do not provide a fixed allocentric frame for the definition of mission-relevant poses. Thus, we build an allocentric map by means of laser-based SLAM and localize towards this map during autonomous operation.

This allocentric map is built by aligning multiple local multiresolution maps, acquired from different view poses [20]. We model different view poses as nodes in a graph that are connected by edges. A node consists of the local multiresolution map from the corresponding view pose. Each edge in the graph models a spatial constraint between two nodes.

After adding a new 3D scan to the local multiresolution map as described in Sect. 3.2, the local map is registered towards the previous node in the graph using the same registration method. A new node is generated for the current local map, if the robot moved sufficiently far. The estimated transformation between a new node and the previous node models a spatial constraint and is maintained as the value of the

respective edge in our pose graph. In addition to edges between the previous node and the current node, we add spatial constraints between close-by view poses that are not in temporal sequence.

From the graph of spatial constraints, we infer the probability of the trajectory estimate given all relative pose observations. Each spatial constraint is a normally distributed estimate with mean and covariance. This pose graph optimization is efficiently solved using the g^2o framework [21], yielding maximum likelihood estimates of the view poses.

3.4 Localization

While traversing the environment, the pose graph is extended and optimized whenever the robot explores previously unseen terrain. We localize towards this pose graph during the entire mission to estimate the pose of the robot in an allocentric frame. When executing a mission, e.g. during the SpaceBot Camp, the robot traverses goal poses w.r.t. this allocentric frame.

Since the laser scanner acquires complete 3D scans with a relatively low rate, we incorporate the egomotion estimate from the wheel odometry and measurements from the IMU to track the robot pose. The egomotion estimate is used as a prior for the motion between two consecutive 3D scans. In detail, we track the pose hypothesis by alternating the prediction of the robot movement given the filter result and alignment of the current local multiresolution map towards the allocentric map of the environment.

The allocentric localization is triggered after acquiring a 3D scan and adding it to the local multiresolution map. Due to the density of the local map, we gain robustness. We update the allocentric robot pose with the resulting registration transform. To achieve real-time performance of the localization module, we track only one pose hypothesis.

During the SpaceBot Camp, we assumed that the initial pose of the robot was known, either by starting from a predefined pose or by means of manually aligning our allocentric coordinate frame with a coarse height map of the environment. Thus, we could navigate to goal poses specified in the coarse height map by localizing towards our pose graph.

3.5 Height Mapping

As a basis for assessing drivability, the 3D map is projected into a 2.5D height map, shown in Fig. 5. In case multiple measurements are projected into the same cell, we use the measurement with median height. Gaps in the height map (cells without measurements) are filled with a local weighted mean if the cell has at least two neighbors within a distance threshold (20 cm in our experiments). This provides a

good approximation of occluded terrain until the robot is close enough to actually observe it. After filling gaps in the height map, the height values are spatially filtered using the fast median filter approximation using local histograms [22]. The height map is suitable for navigation planning (see Sect. 5).

4 Object Perception

For approaching objects and adapting motion primitives to detected objects, RGB images from a wide-angle camera and RGB-D point clouds from an Asus Xtion camera, both mounted on the sensor head, are used. We differentiate between object detection (i.e. estimating an approximate 3D object position) and object registration (i.e. determining an accurate 6D object pose).

The objects provided by DLR are color-coded. We classify each pixel by using a precomputed lookup table in YUV space. When approaching an object, object detection is initially performed with the downwards-facing wide-angle camera (Fig. 4a). Using the connected component algorithm, we obtain object candidate clusters of same-colored pixels. An approximate pinhole camera model calculates the view ray for each cluster. Finally, the object position is approximated by the intersection of the view ray with the local ground plane. The calculated object position is precise enough to allow approaching the object until it is in the range of other sensors.

As soon as the object is in range of the ASUS Xtion camera, the connected component algorithm can also take Cartesian distance into account. We use the PCL implementation of the connected component algorithm for organized point clouds. Since

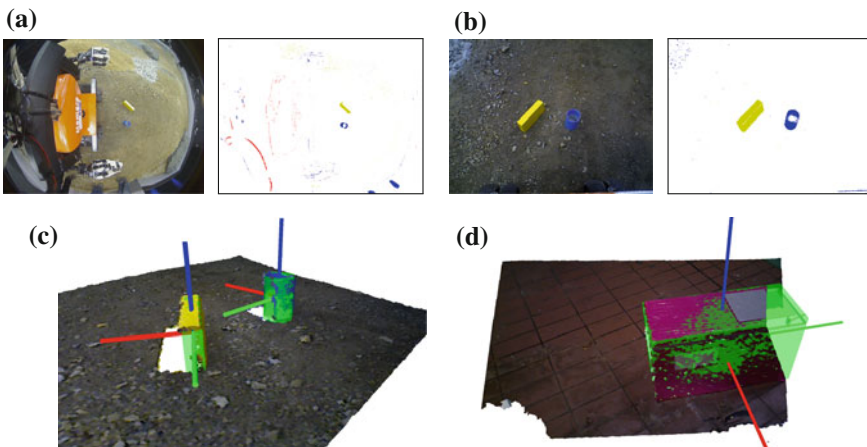


Fig. 4 Object perception. Color segmentation using (a) wide-angle camera and (b) RGB-D camera; RGB-D point clouds showing registered (c) cup and battery and (d) base station. The registered models are shown in *green*

the depth measurements allow us to directly compute the cluster centroid position, and the camera is easier to calibrate, we can approach objects much more precisely using the RGB-D camera (Fig. 4b).

When the object is close enough, we use registration of a CAD model to obtain a precise object pose (Fig. 4c, d). Since color segmentation often misses important points of the objects, we perform a depth-based plane segmentation using RANSAC and Euclidean clustering as detailed by [23] to obtain object clusters. The object models are then registered to the clusters using Generalized ICP [24]. The estimated object pose is then normalized respecting the symmetry axes/planes of the individual object class. For example, the cup is symmetrical around the Z axis, so the X axis is rotated such that it points in the robot's forward direction (see Fig. 4).

5 Navigation

Our autonomous navigation solution consists of two layers: The global path planning layer and the local trajectory planning layer. Both planners are fed with cost maps calculated from the aggregated laser measurements.

5.1 Local Height Difference Maps

Since caves and other overhanging structures are the exception on most planetary surfaces, the 2.5D height map generated in Sect. 3.5 suffices for autonomous navigation planning.

The 2.5D height map is transformed into a multi scale height difference map. For each cell, we calculate local height differences at multiple scales l . We compute $D_l(x, y)$ as the maximum difference to the center cell (x, y) in a local l -window:

$$D_l(x, y) := \max_{\substack{|u-x| < l; u \neq x \\ |v-y| < l; v \neq y}} |H(x, y) - H(u, v)|. \quad (1)$$

$H(u, v)$ values of NaN are ignored. In the cases where the center cell $H(x, y)$ itself is not defined, or there are no other defined l -neighbors, we assign $D_l(x, y) := \text{NaN}$. Small, but sharp obstacles show up on the D_l maps with lower l scales. Larger inclines, which might be better to avoid, can be seen on the maps with a higher l value.

5.2 Path Planning

During the SpaceBot Camp, we used the standard ROS `navfn`² planner. Afterwards, we replaced it with a custom A* planner to consider gradual costs fully, which the ROS planner was not designed to do. We transform the height difference map into a cost map that can be used for path planning.

A combined difference map, \tilde{D} is generated by linear combination of different D_l maps to comprise information about smaller obstacles and larger inclines. The summands from the D_3 and D_6 maps are constrained to a response of $\frac{1}{2}$ to prevent the creation of absolute obstacles from a single scale alone. The smallest scale D_1 is allowed to create absolute obstacles, since sharp obstacles pose great danger to the robot:

$$\tilde{D}(x, y) := \sum_{l \in \{1, 3, 6\}} \begin{cases} \lambda_l D_l & \text{if } l = 1 \\ \min\{0.5; \lambda_l D_l\} & \text{otherwise.} \end{cases} \quad (2)$$

The values for the λ_l parameters were found empirically: $\lambda_1 = 2.2$, $\lambda_2 = 3.6$, $\lambda_3 = 2.5$.

5.2.1 Global Path Planning

For global path planning, we implemented an A* graph search on the 2D grid map. The Euclidean distance (multiplied with the minimum cost in the grid map) is used as the heuristic function for A*. This planning does not account for the robot foot print and considers the robot as just a point in the 2D grid. To ensure the generation of a safe path, we inflate obstacles in the cost map to account for the risk closer to obstacles. The inflation is done in two steps. The cells within the distance of robot radius from absolute obstacles are elevated to absolute obstacle cost. Then for all other cells, we calculate local averages to produce costs that increase gradually close to obstacles:

$$P(x, y) := \{(u, v) : (x - u)^2 + (y - v)^2 < r^2\}, \quad (3)$$

$$D_D(x, y) := \begin{cases} 1 & \text{if } \tilde{D}(x, y) = 1, \\ \sum_{(u, v) \in P(x, y)} \frac{\tilde{D}(x, y)}{\mathcal{I}P(x, y)} & \text{otherwise.} \end{cases} \quad (4)$$

Figure 5 shows a planned path on the height map acquired during our mission at the SpaceBot Camp.

²<http://wiki.ros.org/navfn>.

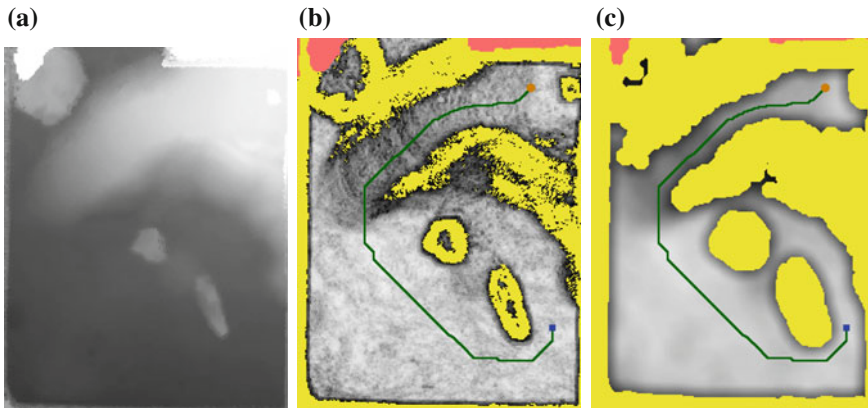


Fig. 5 Navigation planning. **a** 2.5D height map generated by projecting the 3D map. **b** Calculated traversability costs for each cell. **c** Inflated costs used for A* path planning. The *orange dot* represents the current robot position, the *blue square* the target position. *Yellow* regions represent absolute obstacles, *red* regions indicate missing measurements

5.2.2 Local Trajectory Rollout

The found global path needs to be executed by driving omnidirectionally on a local scale. To this end, we use the standard ROS `dwa_local_planner`³ package, which is based on the Dynamic Window Approach [25]. The `dwa_local_planner` accounts for the robot foot print, so cost inflation is not needed.

During navigation, the global plan is updated every 4 s, while the local rollout is re-evaluated with 3 Hz to perform the necessary adaptations to robot movement and environment changes. We also added a simple recovery behavior that first warns the operator crew that the robot is stuck and then executes a fixed backward driving primitive after a timeout expires without operator intervention.

5.3 Base Orientation Control

To prevent the robot from pitching over on the high-incline areas in the arena, we implemented a pitch control mechanism. The pitch angle of the robot is continuously measured using the IMU. We then use a simple proportional controller to compensate for the disturbance. With the commanded angle w , disturbance z , controller gain K_p , plant gain K_s , and plant disturbance gain K_{sz} , the steady state error e_b of the linearized proportional plant evolves with

³http://wiki.ros.org/dwa_local_planner.

$$e_b = \frac{1}{1 + K_s \cdot K_p} \cdot w - \frac{K_{sz}}{1 + K_s \cdot K_p} \cdot z. \quad (5)$$

Since the incline is directly measured, $K_s = 1$ and $K_{sz} = 1$. We found $K_p = 0.8$ to sufficiently stabilize for inclines present at the SpaceBot Camp. When driving up the ramp with $z \approx 15^\circ$, and setpoint $w = 0^\circ$ the resulting error (robot pitch) is $e_b \approx 8.3^\circ$. We found that this compensation enables Momaro to overcome inclines even greater than 20° without pitching over.

6 Manipulation

Since Momaro is a unique prototype, the time used for development and testing had to be balanced between individual submodules. To reduce the need for access to the real robot, we made extensive use of simulation tools. For manipulation tasks, we developed a Motion Keyframe Editor GUI to design motion primitives offline. Finished motions are then tested and finalized on the real robot with the original objects to be manipulated in the field. We show the Motion Keyframe Editor GUI in Fig. 6. With its help, we designed dedicated motions for all specific tasks in the SpaceBot Camp. We give an overview of our custom motions and their purpose in Table 1.

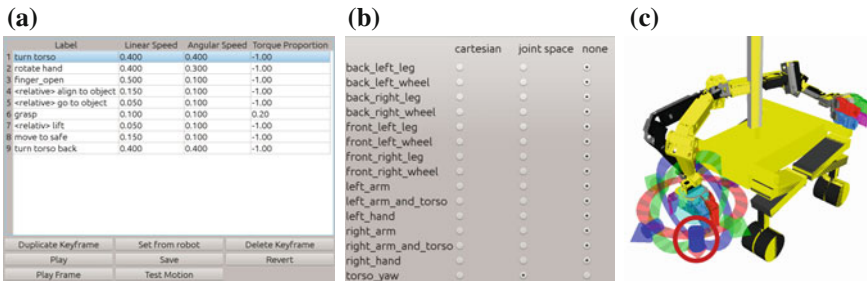


Fig. 6 Keyframe Editor GUI. **a** Motions are designed step by step and can be absolute or relative to perceived objects. **b** The user can select which joint groups are included in the currently edited keyframe and if interpolation between keyframes is Cartesian or joint space. **c** The real position of the robot is indicated in *black*. The currently edited keyframe target is shown in *yellow*. Interactive markers can be used to modify the keyframe pose in 6D. A model of the cup is placed in front of the robot to assist designing relative motions

Table 1 Custom Motions for the DLR SpaceBot Camp 2015

Motion	Purpose	Reference type
Scoop	Fill scoop tool with soil sample	Absolute
Fill cup	Pour soil into cup and discard tool	Relative to cup
Grasp cup right hand	Grasp cup with right hand from above	Relative to cup
Grasp battery left hand	Grasp battery with left hand from above	Relative to battery
Place cup	Place cup on base station	Relative to base
Place battery	Put battery into base station	Relative to base
Toggle switch	Toggle switch on side of base station	Relative to base
Grasp abort {left, right}	Move to initial position	Absolute
Reset {left, right} arm	Move all arm joints in defined position	Absolute
Reset torso	Move torso into initial position	Absolute
Cheer	Cheer to the audience	Absolute

Since it is often impossible or too slow to precisely approach an object in all 6 dimensions, we relax the assumption of absolute positioning. Motions can be designed around a reference object $T_{\text{reference}}$. When the motion is executed, the predefined endeffector pose $T_{\text{endeffector}}$ is transformed in selected keyframes i to match the perceived object $T_{\text{perceived}}$:

$$T_{\text{relative}} = T_{\text{perceived}}^{(i)} (T_{\text{reference}})^{-1} T_{\text{endeffector}}^{(i)}. \quad (6)$$

Figure 7 shows how a motion, designed relative to a reference object, is adapted to a perceived object pose to account for imprecise approach of the object.

As described in Sect. 4, the perceived objects are represented in a canonical form, removing all ambiguities resulting from symmetries in the original objects. For example, the rotation-symmetric cup is always grasped using the same yaw angle. After adaption, the Cartesian keyframes are interpolated.

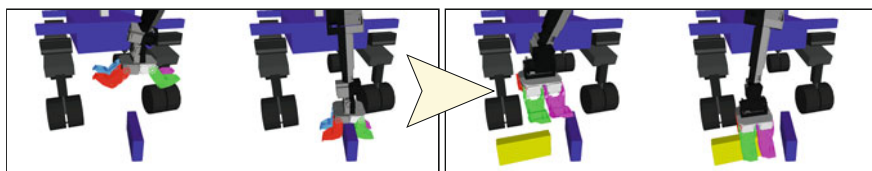


Fig. 7 Grasping objects using keyframe transformation. *Left* The *blue* reference object is grasped as the primitive was designed in the Keyframe Editor. *Right* The primitive is automatically adapted to the perceived pose of the *yellow* object

7 Evaluation

In preparation for the DLR SpaceBot finals, the SpaceBot Cup Qualification tested basic capabilities of the robotic system. To qualify, participants had to solve three tasks which involved exploration and mapping of an arena and manipulation of the cup and the battery, but no assembly. In contrast to the finals, the communication uplink time was unlimited, which lowered the required autonomy level. Using our intuitive telemanipulation approaches, our team was the only team to successfully qualify in the first attempt. Further information about our performance is available on our website.⁴ Since only two other teams managed to qualify using their second attempt, the planned SpaceBot Cup competition was changed to an open demonstration, called the DLR SpaceBot Camp.

The SpaceBot Camp required participants to solve mapping, locomotion, and manipulation tasks in rough terrain. In detail, a cup and a battery had to be located and collected on the planetary surface. If possible, the robot had to take a soil sample at a specific location and fill the cup with it. Next, the robot had to carry both objects to a base station object. The cup had to be placed on a scale located on the base station, and the battery had to be inserted into a slot on the side. By operating a switch on the other side, the base station was switched on. The participants were provided with a coarse map of the environment that had to be refined by the robot's mapping system. The communication link to the operator crew was severely constrained both in latency (2 s per direction) and in availability.

7.1 Mapping and Self-localization

Consisting of different types of stones, sand, and soil, the planetary-like environment was specially challenging for the mapping system—causing slip in odometry and vibrations of robot and sensor.

Our mapping system continuously built an allocentric map of the environment during navigation, guided by waypoints specified on the coarse height map. The coarse map and the allocentric map, generated from our mapping system is shown in Fig. 8. While showing the same structure as the coarse map, the resulting allocentric map is accurate and precisely models the environment. During a mission, the map is used for localization and to assess traversability for navigation. The estimated localization poses are shown in Fig. 11.

Although our mapping system showed very robust and reliable performance in this environment, there was one situation during the run where the operators had to intervene. Due to traversing the abandoned scoop tool—used to take the soil sample—the robot was exposed to a fast and large motion. The 3D scan distorted by the motion caused spurious measurements in the map. The operators decided to

⁴<http://www.ais.uni-bonn.de/nimbro/Explorer>.

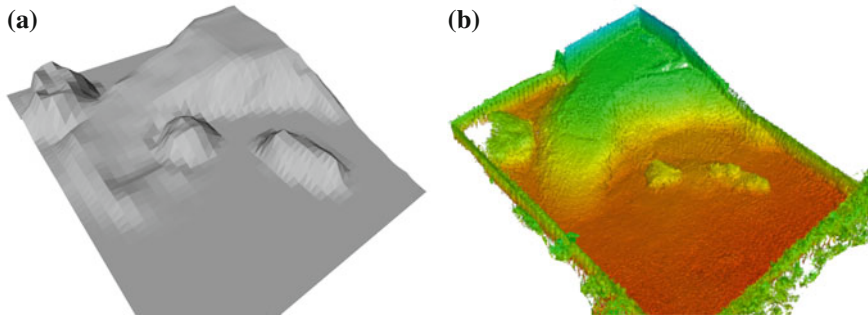


Fig. 8 Map refinement. **a** Provided coarse map of the SpaceBot Camp 2015 arena. **b** The resulting global map from data acquired during the competition

clear the SLAM map using a remote service call to prevent localization failures. The map was rebuilt from this point on and successfully used for the rest of the mission.

7.2 Navigation System

While preparing for the SpaceBot Camp, we learned that our pitch stabilization control method works even under extreme conditions. Being able to reliably overcome obstacles with inclines greater than 20° , we were confident that locomotion would not pose a problem during the competition.

Unfortunately, we only employ stabilization in pitch direction. Turning around the yaw axis on a pitched slope can result in a dangerous roll angle. We dealt with this issue during our final run by placing enough waypoints on the primary slope in the course to ensure proper orientation (see Fig. 10).

7.3 Object Manipulation

While preparing our run, we found the battery slot in the base station to have a significant resistance due to a build-in clamping mechanism. Thus, it could happen that Momaro was not able to push the battery entirely inside. Due to our flexible motion design workflow, we were able to alter the motion so that Momaro would execute small up- and downward motions while pushing to find the best angle to overcome the resistance.

The insertion of the battery requires high precision. To account for inaccuracies in both battery and station pose, we temporarily place the battery on top of the station. After grasping the battery again, we can be sure that any offset in height is compensated. Furthermore, we found it to be error prone to grasp the battery at the

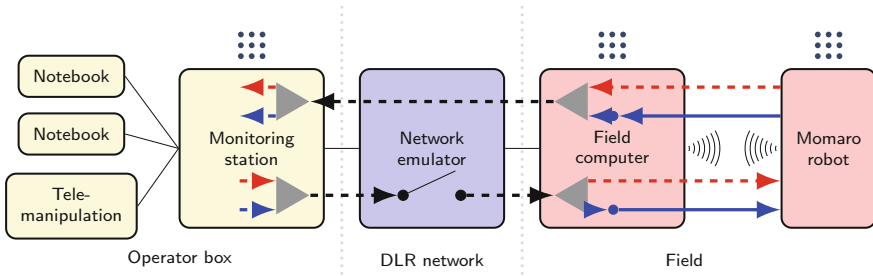


Fig. 9 Communication architecture. Operator components are shown in yellow, DLR-provided components in blue, field network components in red. Solid black lines represent physical network connections. Thick lines show the different data channels (dotted UDP, solid TCP). $\bullet\bullet\bullet$ indicates a ROS master. Streaming links are colored red, message links are shown in blue

very end, which is necessary to entirely push it inside the slot. Instead, we used two steps. First, the battery is pushed in as far as possible until the hand touches the base station. Then we release the clamped battery in the slot. Afterwards, we close the hand and push the battery inside with parts of the wrist and proximal fingers.

Overall, our straightforward keyframe adaption approach proved to be very useful. Compared to motion-planning techniques, it lacks collision avoidance and full trajectory optimization, but it is sufficient for the variety of performed tasks.

7.4 Full System Performance at DLR SpaceBot Camp 2015

Momaro solved all tasks of the SpaceBot Camp with supervised autonomy. Figure 9 illustrates the communication architecture. Figure 10 shows the operator interface used for mission planning. Our team was the only one to demonstrate all tasks including the optional taking of a soil sample. Figure 11 gives an overview of the sequence of performed tasks. A video of our performance can be found online.⁵ See Fig. 11 for detailed images of the subtasks. Timings are listed in Table 2.

Although Momaro was able to complete all tasks, this was not possible fully autonomously. While approaching the battery, a timeout aborted the process. This built-in safety-feature made operator interaction necessary to resume the approach. Without intervention, Momaro would have executed the remainder of the mission without the battery object.

As Momaro reached the main slope of the course, we also approached the time of the first communication blackout, because we lost time in the beginning due to a restart. The operator crew decided to stop Momaro at this point, as we knew that going up would be risky and intervention would have been impossible during the blackout. After the blackout, autonomous operation resumed and Momaro success-

⁵https://youtu.be/q_p5ZO-BKWM.

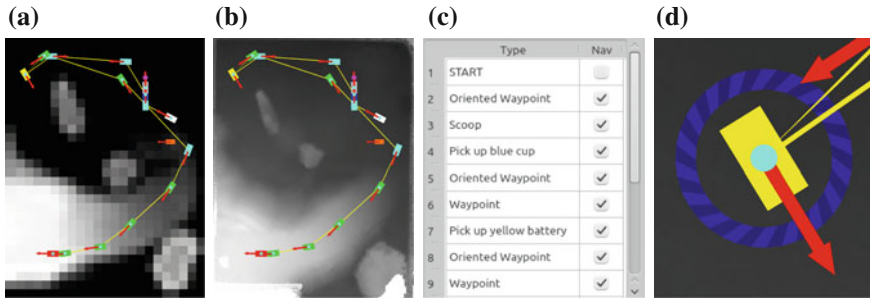


Fig. 10 Mission planning. **a** Mission plan on coarse height map provided by DLR. **b** Mission plan on detailed height map generated from the SLAM map. **c** List representation of the first eight poses. The “Nav” column can be used to disable navigation (e.g. start grasping an object immediately). **d** Pose editing using interactive marker controls. The position can be modified by dragging the rectangle. The pose is rotated by dragging on the *blue circle*

fully went up the ramp to perform the assembly tasks at the base station (Fig. 11). Although the operators paused autonomous navigation at one point on the slope to assess the situation, no intervention was necessary and navigation resumed immediately.

8 Conclusion

In this article, we presented the mobile manipulation robot Momaro and its ground station. We detail the soft- and hardware architecture of the integrated robot system and motivate design choices. The feasibility, flexibility, usefulness, and robustness of our design have been demonstrated with great success at the DLR SpaceBot Camp 2015.

Novelties include an autonomous hybrid mobile base combining wheeled locomotion with active stabilization in combination with fully autonomous object perception and manipulation in rough terrain. For situational awareness, Momaro is equipped with a multitude of sensors such as a continuously rotating 3D laser scanner, IMU, RGB-D camera, and a total of seven color cameras. Although our system was built with comprehensive autonomy in mind, all aspects from direct control to mission specification can be teleoperated through intuitive operator interfaces. Developed for the constraints posed by the SpaceBot Camp, our system also copes well with degraded network communication between the robot and the monitoring station.

The robot localizes by fusing wheel odometry and IMU measurements with pose observations obtained in a SLAM approach using laser scanner data. Autonomous navigation in rough terrain is realized by planning cost-optimal paths in a 2D map

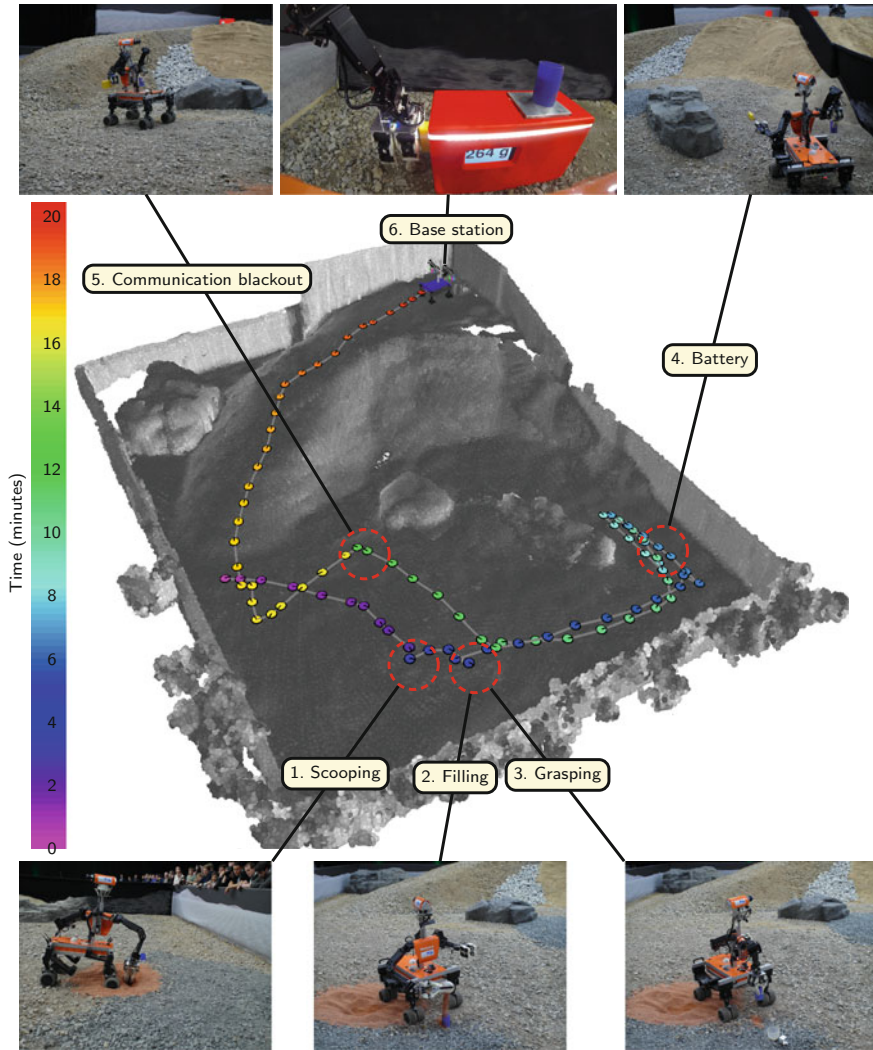


Fig. 11 Overview of the executed mission at SpaceBot Camp. The mission starts by scooping the soil sample, filling it into the cup and grasping the cup, then locating and grasping the battery pack. After waiting until the end of scheduled communication blackout, the mission is concluded by the base station assembly

Table 2 Timings of our run at the DLR SpaceBot Camp 2015

Task	Start time (mm:ss)	End time (mm:ss)	Duration (mm:ss)
Soil sample collection	1:05	1:40	0:35
Fill and grasp cup	2:15	3:05	0:50
Grasp battery	7:00	7:40	0:40
Base station assembly	18:25	20:25	2:00
Total (incl. locomotion)	0:00	20:25	20:25

of the environment. High-level autonomous missions are specified as augmented waypoints on the 2.5D height map generated from SLAM data.

For object manipulation, the robot detects objects with its RGB-D camera and executes grasps and object assembly using parametrized motion primitives.

References

1. Mehling, J., Strawser, P., Bridgwater, L., Verdeyen, W., Rovekamp, R.: Centaur: NASA's mobile humanoid designed for field work. In: Proceedings of ICRA (2007)
2. Borst, C., Wimbock, T., Schmidt, F., Fuchs, M., Brunner, B., Zacharias, F., Giordano, P.R., Konietzschke, R., Sepp, W., Fuchs, S., Rink, C., Albu-Schaffer, A., Hirzinger, G.: Rollin' Justin-mobile platform with variable base. In: ICRA (2009)
3. Stückler, J., Schwarz, M., Schadler, M., Topalidou-Kyniazopoulou, A., Behnke, S.: NimbRo explorer: semiautonomous exploration and mobile manipulation in rough terrain. *J. Field Robot. (JFR)* **33**, 407–558 (2016)
4. Semini, C., Tsagarakis, N., Guglielmino, E., Focchi, M., Cannella, F., Caldwell, D.: Design of HyQ-A hydraulically and electrically actuated quadruped robot. *J. Syst. Control Eng.* **225**, 831–849 (2011)
5. Johnson, M., Shrewsbury, B., Bertrand, S., Wu, T., Duran, D., Floyd, M., Abeles, P., Stephen, D., Mertins, N., Lesman, A., et al.: Team IHMC's lessons learned from the DARPA robotics challenge trials. *J. Field Robot.* **32**, 192–208 (2015)
6. Endo, G., Hirose, S.: Study on roller-walker (multi-mode steering control and self-contained locomotion). In: ICRA (2000)
7. Halme, A., Leppänen, I., Suomela, J., Ylönen, S., Kettunen, I.: WorkPartner: interactive human-like service robot for outdoor applications. *Int. J. Robot. Res. (IJRR)* **22**, 627–640 (2003)
8. Kaupisch, T., Noelke, D., Arghir, A.: DLR spacebot cup—Germany's space robotics competition. In: Proceedings of the Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA) (2015)
9. Heppner, G., Roennau, A., Oberländer, J., Klemm, S., Dillmann, R.: Laurope—six legged walking robot for planetary exploration participating in the SpaceBot Cup. In: WS on Advanced Space Technologies for Robotics and Automation (2015)
10. Sünderhauf, N., Neubert, P., Truschzinski, M., Wunschel, D., Pöschmann, J., Lange, S., Protzel, P.: Phobos and Deimos on Mars—two autonomous robots for the DLR SpaceBot Cup. In: 12th International Symposium on Artificial Intelligence, Robotics and Automation in Space-i-SAIRAS (2014)

11. Schwendner, J., Roehr, T.M., Haase, S., Wirkus, M., Manz, M., Arnold, S., Machowinski, J.: The Artemis rover as an example for model based engineering in space robotics. In: ICRA Workshop on Modelling, Estimation, Perception and Control of All Terrain Mobile Robots (2014)
12. Schwarz, M., Behnke, S.: Local navigation in rough terrain using omnidirectional height. In: Proceedings of the German Conference on Robotics (ROBOTIK), VDE (2014)
13. Rodehutsors, T., Schwarz, M., Behnke, S.: Intuitive bimanual telemanipulation under communication restrictions by immersive 3D visualization and motion tracking. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids) (2015)
14. Schwarz, M., Rodehutsors, T., Schreiber, M., Behnke, S.: Hybrid driving-stepping locomotion with the wheeled-legged robot Momaro. In: ICRA (2016)
15. Cheng, G., Zelinsky, A.: Supervised autonomy: a framework for human-robot systems development. *Auton. Robots* **10**, 251–266 (2001)
16. Gillett, R., Greenspan, M., Hartman, L., Dupuis, E., Terzopoulos, D.: Remote operation with supervised autonomy (rosa). In: 6th International Conference on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS) (2001)
17. Pedersen, L., Kortenkamp, D., Wettergreen, D., Nourbakhsh, I.: A survey of space robotics. In: Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space, pp. 19–23 (2003)
18. Droeschel, D., Stückler, J., Behnke, S.: Local multi-resolution representation for 6D motion estimation and mapping with a continuously rotating 3D laser scanner. In: ICRA, pp. 5221–5226 (2014)
19. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Auton. Robots* **34**, 189–206 (2013)
20. Droeschel, D., Stückler, J., Behnke, S.: Local multi-resolution surfel grids for MAV motion estimation and 3D mapping. In: IAS (2014)
21. Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: G2o: a general framework for graph optimization. In: ICRA (2011)
22. Huang, T., Yang, G., Tang, G.: A fast two-dimensional median filtering algorithm. *IEEE Trans. Acoust. Speech Signal Process.* **27**, 13–18 (1979)
23. Holz, D., Holzer, S., Rusu, R.B., Behnke, S.: Real-time plane segmentation using RGB-D cameras. In: RoboCup 2011: Robot Soccer World Cup XV, pp. 306–317 (2012)
24. Segal, A., Haehnel, D., Thrun, S.: Generalized-ICP. In: Proceedings of Robotics: Science and Systems (2009)
25. Fox, D., Burgard, W., Thrun, S., et al.: The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **4**, 23–33 (1997)

Behavior-Based Collision Avoidance Using a Cylinder-Coordinate Octree

Daniel Schmidt, Fabian Göckel and Karsten Berns

Abstract The research at hand is part of the autonomous excavator project Thor. The long term-goal is the development of an excavator which is capable of performing landscaping tasks without human intervention. As this machine creates huge forces, safety plays an important role. This research extends the behavior-based trajectory generation for the excavation and truck loading operation, working for the undisturbed case only, with collision avoidance concerning obstacles and the machine itself. Based on a data efficient cylinder-coordinate *octree* for storing obstacles, joint movements are inhibited or actively influenced to prevent the machine from hitting objects and itself. Additionally, this research shows the suitability of the extensible reactive behavior-based control approach for the automation of construction machines.

Keywords Robotics · Collision-avoidance · Behavior-based · Cylinder-coordinate octree

1 Introduction

On a modern construction site, the building process itself is highly optimized concerning machine use. Therefore, shorter construction times, lower building costs, or increased precision can only be achieved by making the machines more intelligent. This will also help avoiding dangerous situations for human co-workers. As described in [19], the goal of the THOR (Terraforming Heavy Outdoor Robot) project

D. Schmidt (✉) · F. Göckel · K. Berns
Robotics Research Lab, University Kaiserslautern, 67655 Kaiserslautern, Germany
e-mail: d_smith@cs.uni-kl.de
URL: <http://agrosy.cs.uni-kl.de>

F. Göckel
e-mail: f_goeckel10@cs.uni-kl.de

K. Berns
e-mail: berns@cs.uni-kl.de

is the full automation of landscaping tasks on an 18 t mobile excavator. The decision system is implemented within the modular robot framework *finroc* [18]. A simulation of the working environment including physics and soil behavior [20] is used to safely and cost-efficiently test the control algorithms [21]. It is always possible, that other objects cross the calculated path of the machine and a collision would directly lead to a disaster due to the large forces. Currently, the output of the behavior-based control part is a desired position and orientation (pose), which is then used by an inverse kinematics [14] solver to control the different joints of the machine. The movement of each joint can be inhibited to restrict the workspace in which the excavator may operate.

The scientific contribution of this paper is the development of a dynamic obstacle avoidance consisting of a data efficient map structure combined with a set of safety behaviors leading to safe movements of the machine. It was initially developed in [9] and is further extended within this work (Fig. 1).

The presented background work in Sect. 2 delivers insight into the development framework *finroc*, the concept of behavior-based control, and the basics of the *octree* data structure used as the map. Afterwards, the main components the cylinder-coordinate *octree* and the obstacle avoidance are presented in Sect. 3. The theoretical solution is tested and demonstrated in a simulation environment in Sect. 4. Finally, the work is concluded and an outlook into future developments is given.



Fig. 1 The bucket excavator *Thor* during autonomous truck loading

2 System Background

A classic approach used in control systems for autonomous mobile robots utilizes global knowledge to solve the task alike the sens-plan-paradigm in functional modules [7, 8, 12, 13]. Sensor information from the bottom layer is used to calculate the solution on the highest layer. It mainly showed its operability for indoor scenarios and simpler outdoor robots with only a few degrees of freedom. The large amount of information to handle within real dynamic outdoor scenarios leads to long computation times and replanning in case of environmental changes using these solutions.

In contradiction to this centralized approach, control architectures like the subsumption architecture [4–6] or the integrated behavior-based control (iB2C) [17], used within this work, split up the task into small and rather simple behavior units following the behavior-based paradigm [3]. There exists no global omniscient layer but a set of decentralized behaviors, which only operate on required information. These behavior modules have defined ports to connect them via edges to other modules and can be organized by nesting them inside groups. The standardized ports allow for triggering (stimulation s) or suppressing (inhibition i) the behavior, deliver its confidence with the current status (target rating r), and allow it to stimulate or to inhibit other behaviors (activity a). Additionally, these ports make a reuse in a different context possible [10]. More detailed information can be found in [2, 15] which also show the capability of building complex robots operating autonomously. The behavior module at hand is the result of the developments in [1, 16].

As the main requirement for this work is the arm control of the excavator, a short overview is given here. It is realized as a behavior-based inverse kinematic solver chain, implemented within the iB2C Framework [14]. The solver determines six suitable joint values from the given desired pose of the tool center point (TCP). One control behavior per joint continuously calculates a turning delta to further approach the goal. They are arranged in the same way as the kinematic chain of the excavator boom. The ones closer to the TCP inhibit those which are further away from the TCP. If the determined pose is close enough to the desired one, the solver process is considered to be finished.

Execution of the obstacle avoidance on raw and unfiltered sensor data is not efficient and unsmooth in case of noise. Therefore, a data structure is required which can store, filter, and deliver spatial information about obstacles in three-dimensional space representing the whole construction area. As the whole area can have a square footage of a few hundred meters, contain a lot of different objects, but also regions with identical properties, an equally sized three-dimensional grid requires a huge amount of data and is not an efficient data structure. On the other hand a list of pose vectors towards explicitly stored obstacles would become large in a real scenario and require long times for searching. Therefore, the *octree* structure is used within this research work, gathering its name from the possible eight children of each tree node [11]. As it is only detailed in regions with a lot of obstacles, it is able to span over a wide space. Only cells which contain objects get more and more detailed.

This allows for a detailed but still lightweight, compact, and quick solution. These advantages make the *octree* an ideal concept for storing the environmental data of an excavation area.

3 Main Components

The obstacle avoidance system requires a data structure storing the geometries of environmental objects like trees, buildings, containers, trucks, or other machinery. Additionally, the shape of the machine itself needs to be stored to avoid self-collision. Moreover, it should be possible to add artificial objects which prevent the excavator from entering specific areas. As the machine operates in cylindrical coordinates, a data storage which operates in the same coordinate system avoids unnecessary conversions.

The module `ObstacleOctree` undertakes the task of a gateway to control the communication to and from the *octree* object. It has the tasks of initializing a `CylinderCoordinateOctree`, insertion of objects representing the excavator geometry to avoid self-collision, offer for input ports to allow embedding of obstacles from the preprocessed sensor data, and providing the opportunity to visualize the content and the grid structure of the octree.

3.1 Cylinder-Coordinate Octree

To separate the cells into sub-cells the three dimensions azimuth, length, and height are divided into half's forming the possible eight (2^3) children of a node in the *octree* presented in Definition 1.

Definition 1 (*Cylinder-Coordinate Octree*) A cylinder-coordinate octree is defined as a quadruple (N, E, S, N_0) with

- N is the finite, non-empty set of nodes (n_i) ,
- E is the set of edges $E = \{e_i | e_i = (n_i, n_j), i > j, |\{n_i | (n_i, n_j) \in E\}| \leq 8\}$,
- S is the set of states $S = \{(n_i, s_j) | s_i \in \{empty, semi-occupied, occupied\}, n_j \in N\}$, and
- N_0 is the initial set of N containing only the root node n_0 .

Figure 2 shows two separation steps of the main tree into sub-trees. In the second division step the original space is separated into 64 sub-cells, although a real separation is only performed, if the cell is semi-occupied.

Usually, large regions of the *octree* stay empty and thus also undivided. A substantial aspect of the low memory allocation is the high processing speed for all operations. This is due to the fact, that all methods operate recursively and terminate in case the requested region of the tree is empty. Each cell has a status property

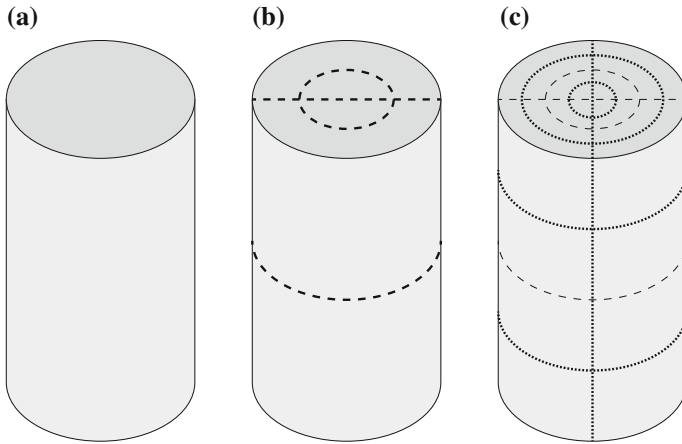
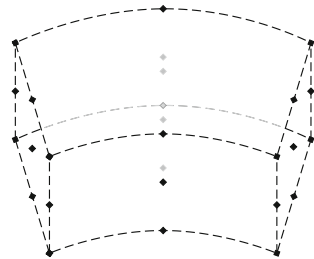


Fig. 2 The *octree* before subdivision (a), in the first subdivision (b), and in the second subdivision with 64 cells (c)

which is either set to *occupied*, *semi-occupied* or *empty*. Although the structure of the *cylindrical octree* is the same as for the Cartesian one, the method algorithms compute differently in most cases. After initialization the tree only consists of an *empty* root cell which covers the whole space. Each cell is able to have either zero or eight sub-cells which inherit all properties from the parent cell—except of the dimensions. To fill the *octree* with data, the class provides several methods to insert either a point cloud or a geometric object like a cylinder, cuboid, or ellipsoid. It is possible to retrieve information like the shortest distance from a given point to the nearest object.

The main part of the structure is the `CylinderCoordinateOctree` starting with an empty root cell. Its minimum cell dimensions define the achievable storage precision. As a full geometric comparison would be too complex for the insertion of elements, the cell is fragmented into a grid structure. Therefore, adding new elements only requires a test for these grid cells which tremendously speeds up the process. A generalized cell of the cylindrical *octree* with its grid points is shown in Fig. 3.

Fig. 3 Grid structure of a general polar cell with 27 points at the start, center, and end of each coordinate system dimension



Another algorithmic improvement allows for the insertion of objects in any rotated or translated state without complex calculations and independently from the object type. The grid points are multiplied with the negative rotation matrix and then subtracted with the translation vector. This way the comparison takes place at the origin of the coordinate system which is less complex to compute.

The main sources of environment data are three-dimensional point clouds delivered by laser scanners. They are inserted based on an recursive approach which perfectly fits to the tree structure of the *octree*. Algorithm 1 shows the rough steps of the procedure.

Algorithm 1 Insertion of a point cloud in the *octree*.

```

1: procedure INSERTPOINTS(points,threshold,attribute)
2:   return InsertPointsRecursive(root_cell,points,threshold,attribute);
3: procedure INSERTPOINTSRECURSIVE(cell, points, threshold, attribute)
4:   if (SeparateCell(cell)) then
5:     for (Go through all points) do
6:       for (Go through all sub-cells) do
7:         if (points_belonging_to_sub-cell > 0) then
8:           sub-cell.status = eOCS_SEMI_OCCUPIED;
9:           InsertPointsRecursive(sub-cell, subPoints, threshold, attribute);
10:  else
11:    if (points.size() > threshold) then
12:      cell.status = eOCS_OCCUPIED;
13:      cell.attribute = attribute;
14:  return cell;

```

More complicated is the process of inserting full objects into the *octree*. It starts at the root cell and recursively checks which of the sub-cells are covered by the object, finishing when the minimum cell size is reached. Figure 4 visualizes the covering decision process for a cylindrical object. Therefore, a grid representation of the cell and the object is used to determine which case is relevant. Only the generation of

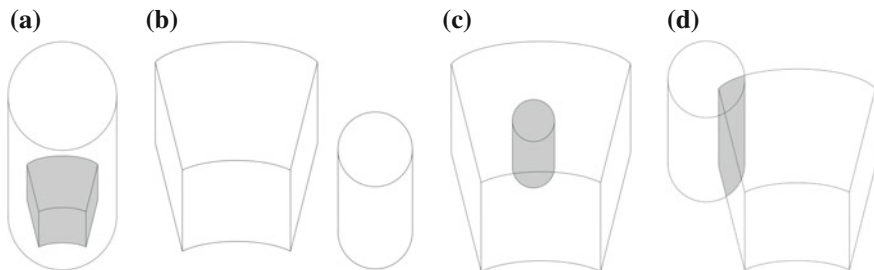


Fig. 4 Visualization of the covering decision process for a cylindrical object. The cylinder either covers the cell completely (a), is completely outside of the cell (b), is completely inside of the cell (c), or is partially inside of the cell (d)

the grid representation and the inclusion calculation are different for the insertion of different objects. The following steps are executed to perform the task:

1. Generate the grid structure of the object (cylinder, cuboid, or ellipsoid).
2. Generate the grid representation for all sub-cells for the following cases:

Case 1: If every grid point is inside the object, the status is *occupied*.

Case 2: In reverse it must be checked, if all object grid points are outside of the sub-cell leading to an *empty* cell.

Case 3: Otherwise, the cell is *semi-occupied*.

3. Only if the cells status is *semi-occupied* a deeper recursion step is necessary (very efficiency for large unoccupied workspaces).
4. This recursion continues, as long as the separation is successful.

The *octree* also provides different data request methods, which allow for retrieving simple information like the cell or attribute status. The most important method for the obstacle avoidance is the calculation of the shortest distance vector from a given point to the next obstacle object. This method works in the following way:

1. Perform a recursion to the given point and stop if the status is *occupied*.
2. Otherwise the return path is checked for close objects. If the parent cell is *semi-occupied*, a recursive search is performed within the seven remaining sub-cells to find an object with a smaller distance than the shortest distance.
3. To improve the efficiency, further recursion is only performed, when it is currently possible that the cell contains a closer object than the current one.
4. Because of the fact, that the examination is started directly at the given point, close objects are found early.

This algorithm allows for getting very fast results, which is important as this will be the mostly used method called by the safety behaviors to calculate their related joint inhibition value.

3.2 Obstacle Avoidance

The essential core functionality of the obstacle avoidance system is implemented inside the `WorkspaceLimiter` module. It computes the inhibition of the excavator joints based on the stored obstacle data. Therefore, it has to continuously fetch the current state of the arm joints, send requests for every joint to the `CylinderCoordinateOctree` module for the nearest obstacle, receive the answer data from the previous cycle, calculate the workspace for every joint consisting of minimum and maximum values for every dimension, include only obstacles which are currently reachable for the excavator arm, limit the workspace only if really necessary as the inhibition values must not obstruct the excavator during operation, and finally publish the calculated workspace limits.

Behavior-based restriction modules are used to produce the actual inhibition and gear into the behavior chain for each joint — one to restrict the azimuth, the length, and the height of the related joints. Whereas length and height is influenced by every joint, only the boom and TCP influence the azimuth angle. Therefore, the azimuth restriction module only handles the latter ones. The task of the restriction modules is to use the current joint states and the minimum and maximum values from the `WorkspaceLimiter` to generate inhibition values that can be linked with the respective arm joint controller behaviors. The following three different cases are possible:

1. **Full (1):** In case the current value is higher than the given maximum or lower than the minimum, the inhibition is fully activated.
2. **Approximated ([0,1]):** Inside a defined approximation range the inhibition value gets proportional adjusted between 0 and 1.
3. **None (0):** If the value is neither in the safety or in approximation range nor higher or lower than the maximum or minimum no inhibition is necessary.

An additional safety feature is a history smoother that eliminates the effects of peaks in the calculated obstacle distance values due to sensor noise. It guarantees, that the full inhibition status is only abandoned when it is really safe and not as a consequence of a single incorrect value, as depicted in Fig. 5.

The `WorkspaceLimiter` uses the information stored in the obstacle tree to calculate the workspaces for every component of the excavator arm. It also computes at witch point on the component the distance to the obstacle is the smallest, as visualized in Fig. 6. The structure for a single height workspace calculation of one component is demonstrated by algorithm 2. Input parameters are the positions of the start joint (*joint1*) and the end joint (*joint2*) of the component, the distance of the nearest obstacle to *joint1* (*obstacle1_distance*), the position of the nearest obstacle to *joint2* (*obstacle2_position*), the length of the component (*component_length*), detection boundary factor, that controls the extension of the field of observation (*detection_boundary_factor*), and the inhibition ranges for azimuth, length and height which indicate the ranges in which a workspace limitation is performed. As a collision with the component is only possible if the obstacle distance is smaller than the component length, this check is performed in line

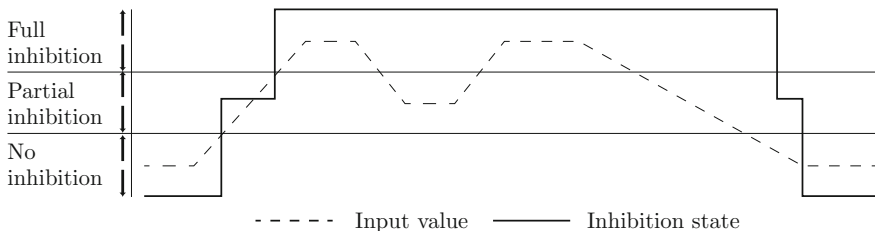
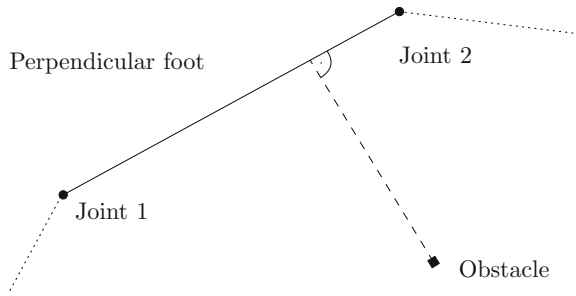


Fig. 5 Visualization of the inhibitions state change. The history value prevents the inhibition to drop by small input deviations and so smooths the output

Fig. 6 Depiction of the perpendicular foot identification



2. Afterwards line 3 tests, if the obstacle is lying inside the inhibition range of the other dimensions. This is done because it is only necessary to consider an obstacle for the height limitation, if azimuth and length are lying inside a given range. Otherwise an obstacle that may be very close in height but with a different azimuth would cause an inhibition, which would lead to unwanted effects, as the joint movements are unnecessarily restricted. As it is important to not collide with any component, the perpendicular foot is used, which delivers the shortest distance to the component itself. In lines 6–9 it is differentiated between the minimum and maximum cases based on the obstacles position. Discrimination of joint limitations is prevented by setting only one of the limitations per obstacle. The case of equal distances is skipped on purpose, as *joint2* lays aside the obstacle and a limitation would not make much sense. The structure of the length and azimuth workspace calculations is basically the same for all joints. There are three modules with similar functions to restrict the behaviors—One for the height, the length, and for the azimuth. The modules perform the inhibition calculations for all responsible joints in parallel. Initially, the inhibition values are calculated based on the joint angle and limitation values. These are summed up and restricted for the kinematic dependent joints.

Algorithm 2 Extract of the height workspace calculation for a component.

```

1: perpendicular_foot=CalculatePerpendicularFoot(obstacle2_position, joint1_position,
joint2_position);
2: if (obstacle1_distance ≤ component_length · detection_boundary_factor) then
3:   if (perpendicular_foot.Azimuth() - obstacle2_position.Azimuth() <
4:     inhibition_range_azimuth) && (perpendicular_foot.Length() -
5:     obstacle2_position.Length() < inhibition_range_length) then
6:     if (obstacle2_position.Height() < joint2_position.Height()) then
7:       out_min_height.Publish(obstacle2_position.Height());
8:     else if (obstacle2_position.Height() > joint2_position.Height()) then
9:       out_max_height.Publish(obstacle2_position.Height());

```

Algorithm 3 shows the calculation of the inhibition value by means of an example maximum height inhibition calculation. Three different inhibition states are distinguished and the history smoother is implemented. Each inhibition value has an own

history value, which is initialized as soon as a full inhibition is reached. If the input value falls below the full inhibition level, the history value is decreased, instead of the inhibition itself. This way, the inhibition remains at its maximum level for several cycles.

Algorithm 3 Height inhibition calculation

```

1: procedure CALCULATEHEIGHTINHIBITION(actual_height, max_height, inhibition_history)
2:   if (actual_height ≥ (max_height - inhibition_range)) then                                ▷ Full Inhibition
3:     inhibition_history = 1.0;
4:     return 1.0;
5:   else
6:     if (inhibition_history > 0.0) then                                                ▷ Smoothing to avoid inhibition lags
7:       inhibition_history -= inhibition_history_factor;
8:       return 1.0;
9:     else
10:      if actual_height ≤ (max_height - approximation_range -
11:        inhibition_range) then                                                        ▷ No inhibition
12:        return 0.0;
13:      else                                                                              ▷ Linear partial inhibition
14:        return (1.0 -  $\frac{\text{max\_height} - \text{inhibition\_range} - \text{actual\_height}}{\text{approximation\_range}}$ );

```

4 Experiments

The simulation environment from [20, 21] is used to test the implementation. Next to using artificial object primitives, also the ones for self-collisions were integrated. Figure 7a shows the self-collision model of the excavator and the model of the nearby truck. It is represented by three cubic objects that are forming the rough structures of the machine. The first covers the bottom of the machine with the wheels and stabilizers, the second forms the main torso, and the third represents the cabin. The digging-arm itself is not inserted as an obstacle, because a collision with each other is not possible from the structure itself. The truck is inserted as a cubic object. In Fig. 7b the workspace limiter is used with the implemented models. The planes represent workspace limitations for the single elements to prevent the boom from colliding with the truck.

Within the simulation the joints are always safely stopped before elements hit an obstacle. Tests inserting various objects close to the current position of the excavator boom were performed to evaluate the response time. In all cases the time between insertion and an active response of the system was between 40 and 110 ms. This time increases up to 450 ms until the simulated machine has come to a complete hold. For test cases where objects were inserted farther away from the current boom elements and out of the direct trajectory path, the obstacle avoidance lead to an online adaptation of the executed trajectories. In some of these cases the machine movement

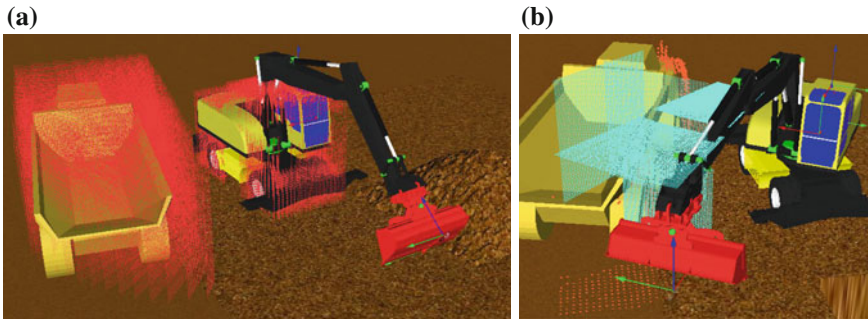


Fig. 7 (Self-)collision models of the excavator and the nearby truck inside the *octree* visualized as red pointclouds (a) and the obstacle avoidance system in action with these models visualized as turquoise distance planes (b)

was stopped as the goal was not reachable without farther leaving the current path. Due to numerical inconsistencies it could sometimes happen that the machine moves very slowly towards the obstacle if the distance to the goal is very large and is kept for a long time without replanning. An active obstacle surpassing strategy is needed in the future for these cases, which sets new intermediate goals, but was not part of this research work.

5 Conclusion and Future Work

Main goal of this work is the development of an obstacle avoidance system for the autonomous bucket excavator *Thor*. To efficiently calculate the behavior inhibitions, an adequate data structure is found and adapted to the given environment. Although other maps types could theoretically be used and *octrees* are not new, those solutions would either lead to a larger memory consumption or data retrieval times. Furthermore, to the author's knowledge there has been no research work using a cylindric one. In this case it is needed to utilize the benefits of a central-perspective system. The result is a universal obstacle avoidance system, that provides a functionality to prevent a collision of the digging-arm with the torso (*self-collision*). It is also possible to consider other environmental objects that are detected with sensor systems and can be defined as point clouds or geometrical objects (*environmental-collision*). Furthermore, it is possible to limit the digging workspace, for example to avoid the demolition of cables and lines or other objects in the ground (*workspace-collision*) which can not be directly detected. In addition the system can be extended and optimized easily for further requirements. It can also be adapted to other robots as long as they are containing a reactive control approach or a distinct safety layer.

There is still some work, that has to be done. When the obstacle detection system of the robot is completed, the obstacle avoidance should be tested with real

obstacles. To use the system on the real machine, it is also important to adjust the parameters and to do a series of testings, to validate the function of the system in a real environment. Extensions of the current implementation would be dynamic (self-)collision obstacle models based on dynamic calculations or an active-intervention system which would not only stop joints but actively move them around obstacles.

Acknowledgements The work at hand was funded from the Federal Ministry of Education and Research (BMBF) under grant agreement number 01IS13027D.

References

1. Albiez, J.: Verhaltensnetzwerke zur adaptiven Steuerung biologisch motivierter Laufmaschinen. GCA Verlag (2007)
2. Albiez, J., Luksch, T., Berns, K., Dillmann, R.: An activation-based behavior control architecture for walking machines. *Int. J. Robot. Res.* Sage Publications **22**, 203–211 (2003)
3. Arkin, R.: Behaviour-Based Robotics. MIT Press (1998). ISBN-10: 0-262-01165-4; ISBN-13: 978-0-262-01165-5
4. Brooks, R.: A robust layered control system for a mobile robot. *IEEE J. Robot. Autom.* **RA-2**(1), 14–23 (1986)
5. Brooks, R.: New approaches to robotics. Technical report, Artificial Intelligence Laboratory, MIT (1991)
6. Brooks, R.A.: Intelligence without reason. A.I. Memo No. 1293, Massachusetts Institute of Technology (MIT), Artificial Intelligence Laboratory (1991)
7. Crowley, J.L.: Navigation for an intelligent mobile robot. *IEEE J. Robot. Autom.* **1**(1), 31–41 (1985)
8. Giral, G., Chatila, R., Vaisset, M.: An integrated navigation and motion control system for autonomous multisensory mobile robots. In: *Autonomous Robot Vehicles*, pp. 420–443. Springer (1983)
9. Göckel, F.: Development of a behavior-based obstacle collision avoidance for the autonomous excavator thor. Bachelors thesis, Robotics Research Lab, Department of Computer Sciences, University of Kaiserslautern, October 2013
10. Kiebusch, L., Proetzsch, M.: Tutorial: Control system development using ib2c. Technical report, Robotics Research Lab, University of Kaiserslautern, 2011 unpublished
11. Meagher, D.: Octree encoding: a new technique for the representation, manipulation and display of arbitrary 3-d objects by computer. Technical report, Rensselaer Polytechnic Institute (1980)
12. Moravec, H.P.: The standford cart and the cmu rover. In: *Proceedings of the IEEE*, pp. 872–884 (1983)
13. Nilsson, N.J.: Shakey the robot. Technical Report 323, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, Apr 1984
14. Pluzhnikov, S., Schmidt, D., Hirth, J., Berns, K.: Behavior-based arm control for an autonomous bucket excavator. In: Berns, K., Schindler, C., Dreßler, K., Jörg, B., Kalmar, R., Zolynski, G. (eds.): *Proceedings of the 2nd Commercial Vehicle Technology Symposium (CVT 2012)*, pp. 251–261, Kaiserslautern, Germany, March 13–15 2012. Shaker Verlag (2012)
15. Proetzsch, M.: Development Process for Complex Behavior-Based Robot Control Systems. RRLab Dissertations. Verlag Dr. Hut, 2010. ISBN: 978-3-86853-626-3
16. Proetzsch, M., Luksch, T., Berns, K.: Fault-tolerant behavior-based motion control for offroad navigation. In: *Proceedings of the 20th IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4697–4702, Barcelona, Spain, April 18–22 2005

17. Proetzsch, M., Luksch, T., Berns, K.: The behaviour-based control architecture iB2C for complex robotic systems. In: Proceedings of the 30th Annual German Conference on Artificial Intelligence (KI), pp. 494–497, Osnabrück, Germany, September 10–13 2007
18. Reichardt, M., Föhst, T., Berns, K.: On software quality-motivated design of a real-time framework for complex robot control systems. In: Proceedings of the 7th International Workshop on Software Quality and Maintainability (SQM), in conjunction with the 17th European Conference on Software Maintenance and Reengineering (CSMR), Genoa, Italy, March 5 2013
19. Schmidt, D.: Dynamic modelling and control of a mobile bucket excavator. Project thesis, Robotics Research Lab, Department of Computer Sciences, University of Kaiserslautern, September 11 2008 (unpublished)
20. Schmidt, D., Proetzsch, M., Berns, K.: Simulation and control of an autonomous bucket excavator for landscaping tasks. In: Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA 2010), pp. 5108–5113, Anchorage, Alaska, USA, May 3–8 2010
21. Wettach, J., Schmidt, D., Berns, K.: Simulating vehicle kinematics with simvis3d and newton. In: 2nd International Conference on Simulation, Modeling and Programming for Autonomous Robots, Darmstadt, Germany, 15–18 November 2010

An Integrated Robotic System for Autonomous Brake Bleeding in Rail Yards

**Huan Tan, Shiraj Sen, Arpit Jain, Shuai Li, Viktor Holovashchenko,
Ghulam Baloch, Omar Al Assad, Romano Patrick, Douglas Forman,
Yonatan Gefen, Pramod Sharma, Frederick Wheeler,
Charles Theurer and Balajee Kannan**

Abstract Current operations in rail yards are dangerous and limited by the operational capabilities of humans being able to perform safely in harsh conditions while maintain high productivity. Such issues call out the need for robust and capable autonomous systems. In this paper, we outline one such autonomous solution for the railroad domain, capable of performing the brake bleeding inspection task in a hump yard. Towards that, we integrated a large form factor mobile robot (the Clearpath Grizzly) with an industrial manipulator arm (Yasakawa Motoman SIA20F) to effectively detect, identify and subsequently manipulate the

H. Tan · S. Sen · A. Jain · V. Holovashchenko · G. Baloch · O. Al Assad · R. Patrick ·
D. Forman · Y. Gefen · F. Wheeler · C. Theurer · B. Kannan (✉)
GE Global Research, Niskayuna, NY, USA
e-mail: balajee.kannan@ge.com

H. Tan
e-mail: huan.tan@ge.com

S. Sen
e-mail: shiraj.sen@ge.com

A. Jain
e-mail: arpit.jain@ge.com

V. Holovashchenko
e-mail: viktor.holovashchenko@ge.com

G. Baloch
e-mail: baloch@ge.com

O. Al Assad
e-mail: omar.alassad@ge.com

R. Patrick
e-mail: romano.patrick@ge.com

D. Forman
e-mail: douglas.forman@ge.com

Y. Gefen
e-mail: yonatan.gefen@ge.com

brake lever under harsh outdoor environments. In this paper, we focus on the system design and the core algorithms necessary for reliable and repeatable system execution. To test our developed solution, we performed extensive field tests in a fully operational rail yard with randomly picked rail cars under day and night-time conditions. The results from the testing are promising and validate the feasibility of deploying an autonomous brake bleeding solution for railyards.

Keywords Field robotics • Autonomous system • 3D detection and segmentation • Deep learning • Dynamic environment • Manipulation • Trajectory planning • System design • Rail yard operations • Brake bleeding

1 Introduction

The challenges in the modern rail yard are vast and diverse. Railroad classification yards, or hump yards, play an important role as consolidation nodes in rail freight networks. At classification yards, inbound trains are disassembled and the railcars are sorted by next common destination (or block) [1]. Based on various studies for the rail industry, the efficiency of yards in part drives the efficiency of the entire rail network [2]. The hump yard is generally divided into three main areas: the receiving yard, where inbound trains arrive and are prepared for sorting, the class yard, where railcars are sorted into blocks and the departure yard, where blocks are assembled into outbound trains, inspected and then depart.

In the future, rail yards will be a hub of technological advancements where autonomous mobile robotic solutions work in coordination with human yard operators and cloud-based data analytics to enhance human safety [3] and drive services productivity (see Fig. 1).

Current solutions for field service operations are labor-intensive, dangerous, and limited by productivity and cost. Furthermore, efficient system level-operations require integrated system wide solutions, more than just point solutions to key challenges [4]. The nature of these missions dictates that the tasks and environments cannot always be fully anticipated or specified at the design time, yet an

F. Wheeler
e-mail: wheeler@ge.com

C. Theurer
e-mail: theurer@ge.com

S. Li
Rensselaer Polytechnic Institute, Troy, NY, USA
e-mail: fightever2010@gmail.com

P. Sharma
Microsoft Corporation, Seattle, WA, USA
e-mail: Sharma.pramod@ge.com

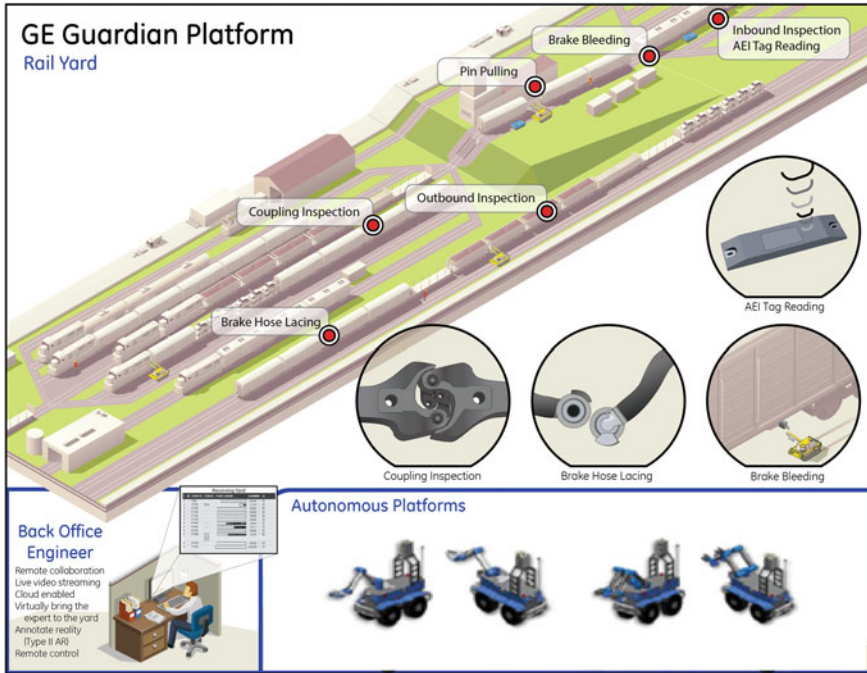


Fig. 1 GE guardian platform for autonomous railyard inspection services

autonomous solution must have the essential capabilities and tools to carry out the mission even if it encounters situations that were not expected.

Solutions for typical railyard problems, such as brake bleeding, brake line lacing, coupling cars, etc., require combining mobility, perception, and manipulation towards a integrated autonomous solution which can be tightly integrated with railroad yard enterprise system. Currently, there exist a number of mobile manipulation systems that have been successfully demonstrated in controlled indoor environments. An exhaustive survey of such systems and techniques is beyond the scope of this paper. Ciocarlie et al. [5] showed how a PR2 robot can utilize a 3D sensor for object detection and manipulation in cluttered tabletop environments. The Intel HERB mobile manipulation platform has demonstrated impressive capabilities for picking up and placing of objects in indoor environments [6]. Nguyen et al. [7] presented the EI-E robot that could autonomously fetch objects from flat surfaces. All these platforms make certain assumption with regards to the environment in order to simplify the problem of robot perception. When placing robots in an outdoor environment, technical challenges increase largely, but field robotic application benefits both technically and economically [8]. Some researchers have used robots in the domains of agriculture [9, 10], mining [11, 12], transporting [13, 14], etc. The DARPA Robotics Challenge (DRC), aimed at advancing the capabilities of robotic systems for disaster response, is one of the few examples of robots working in unstructured outdoor environments [15]. This challenge required a robot to collaborate with humans in order to achieve varied



Fig. 2 Typical cars and brake levers

task objectives such as opening a door, turning a valve, and drilling a hole. The difficulty associated with detecting objects reliably in outdoor environment results in the problem of object detection primarily being performed by a human operator, fitting models to sensory data [16].

One key challenge in yard operation is that of bleeding brakes on inbound cars in the receiving yard. Railcars have pneumatic breaking systems that work on the concept of a pressure differential. Figure 2 displays different types of brake lever robots need to grasp and pull. Interestingly, the size of the brake lever is significantly small compared to the size of the environment and the rail cars. In addition, there are lots of variations on the shape, location, and the material of the brake levers. Over 1.4 million rail cars in the US which circulate throughout the system; brake levers are subject to damage and repair and being moved into different orientations and poses. Coupled with that is the inherent uncertainty in the environment; every day, rail cars are placed at different locations, and the spaces between cars are very narrow and unstructured.

In our vision, we believe that intelligent autonomy, robust perception, and robust actuation are the keys to a successful autonomous robotic system. In this paper we outline one such autonomous solution for performing the brake bleeding task in a rail yard environment. We believe our solution is one of the first of its kind in automating the brake bleeding process for rail yards, addressing a real need for robotics in the domain of industrial applications.

The rest of the paper is organized as follows: Sect. 2 explains the system design in detail, Sect. 3 discusses our experimental results, and Sect. 4 summarizes the results and identifies future work scope.

2 System Design

The developed robotic system is used to actuate brake levers on rail yards, which is a typical field operation. The system involves the robot autonomously navigating within the track corridor along the length of the train moving from car to car (given an initial coarse estimate of the brake rod location from rail database), locating the

brake rod, positioning itself next to the brake rod before actuating the brake rod. During the autonomous navigation, the robot needs to maintain a distance of separation (4") from the plane of the railcar while moving forward. In order to ensure real-time brake rod detection and subsequent estimation of the brake rod location, a two-stage detection strategy is utilized. Once the robot has navigated near to the brake rod location an extremely fast 2D vision-based search algorithm confirms a coarse location of the brake rod. The second stage of the algorithm involves building a dense model for template-based shape matching (brake rod) to identify the exact location and pose of the break rod. Once the robot reaches the desired location, the final step in the process is to actuate and manipulate the rod with a robotic arm. The manipulation solution needs to be flexible enough to handle variations in the environment, the location and state of the rod and in sensing and perception. The detail of the grasping strategy is beyond the scope of this paper.

2.1 System Architecture

The brake bleeding architecture is composed of three layers: Physical Layer, Processing Layer, and Planning Layer (see Fig. 3). The physical layer deals with the robots and sensors. For the physical layer, we leverage commercially available sensors and platforms integrating to build a cohesive and highly capable platform. The details of the actual platform are described later in Sect. 3.

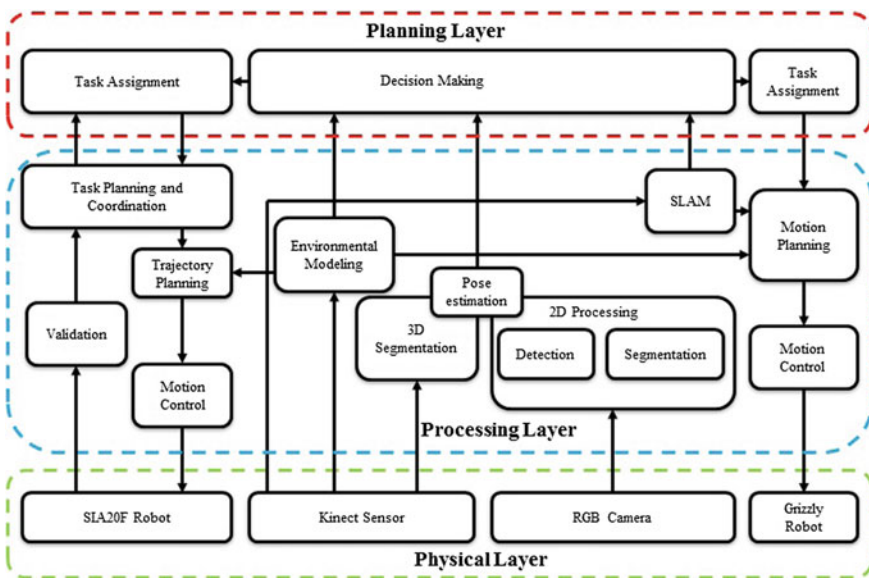


Fig. 3 System architecture

The processing layer is the crucial part of the system, since most of the algorithms and functions are implemented on this layer. Based on the fields of study, we categorize the major components into four modules: *Deliberation*, *Perception*, *Navigation*, and *Manipulation*. The deliberation module is responsible for planning and coordinating all the behaviors in the system. It gets processed information from sensors and makes decisions to move the Grizzly robot and the SIA20F robot. The perception module processes sensory information and finds the poses of brake levers in the environment. The navigation module controls the Grizzly robot to move to desired position. In order to move safely and precisely, a variant of RTAB-Map [17] algorithm based on Environmental Modeling is implemented to provide necessary information for motion planning. Manipulation module is used for controlling the SIA20F robot to touch the lever. Environmental Modeling is used to help the robot to plan the motion trajectories while avoiding collision.

Finally, in the planning layer, all sensory information and states information are collected from lower layers towards action selection. According to requirements of a task, the system will make different decisions based on current task-relevant situation. A state machine ties all the layers together and transfer signals from navigation to perception and then subsequently manipulation. Whenever there is an emergency stop signal generated or there are error information reported by any of the three sub-modules, the system safety primitives are triggered, preventing any damages to both the robot and the environment.

2.2 *Environmental Modeling and Platform Motion Control*

Moving the platform between different locations in rail yards to perform tasks is a basic requirement for mobile manipulation systems. We provide two operation modes in the navigation module: one is the teleoperation mode, and the other is autonomous mode. Using the triggered signal from a joy stick, the deliberation module sends a command to the navigation module to switch the mode. In the autonomous mode, the robot moves to search brake levers in the environment. The navigation module is tightly coupled with the detection module, moving the robot in a trajectory parallel to the plane of the railcar until a lever is found. Figure 4 displays the state machine for the navigation module.

Environmental Modeling

In order to move in the environment, the robot needs to model and understand the environment. Currently, the robot motion is limited to being able to navigate the length of a train (typically about 100 rail cars long) and subsequently does not need to move long distance, which means global planning is not required. (The ultimate goal is to move the robot autonomously along 100 rail cars.) However, even in the local planning and movement, the robot needs to safely operate in its environment. Towards that, we implemented a 3D point cloud-based SLAM algorithm in our system. There are lots of SLAM algorithms off-the-shelf and we

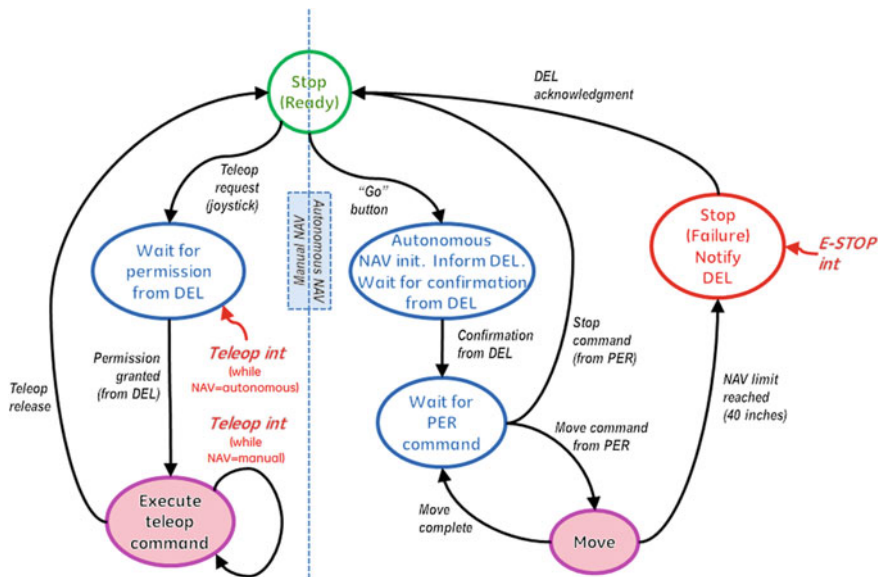


Fig. 4 Navigation state machine

choose Real-Time Appearance-Based Mapping (RTAB-Map) [17], which is a RGB-D Graph-Based SLAM approach based on an incremental appearance-based loop closure detector. In our testing, we found it to be very stable under most of the weather conditions. Using RTAB-Map, the robot knows the relative location of itself, which is used to close the motion control loop. At the same time, we use point cloud data to recognize the plane of the cars. This information is used to keep the robot away from the rail cars and maintain a pre-defined distance of separation (see Fig. 5).

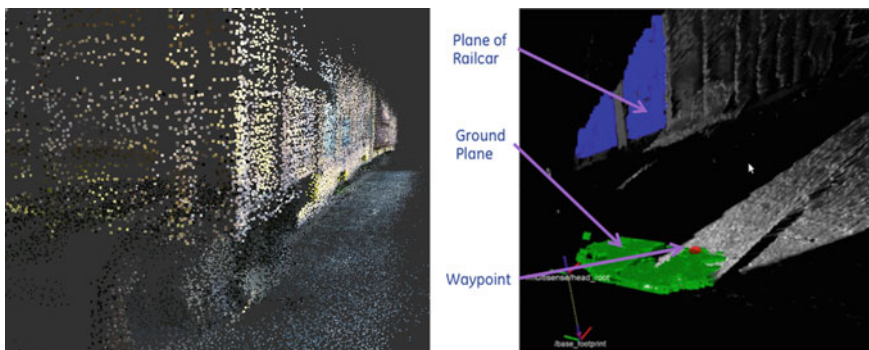


Fig. 5 Point cloud building and subsequent waypoint generation for navigation

Motion Planning and Control

There are two planning modules implemented in our system. One is to minimize the distance between the robot and the desired location, and the other is to keep a safe distance between the robot and objects in the environment, including the tracks, keep out zones, geo fences, humans, etc. The computed control commands from the goal controller for driving the robot forward and from the safety controller for keeping the robot away from rail cars are fused to compute a velocity command which will control the motors of the robot to move in the environment using the Artificial Potential Field [18] algorithm.

2.3 Target-of-Interest Detection and Pose-Estimation [19]

Detecting and finding the correct 6D pose (3D position + 3D orientation) of the correct brake lever is a necessary first step to actuating the brake lever. Our approach for object detection in unconstrained environments relies on obtaining object hypothesis candidates by fusing detection results from 2D images and 3D point clouds. The candidates are then combined temporally and reasoned upon in an online fashion by using real time Simultaneous Localization and Mapping (SLAM). We propose a confidence function that ranks the candidate detections based on its detection score, spatial, and temporal consistency. The confidence function also takes into account the uncertainty in detection location due to occlusion or SLAM misalignment. Figure 6 shows the pipeline of our approach.

Image Based Object Detector

Rail cars and brake levers are different, and they may change due to damage, repairing, and daily use, such that the shapes and poses are different. The deployment of new rail cars raises more challenges for robots. In our system, we use deep learning to train an object detector from images, which enables robot to detect and segment brake levers using the trained model, even when the brake lever is not included in our training model. Our detector utilizes the layered architecture of deep CNN [20] to learn the object representation. Each layer comprises of linear

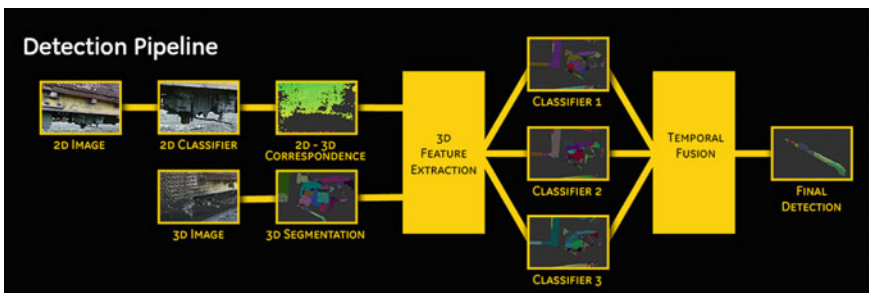


Fig. 6 Detection and pose estimation

and non-linear operators that are learned jointly in an end-to-end system using a task specific loss function.

The parameters of CNN are learned using back-propagation to reduce the overall classification error [21].

We used a ImageNet model [22] trained on the AlexNet architecture. Deep learning detectors have high precision provided there is enough training data. However their runtime performance is slow due to the high computational requirements of the multilayered neural network. In this paper, a Deformable Part-based Model (DPM) [23] detector is used to filter the image and reduce the number of object candidates being passed to the neural network. A DPM learns the appearance and spatial arrangement of the object and its parts using histogram of oriented gradient features. An Ensemble of Shape Functions (ESF) [24] is extracted from each of the 3D segments. Three histograms are generated from each of these shape functions, including the histogram of the connecting lines generated from random points that lie on the object surface, the histogram of connecting lines that do not lie on the object surface, and the histogram for the case when part of the connecting lines lie on the object surface. The 3D feature descriptor comprising of multiple histograms is encoded into a single high dimensional feature descriptor by using Fisher Vector encoding. Two sets of classifiers were trained using Support Vector Machines (SVM) [25] and Real Ad-Boost. Multiple classifiers allow us to leverage strengths of individual classifier to obtain a robust score for detection.

3D Segmentation

Object proposals from 3D point cloud are obtained through segmentation process. We use Locally Convex Connected Patches (LCCP) [26] algorithm to segment the 3D point cloud [27]. LCCP is a bottom up approach that merges super voxels into object parts based on a local convexity/concavity criterion. A basic filtering step based on size and shape constraints are applied to these segments to reduce the object hypothesis space.

Temporal Fusion

A voting scheme is utilized for hypothesis evaluation, wherein each hypothesis votes for the position of the object.

2.4 Trajectory Planning and Actuation

The ultimate goal is to actuate (touch) brake levers on rail cars. After the detection is done, the robot knows where the target of interest, i.e., the brake lever, is. Using the pose estimation results, the robot plans a motion trajectory in the workspace based on the current environmental modeling results. Then the controller drives the arm safely to touch the brake lever. Figure 7 displays the state machine of manipulation.

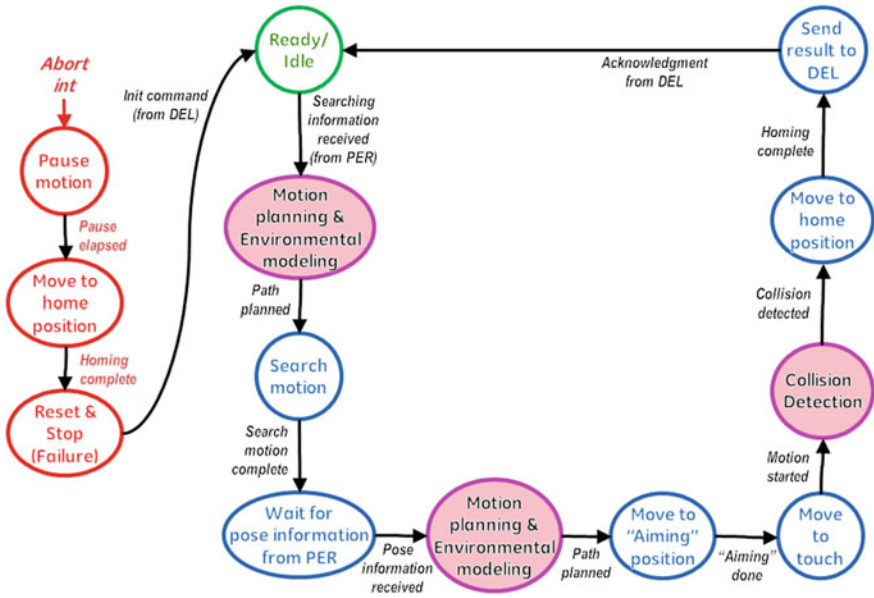


Fig. 7 Manipulation state machine

Environmental Modeling

Our system is used to touch brake levers on rail cars, which means the majority of the motions are “moving toward”. In our testing, we found that grid-based algorithm is enough for modeling our environment.

The environmental data is collected using Kinect. All the point cloud data points are processed and grouped into a grid, i.e., OctoMap [28]. Figure 8 displays an example of modeling the environment.

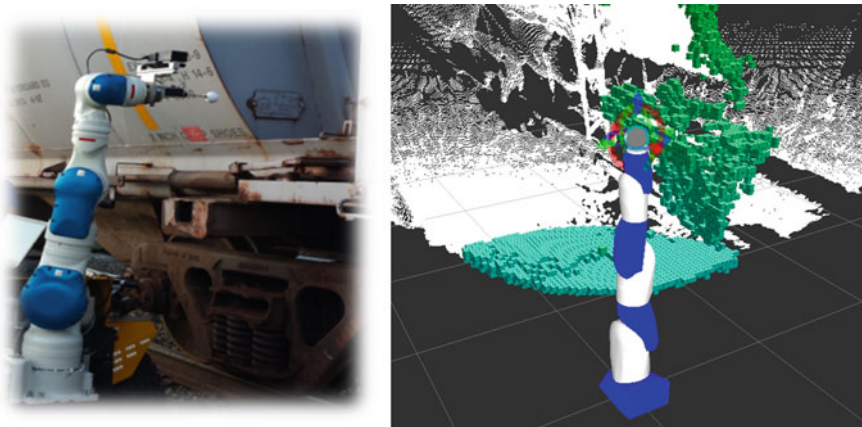


Fig. 8 Environmental modeling

Trajectory Generation

Two requirements are set up for the motion planning algorithm: 1. Fast; 2. Safe. We used the Robot Operating System (ROS) [29] Motoman Package [30] developed by ROS-Industrial Consortium as the basic motion planning driver and created three ROS services, including linear trajectory planning in joint space, linear trajectory planning in Cartesian space, and Point-to-Point trajectory planning in joint space, for motion planning in our system.

Linear Trajectory Planning in Joint Space

When the robot is moving in an open space for a long distance and far away from the cars and brake levers, we do not need to worry about the motion patterns. Then we define the starting position and target position of the motion. Using Artificial Potential Field algorithm, we can find the way points on the desired motion, which is linear in all 6 degrees of freedom, but non-linear in the Cartesian space. Then we assign velocities to each way points depending on the task requirements.

When integrated with the deliberation module, we cannot use this method directly. Because the positions of the brake levers are defined as 6D poses in Cartesian space. After get the 6D poses in the Cartesian space, using Inverse Kinematics, we convert the 6D pose from Cartesian space to 6 joint angles in the joint space. Then this service can compute the desired joint angles on the motion trajectory.

Linear Trajectory Planning in Cartesian Space

This service is an enhanced version of the first service. We use Artificial Potential Field algorithm to find the way points on a desired motion trajectory in the Cartesian space. Using Inverse Kinematics, we get corresponding way points in the joint space. Then we assign velocities to all the way points.

Point-to-Point Trajectory Planning in Joint Space

This service is created for homing or resetting. We just defined the target joint angles without any internal way points. This method could bring the robot to the target position as fast as possible.

Motion Control

The generated motion trajectories are defined as a sequence of way points in the joint space. Each way point contains the information of 7 joint angles, timing stamp, and velocity. The joint angles, timing stamp, and velocity are put in the vector of points. The standard MotoROS program is run on the robot to receive the command to drive the robot to move along the desired motion trajectory.

Validation

A microswitch produced by Honeywell is attached to the wrist of the SIA20F robot. And the other end is a white ball. Whenever the ball touches the brake lever, the microswitch will be triggered to provide feedback information to the robot.

3 Experimental Results and Discussion

The proposed design was developed and validated on a practical mobile robotic platform. Figure 9 displays the created system. We use a Clearpath Grizzly mobile robot as our base platform integrated with a 7-degree of freedom Yasakawa Motoman SIA20F manipulator arm coupled with a custom end effector. Towards environmental modeling and brake lever detection, we used two RGB cameras for 2D processing including detection and segmentation in conjunction with two Kinect sensors. The point cloud information is also used for mapping and localization for the mobile platform.

Two Lenovo W541 laptops, which are equipped with Intel Core i7-4910MQ 2.90GHZ GPU and 32.0G RAM Memory, are used in our system. One is responsible for motion control of the mobile platform, detection and pose-estimation of targets, and high level state machine. The other is majorly used for controlling the SIA20F robot to plan the motion trajectories and physically touch brake levers. Robot Operating System (ROS) is used as the low-level IPC for our system.

3.1 Detection Results

The performances of the image-based detectors were evaluated on 2000 color images collected from 80 test rail cars. Figure 10 shows the performance of the two image-based detectors. In our approach, with SLAM incorporated, our system achieved 85 % detection performance. This demonstrates that temporally fusing detection results significantly improve object detection accuracy. Moreover, our perception system performed equally well on detection task during the day and the night time validating that our approach is robust to environmental and illumination changes.

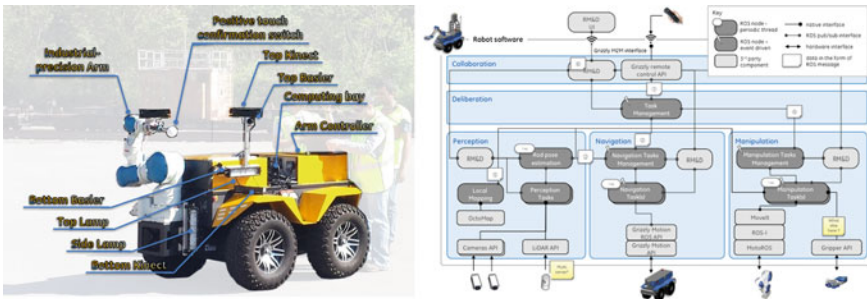


Fig. 9 Development platform (hardware and software centric views)



Fig. 10 Test environment (GE Railcar Repair Facility) and test conditions (day and night)

3.2 Motion Planning Results

After we successfully find a brake lever, we need to plan a trajectory for the SIA20F to touch the lever. When the environment is open and there are no obstacles in the environment, as long as the brake lever is in the reachable space of the manipulator, the trajectory planning is 100 % successful.

3.3 System Testing Results

In GE’s Railcar repair facility in Sayre, PA (see Fig. 11), we evaluated our system on 61 rail cars over multiple days and times (see Fig. 11) to account for varying

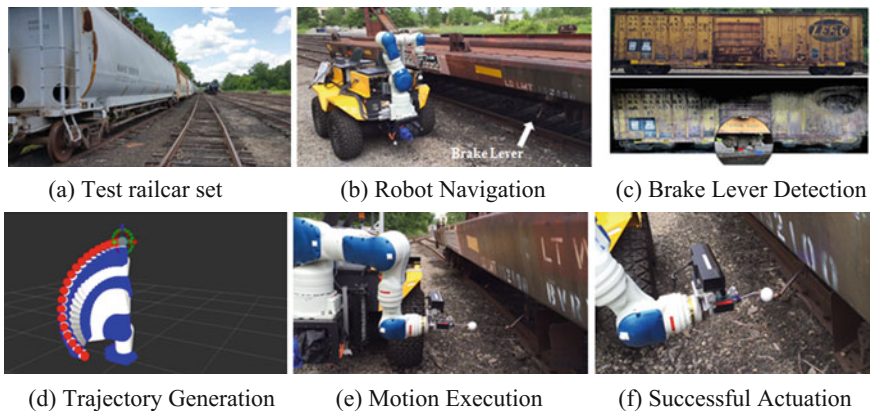


Fig. 11 Autonomous brake bleeding operations—from left to right; the robot is lined next to a set of railcars, navigates to the first one, detects the brake lever, moves into position adjacent to the brake lever, calculates appropriate trajectory and actuates brake lever. **a** Test railcar set, **b** Robot navigation, **c** Brake lever detection, **d** Trajectory generation, **e** Motion execution, **f** Successful actuation

environment conditions. For our tests, we randomly picked cars in the rail with the distribution of the cars ranging from hoppers to tankers, box and gondolas. For testing, we restrict the search distance to about 2.5 m. A test is considered as a success if the robot touches the designated brake lever, otherwise it is a failure. We randomly picked rail cars from our rail yard and tested 61 times.

Figure 11 displays the scenario of the testing process: (a) the robot is placed in a rail yard, where it needs to handle the brake levers along a rail track, (b) the robot is initially placed on a location about 2.5 m away from the brake lever and it moves along a rail track, (c) the robot keeps detecting the brake lever in the environment using visual information, (d) the robot stops at a location and plans a motion trajectory to move the robot manipulator to touch the brake lever, (e) the arm moves toward the brake lever, and (f) touches the brake lever without hitting obstacles in the environment. After the lever is touched, the robot moves to the next location.

Raw Testing Results

In 61 trials, our system successfully touched the brake levers 41 times. The overall success rate of touching is 67.21 %. Table 1 displays the original data collected.

Adjusted Testing Results

In the 20 failed trials, we categorized the cause of failure into four types: Detection Error, Navigation Error, Manipulation Error, and Human Operation Error. Specifically, the Human Operation Error includes the wrong placement of the robot, initialization is not correct, sensor is not turned on, etc. The original distribution of the causes of errors is displayed in Table 2.

Detection errors and Manipulation errors are produced due to technical reasons. For example, the detection module may find a wrong lever, which is a positive false error; the detection module may not be able to find any levers in the environment, which is a negative true error. Sometimes, lighting may affect the quality of the

Table 1 Original testing results

Error type	Occurrence
1. Perception—Not found	1
2. Perception—Wrong detection	7
3. Perception—Back to prior detection	0
4. Perception—Undefined	1
5. Navigation	1
6. Navigation—Motor malfunction	4
7. Human error—Positioning	3
8. Human error—Undefined	1
9. Manipulation—Obstacle collision	0
10. Manipulation—Motion incomplete	1
11. Other	1
	Total errors: 20
	Overall success rate: 67.21 %

Table 2 Error distribution

Trials	Categorization	Percentage (%)
Success	Successful	67.21
Failure	Detection error	14.75
	Navigation error	8.20
	Manipulation error	1.64
	Human operation error	8.20

Table 3 Adjusted testing results

Trials	Categorization	Percentage (%)
Success	Successful	81.97
Failure	Detection error	14.75
	Navigation error	1.64
	Manipulation error	1.64
	Human operation error	0.0

images, especially when testing is conducted in strong sunlight. However, the human operation error can be removed after the users are well trained and eventually this part of the human operation will be automated. 4 navigation errors are produced by the malfunction of the motors. If we replace those motors, the errors can also be removed. Then we remove the human operation errors and 4 navigation errors, the adjusted result is displayed in Table 3.

4 Conclusion

This paper proposes an integrated system design of deploying mobile robots in rail yards for autonomous operations. We integrate two robots to perform autonomous brake bleeding in rail yards and the system design is explained in detail in this paper. The developed system has been tested in a rail yard and the experimental results validate that our system can successfully navigate, find, and touch brake levers. As mentioned earlier, this is an on-going project. The next step is to develop grasping and pulling capabilities for the industrial manipulator. The system will be tested more in more rail yards under different conditions.

References

1. https://en.wikipedia.org/wiki/Rail_yard
2. Oum, T.H., Waters, W.G., Yum, C.: A survey of productivity and efficiency measurement in rail transport. *J. Transp. Econ. Policy* 9–42 (1999)
3. Drudi, D.: Railroad-related work injury fatalities. *Monthly Lab. Rev.* **130**, 17 (2007)

4. Thorpe, C., Durrant-Whyte, H.: Field robots. In: Proceedings of the 10th International Symposium of Robotics Research (ISRR'01) (2001)
5. Meeussen, W., Wise, M., Glaser, S., Chitta, S., McGann, C., Mihelich, P., Marder-Eppstein, E.: Autonomous door opening and plugging in with a personal robot. In: Proceedings of 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 729–736 (2010)
6. Srinivasa, S.S., Ferguson, D., Helfrich, C.J., Berenson, D., Collet, A., Diankov, R., Gallagher, G., Hollinger, G., Kuffner, J., Weghe, M.V.: HERB: a home exploring robotic butler. *Auton Robots* **28**(1), 5–20 (2010)
7. Nguyen, H., Anderson, C., Trevor, A., Jain, A., Xu, Z., Kemp, C.C.: El-E: an assistive robot that fetches objects from flat surfaces. In: Proceedings of the 2008 International Conference on Human-Robot Interaction (2008)
8. Pedersen, S.M., Fountas, S., Have, H., Blackmore, B.S.: Agricultural robots: an economic feasibility study. *Precis. Agric.* **5**, 589–595 (2002)
9. Blackmore, B.S.: A systems view of agricultural robots. In: Proceedings 6th European Conference on Precision Agriculture (ECPA), pp. 23–31 (2007)
10. Redhead, F., Snow, S., Vyas, D., Bawden, O., Russell, R., Perez, T., Brereton, M.: Bringing the farmer perspective to agricultural robots. In: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems, pp. 1067–1072 (2015)
11. Green, J.: Underground mining robot: a CSIR project. In: Proceedings of 2012 IEEE International Symposium on Safety, Security, and Rescue Robotics, pp. 1–6 (2012)
12. Skonieczny, K., Delaney, M., Wettergreen, D.S., “Red” Whittaker, W.L.: Productive lightweight robotic excavation for the moon and mars. *J. Aerosp. Eng.* **27**(4) (2013)
13. Jensen, M.A., Falk, D.B., Sørensen, C.G., Blas, M.R., Lykkegaard, K.L.: In-field and inter-field path planning for agricultural transport units. *Comput. Ind. Eng.* **63**(4), 1054–1061 (2012)
14. Suessemilch, I., Rohrer, C., Roesch, R., Guenther, C., Von Collani, Y., Linder, S., Fischer, V.: Projection unit for a self-directing mobile platform, transport robot and method for operating a self-directing mobile platform. U.S. Patent Application 14/447,501, filed 30 July 2014 (2014)
15. Pratt, G., Manzo, J.: The DARPA robotics challenge [competitions]. *IEEE Robot. Autom. Mag.* **20**(2), 10–12 (2013)
16. Johnson, M., Shrewsbury, B., Bertrand, S., Tingfan, W., Duran, D., Floyd, M., Abeles, P.: Team IHMC’s lessons learned from the DARPA robotics challenge trials. *J. Field Robot.* **32**(2), 192–208 (2015)
17. Labbe, M., Michaud, F.: Online global loop closure detection for large-scale multi-session graph-based slam. In: Proceedings of 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2661–2666 (2014)
18. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **5**(1), 90–98 (1986)
19. Li, S., Jain, A., Sharma, P., Sen, S.: Robust Object Detection in Industrial Environments by Using a Mobile Robot (2016)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 1097–1105 (2012)
21. LeCun, Y., Bengio, Y.: Convolutional networks for images, speech, and time series. In: *The Handbook of Brain Theory and Neural Networks* 10 (1995)
22. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
23. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: Proceedings of 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)

24. Wohlkinger, W., Vincze, M.: Ensemble of shape functions for 3D object classification. In: Proceedings of 2011 IEEE International Conference on Robotics and Biomimetics, pp. 2987–2992 (2011)
25. Suykens, J.A.K., Vandewalle, J.: Least squares support vector machine classifiers. *Neural Process. Lett.* **9**(3), 293–300 (1999)
26. Stein, S., Worgotter, F., Schoeler, M., Papon, J., Kulvicius, T.: Convexity based object partitioning for robot applications. In: Proceedings of 2014 IEEE International Conference on Robotics and Automation, pp. 3213–3220 (2014)
27. Rusu, R.B., Cousins, S.: 3D is here: Point cloud library (PCL). In: Proceedings of 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 1–4 (2011)
28. Wurm, K.M., Hornung, A., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: a probabilistic, flexible, and compact 3D map representation for robotic systems. In: Proceedings of the 2010 ICRA Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation, vol. 2 (2010)
29. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source Robot Operating System. In: Proceedings of 2009 ICRA Workshop on Open Source Software, vol. 3, no. 2, p. 5 (2009)
30. http://wiki.ros.org/motoman_driver

Part III
Flying Robot

I Believe I Can Fly—Gesture-Driven Quadrotor Control Based on a Fuzzy Control System

Leon Spohn, Marcus Bergen, Nicolai Benz, Denis Vonscheidt, Hans-Jörg Haubner and Marcus Strand

Abstract Conventionally, drones and robots are controlled using joysticks and remote controls. Here is a new idea: to approximate the feeling of “being the bird”, we introduce 3D fly-gestures and add VR-technology to stimulate our eyes. We chose a quadrotor and the combination of a Kinect, Virtual Reality Glasses and a Fuzzy-Controller as interface for processing instructions.

Keywords Human-Maschine-Interface · Gesture Control · Fuzzy Control

1 Introduction

Remotely controlled aerial systems are a growing matter for industrial and private use. It brings along a great variety of future driven ideas. Especially drones are starting to find their way into many areas of today’s daily routines. Branches that currently make use of that technology are e.g. the military, industrial applications for delivery or reconnaissance. Or some are just plane lovers and use them for fun. What might be fascinating about it the most, is the feeling of freedom in movement and just experiencing the bird eyes view. In this article we propose a system for being the bird instead of only having a bird eyes view, so you really believe you can fly.

2 The Technology

While gesture control had its focus on touch gestures [1, 2] or gestures with hand attached sensors [2] more and more applications using 3D gestures are visible. Mainly enabled by the introduction of the Kinect RGBD sensor, the community

L. Spohn · M. Bergen · N. Benz · D. Vonscheidt · H.-J. Haubner · M. Strand (✉)
DHBW, Karlsruhe, Germany
e-mail: strand@dhbw-karlsruhe.de



Fig. 1 Project setup overview

developed a large set of 3D gesture control applications [3–5]. Controlling a quadrotor deserves a new set of hand/body gestures together with intelligent processing.

The quadrotor originates from the helicopter family, but is carried and driven by a system of four rotors. All rotors operate on the same level and create a stream (vertical to themselves), which results in the stable flight behavior of the aircraft. Furthermore, each rotor can be addressed individually, making the quadrotor capable of spinning on the spot or tilting in any direction. Today they are used in the same areas as conventional drones, although they almost have the properties of a helicopter. The project setup is as follows (see Fig. 1):

- The quadrotor: Parrot AR.drone 2.0. It provides a wifi network to which the computer connects.
- For this project, a Windows system of version 7 or higher is necessary. The computer should also include a wireless LAN adapter, an HDMI slot and a USB port.
- The next part is an XBOX Kinect device, connected to the mentioned USB port. The Kinect is for recording our gesture movement.
- For visualizing the drone's camera pictures, a pair of Zeiss VR-Goggles is linked to the HDMI slot.
- Another important part for the whole project is the Kinect SDK (Software Development Kit) provided by Microsoft.

3 Kinematics

Kinematics is the science of describing an object's movements through space. In order to explain the drones Kinematics, we need to look at the fact that the four rotors blowing down the air cause a thrust, and thus a propulsive force. In order to avoid self-rotation, the rotors spin pairwise into opposite directions (Figs. 2 and 3).

Fig. 2 Spinning directions

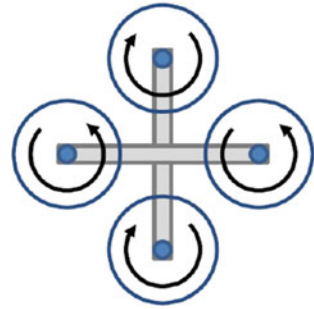
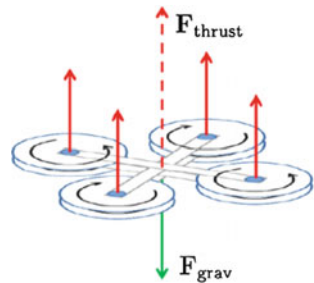


Fig. 3 Force vectors



The quadrotor can be navigated by energizing each rotor with a certain intensity. To let the drone hover on one spot, all rotors must have the same thrust and $F_{thrust} = -F_{grav}$. In order to let the drone ascend, all rotors must increase their thrust consistently and $F_{thrust} > F_{grav}$. In order to let the drone descend, all rotors must decrease their thrust consistently and $F_{thrust} < F_{grav}$.

To let the drone tilt and fly into a certain direction, the rotor of the tilting direction must decrease thrust and the opposing rotor must proportionally increase thrust. The other two rotors must have equal thrust. To turn right on the spot, the pair of rotors spinning right must have greater thrust than the ones spinning left and two rotors of a pair must have same thrust. To turn left on the spot, the pair of rotors spinning left must have greater thrust than the ones spinning right and two rotors of a pair must have same thrust.

The total rotation is defined by the rotation energy of each single rotor:

$$\tau = \tau_1 - \tau_2 + \tau_3 - \tau_4 = \tau_1 + \tau_3 - (\tau_2 + \tau_4)$$

4 Fuzzy-Controller

The fuzzy-controller operates the drone movement based on the data it receives from the Kinect. Its purpose is to translate it into steering instructions and to optimize the flight behavior. It is based on Lotfi Asker Zadeh’s Fuzzy-Set-Theory

[6], which is basically the control of a system being described by non-strict rules. There are 3 stages of fuzzy control: Fuzzification, Inference and Defuzzification. Those will be explained in the following.

4.1 Fuzzification

The fuzzification's purpose is to turn precise measured values into rather linguistic terms. Therefore, we define a membership function μ for every linguistic term. μ describes the affiliation of a value to the linguistic term in a range from 0 to 1.

$$\mu_A: U \rightarrow [0, 1]$$

The various membership functions belong to a linguistic variable (describing the input variables logically). Thus, sharp values become vague descriptions, which build the basis for inference. See Fig. 4 for in- and output variables. Figure 5 depicts a sample membership function.

4.2 Inference

Within the step of inference, the result of fuzzification is being evaluated and a conclusion is drawn. This happens based on previously defined rules. These rules have the following form: IF "premise" THEN "conclusion". The premise is a requirement for a rule to be fulfilled and the conclusion is the result of the rule. E.g.:

Rule R: IF $x = A$ THEN $y = B$
 Factum: x has more or less the property A
 Conclusion: y has more or less the property B

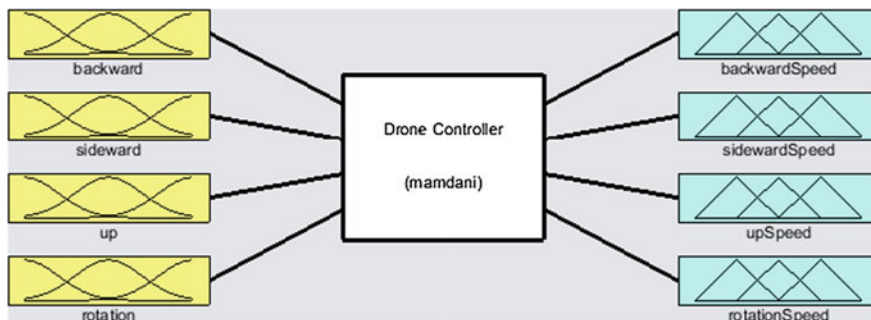


Fig. 4 In- and output variables

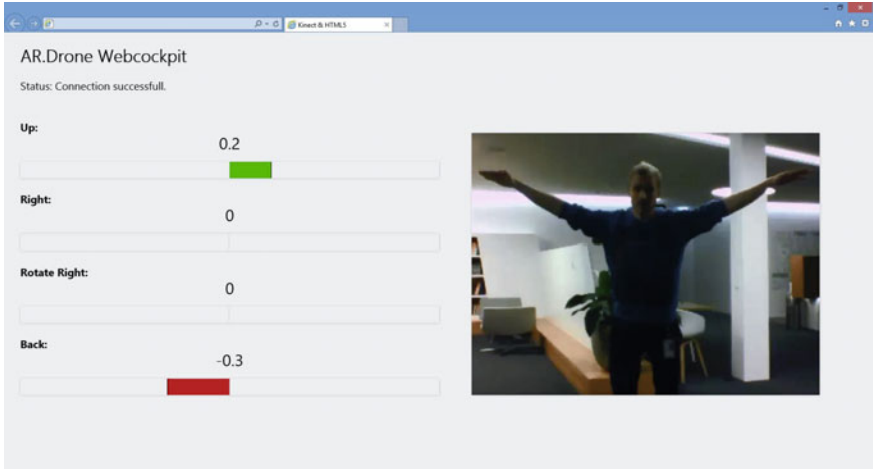


Fig. 5 AR.Drone web cockpit

An important matter is how to process a rule that can be just partially fulfilled. The answer is: a conclusion can only be as true as the premise. This leads us to using the minimum operator (choosing the minimum of two values).

$$A \rightarrow B: \mu_{A \rightarrow B}(x, y) := \min\{\mu_A(x), \mu_B(y)\}$$

For the usual set operators calculation rules are defined, so they can also be used on fuzzy sets.

Disjunction:	$A \cup B: \mu_{A \cup B}(x)$	$:= \max\{\mu_A(x), \mu_B(x)\}$
Conjunction:	$A \cap B: \mu_{A \cap B}(x)$	$:= \min\{\mu_A(x), \mu_B(x)\}$
Complement:	$A^c: (x)$	$:= 1 - \mu_A(x)$

So now rules can be negated (complement) and have multiple premises, that are connected by a disjunction or conjunction. The conclusion of the result evaluations can be conflicting, which can be solved by aggregation operators such as maximum or sum.

4.3 Defuzzification

The principle of the defuzzification is to change vague terms back into sharp values. There are multiple ways to implement this but, due to its simplicity, we used the “Center-of-Gravity”-Method. Therefore the solution set has to be seen as an area or graph, of which the center of gravity is built. To calculate the abscissa value of the center of gravity, the following formula is used:

$$u_S = \frac{\int_{u_{min}}^{u_{max}} u * \mu_B(u) du}{\int_{u_{min}}^{u_{max}} \mu_B(u) du}$$

- u_s abscissa value of center of gravity
- u_{min} start of membership function
- u_{max} end of membership function
- B result fuzzy-set of inference
- μ_B membership function of fuzzy-set B.

4.4 Usage in This Project

In our project, the purpose of the fuzzy-controller is to translate many simultaneous body movements into the appropriate control instructions for the drone. A great benefit is that the movement transformation can be translated into understandable rules.

The chosen input terms need to contain the relevant movement information of the person standing in front of the Kinect. For example the arm movement: “pointing downwards”, “holding horizontally” and “pointing upwards” are assigned to the same term with the values $-1, 0$ and 1 . This is done the same way for all, like terms as “rotation” and “tilting”. See Fig. 5 for the GUI.

In this project all input variables were connected by the AND-operator, for which the Minimum-operator is used. The controller’s behavior is heavily influenced by the defined rules. In Fig. 6 you can see that all four linguistic variables have been assigned 5 terms each to define the output. In Fig. 7 we see, as an example for all inputs, the membership functions for the input “backward”. E.g. for the linguistic term “strong forward” (marked in red), which reaches a value of 1 for input values from -1 to -0.8 . Its membership degree decreases until it reaches 0 at an input value of -0.4 .

backwardSpeed	sidewardSpeed	upSpeed	rotationSpeed
strongForward	strongLeft	strongDown	strongLeft
mediumForwad	mediumLeft	mediumDown	mediumLeft
zero	zero	zero	zero
mediumBackward	mediumRight	mediumUp	mediumRight
strongBackward	strongRight	strongUp	strongRight

Fig. 6 Linguistic terms of the output variables

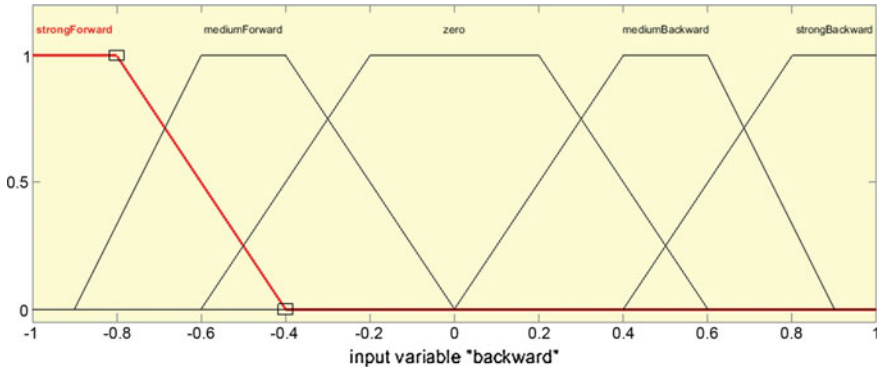


Fig. 7 Membership function for *strong forward* at input *backward*

The membership functions were chosen in a way they promote the desired flight behavior. For example “mediumForward” and “mediumBackward” both almost reach the center of “zero”, causing the same direct behavior of the drone for bending forward as well as backwards, just in the opposite direction. Figures 8 and 9 show the view of the “bird” including a virtual horizon and the system in action respectively.

Fig. 8 Head up display

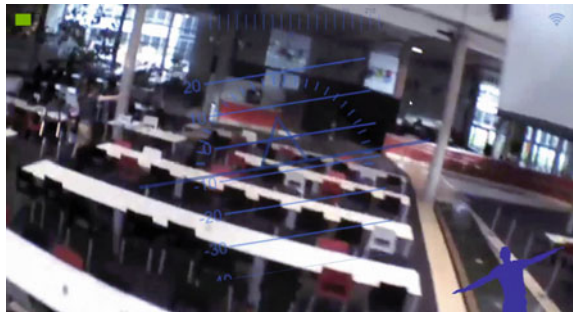


Fig. 9 He believes he can fly



5 Conclusions

As mentioned in the beginning, this project's main purpose was simply to have fun with a quadrotor. It was a playful idea that involved a serious technical challenge. The result of this project is something that realized this idea. Nonetheless, there are still some things that need to be done on it.

One idea for carrying on this project, would be to optimize the flight behavior of the drone. Since it is mostly used indoors (no weather influences), it would be nice to make it avoid collisions with, e.g. doors or walls and to have an overall stable flight. This part was picked up by another team of two people and will be tested and developed soon.

References

1. Akyol, S., Canzler, U., Bengler, K., Hahn, W.: Gesture control for use in automobiles. In: MVA, pp. 349–352 (2000)
2. Kela, J., Korpipää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L., Marca, D.: Accelerometer-based gesture control for a design environment. *Pers. Ubiquit. Comput.* **10** (5), 285–299 (2006)
3. Trail, S., Dean, M., Odowichuk, G., Tavares, T.F., Driessen, P.F., Schloss, W.A., Tzanetakis, G.: Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the Kinect. In: NIME (2012)
4. Song, W., Guo, X., Jiang, F., Yang, S., Jiang, G., Shi, Y.: Teleoperation humanoid robot control system based on kinect sensor. In: 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), vol. 2, pp. 264–267. IEEE (2012)
5. Wang, Y., Yang, C., Wu, X., Xu, S., Li, H.: Kinect based dynamic hand gesture recognition algorithm research. In: 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), vol. 1, pp. 274–279. IEEE (2012)
6. Zadeh, L.A.: *Fuzzy Sets and Systems* (1965)

ROS-Gazebo Supported Platform for Tag-in-Loop Indoor Localization of Quadcopter

Shuyuan Wang and Tianjiang Hu

Abstract Localization and navigation inside GPS-denied buildings has been one of the main technological challenges of quadcopter researches. Hereafter, this paper proposes and develops a supporting research platform integrated with 2D tag visual fiducials for quadcopter indoor autonomous localization. Robot operating system (ROS) and Gazebo are simultaneously fused into the integrated platform. Under such circumstances, tag-involved images are sequentially captured via on-board cameras, while vehicle position/posture is achieved via off-board processing based on the open-source AprilTag algorithm. Simulation and experiments of the AR.Drone 2.0 are conducted to demonstrate the system architecture and workflow of the developed tag-in-loop indoor localization research platform. The results validate the effectiveness and application potentials of the ROS-Gazebo platform to support quadcopters' autonomous indoor localization, flight autopilot, and cooperative control, etc.

Keywords Robot operating system (ROS) · Gazebo · Indoor localization · Tag-in-loop · Quadcopter · AprilTag

1 Introduction

Unmanned aerial vehicles (UAVs) have attracted more popularity due to their good performance and great potentials in military operations and civilian tasks. Quadcopters have been definitely one of popular UAV representatives in the past few years, thanks to the amazing TED shows by Vijay Kumar (http://www.ted.com/speakers/vijay_kumar) [1, 2] and Raffaello D'Andrea (http://www.ted.com/speakers/raffaello_d_andrea) [3].

S. Wang · T. Hu (✉)
College of Mechatronics and Automation,
National University of Defense Technology,
410073 Changsha, China
e-mail: t.j.hu@nudt.edu.cn

In terms of popularized application requirements, autonomous localization without global positioning satellite supporting environments, e.g. inside buildings, tunnels, and downtowns, is still one key field of quadcopter researches. Since indoor situations are usually unknown and GPS-denied, the flying quadcopters load on-board sensors for self-positioning within such indoor environments. In the past works, several kinds of sensors have been on-board explored and tested, such as laser scanners [4], monocular cameras [5–7], stereo cameras [8] and RGB-D sensors [9]. From the point of view of applications, monocular cameras can not only provide abundant information but also has lower weight, size and power consumption.

In the past few years, multi-sensor fusion algorithms for UAV indoor localization have emerged in large numbers [10–13]. It is still a key open problem for algorithm parameter tuning and performance evaluation. Eventually, this paper concentrates on constructing a general ground-truth research platform for UAV indoor localization. Robot operating system (ROS) and Gazebo are considered and selected as the basic framework due to their generality and extensive concerns. Furthermore, 2D code tag [14] is used as visual artificial landmarks to facilitate ground-truth positioning of quadrotor-like UAVs.

Supported by the developed ROS-Gazebo environment, scientists and researchers can smoothly design or evaluate indoor localization algorithms, such as AprilTag [15], ARToolkit [16], and Bokode [17]. In this study, the parrot AR. Drone 2.0 is adopted as the flying vehicle within the simulation and experimental scenarios [18]. Furthermore, parameter tuning and performance evaluation of the open-source AprilTag [15] is demonstrated as an illustrative example of the developed platform.

The remainder of this article is organized as follows. A ROS-Gazebo supported system is proposed and developed for indoor localization researches in Sect. 2. System architecture and workflow are both presented as well. In Sect. 3, a typical application is demonstrated by using 2D Tag as the visual fiducials. General principle and workflow of tag-in-loop positioning algorithms are depicted respectively. Meanwhile, the algorithm is also implemented in the ROS environment [19]. Simulation and experiments of an AR.Drone 2.0 quadcopter are conducted to demonstrate the effectiveness and feasibility of the developed ROS-Gazebo supported system in Sect. 4. Finally, Sect. 5 makes concluding remarks and future work.

2 ROS-Gazebo Supported System Development

2.1 System Architecture

In this study, a ROS-Gazebo supported system is conducted to support concept demonstration, algorithm tuning and performance evaluation of quadcopter

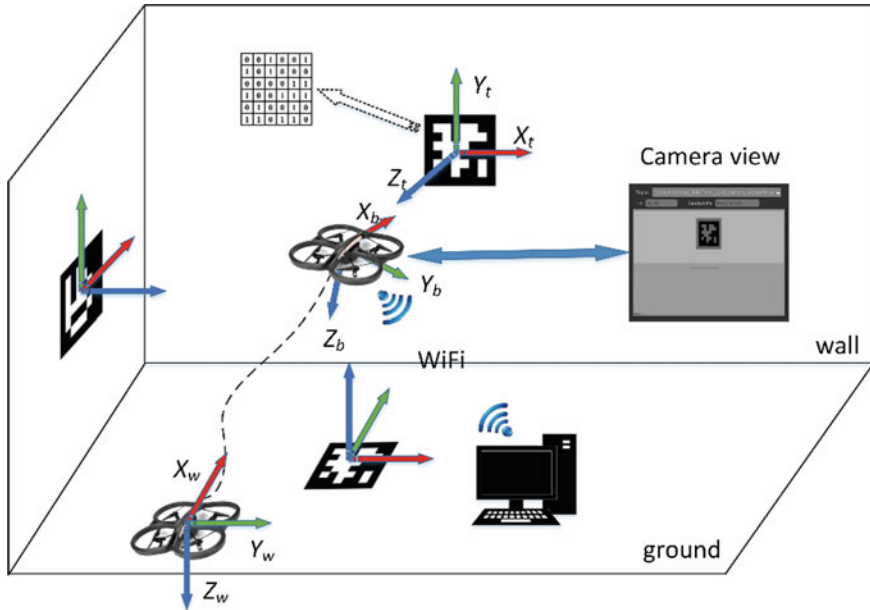


Fig. 1 Schematic diagram of a ROS-Gazebo supported system

indoor localization. The system is composed of AR.Drone 2.0 quadcopter, visual fiducials (2D coded tag) and ground-based laptop, as shown in Fig. 1.

Coordinates

Several coordinates are involved in the visual landmarks aided indoor localization algorithms. These coordinate systems include the camera image plane coordinate $S_c-O_cx_cy_cz_c$, the tag coordinate $S_t-O_tx_t y_t z_t$, the body coordinate $S_b-O_bx_b y_b z_b$ and the world coordinate $S_w-O_wx_w y_w z_w$. As shown in Fig. 1, the origin O_w of the world coordinate is set as $[x_{t1} \ y_{t1} \ z_{t1}]^T$ in the tag coordinate.

The world coordinate is kept stable, while the body coordinate is evolving with the flying vehicle. For convenience, the world coordinate system S_w is regarded to collapse with the body coordinate system S_b at the initial moment t_0 , as follows,

$$S_w = S_b(t_0) \tag{1}$$

The position/posture achieved by the landmark-aided localization algorithm is usually compared with ground truths to facilitate parameter tuning and evaluation. Such comparisons ought to be in the same coordinate. Therefore, the transformation among the defined coordinates is necessarily considered and designed. The transformation matrix from S_i to S_j is defined as:

$$C_{S_i}^{S_j} = \begin{bmatrix} R_{S_i}^{S_j} & T_{S_i}^{S_j} \\ 0 & 1 \end{bmatrix} \quad (2)$$

where $R_{S_i}^{S_j}$ denotes the rotating transform matrix and is expressed as

$$R_{S_i}^{S_j} = \begin{bmatrix} n_{ix}^j & o_{ix}^j & a_{ix}^j \\ n_{iy}^j & o_{iy}^j & a_{iy}^j \\ n_{iz}^j & o_{iz}^j & a_{iz}^j \end{bmatrix} \quad (3)$$

in which, the first column of $\begin{bmatrix} n_{ix}^j & n_{iy}^j & n_{iz}^j \end{bmatrix}^T$ denotes three components of the x_i axis within the S_j coordinate, the second column of $\begin{bmatrix} o_{ix}^j & o_{iy}^j & o_{iz}^j \end{bmatrix}^T$ is three components of the y_i axis related to S_j coordinate system, and the last column of $\begin{bmatrix} a_{ix}^j & a_{iy}^j & a_{iz}^j \end{bmatrix}^T$ means three components of the z_i axis within the S_j coordinate system.

Furthermore, $T_{S_i}^{S_j}$ in (2) denotes the translation vector and is given as

$$T_{S_i}^{S_j} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (4)$$

In this study, the tag coordinate S_t is selected as the uniform coordinate for comparisons, so the transformation matrices $S_w \rightarrow S_t$, $S_t \rightarrow S_c$ are required and presented as:

$$C_{S_w}^{S_t} = \begin{bmatrix} 0 & 1 & 0 & x_{t1} \\ 0 & 0 & -1 & y_{t1} \\ -1 & 0 & 0 & z_{t1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$C_{S_t}^{S_c} = \begin{bmatrix} n_{ix}^c & o_{ix}^c & a_{ix}^c & x_{c1} \\ n_{iy}^c & o_{iy}^c & a_{iy}^c & y_{c1} \\ n_{iz}^c & o_{iz}^c & a_{iz}^c & z_{c1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

In the ROS-Gazebo supported system, the ground truths give the quadcopter position/posture in S_w . Since the AprilTag algorithm is used as an illustrative example of autonomous localization inside buildings, the algorithm actually outputs the position/posture of the 2D tag related to the on-board camera, namely within the camera coordinate S_c . Uniformly transforming the quadcopter position/posture into S_t makes performance comparisons feasible. The position of quadcopter in S_w is $[x_w \ y_w \ z_w]^T$, the position of quadcopter in S_t is $[x_t \ y_t \ z_t]_{gro}^T$.

$[x_t \ y_t \ z_t]_{gro}^T$ can be got from $[x_w \ y_w \ z_w]^T$ by using the transfer matrix. The coordinate transformation formula is as follows:

$$\begin{bmatrix} x_t \\ y_t \\ z_t \\ 1 \end{bmatrix}_{gro} = C_{S_w}^{S_t} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{7}$$

Assume P_i as one point on the 2D tag. The coordinate of P_i in S_t is $[x_{ti} \ y_{ti} \ z_{ti}]^T$, which is determined by measuring the tag physical size. On the other side, the P_i coordinate in S_c is set as $[x_{ci} \ y_{ci} \ z_{ci}]^T$, which can be obtained from the camera intrinsic matrix. The two coordinates satisfy the following relation:

$$\begin{bmatrix} x_{ci} \\ y_{ci} \\ z_{ci} \\ 1 \end{bmatrix} = C_{S_t}^{S_c} \begin{bmatrix} x_{ti} \\ y_{ti} \\ z_{ti} \\ 1 \end{bmatrix} \tag{8}$$

In practical applications, P_i is generally taken as the corners of the 2D tag. All the four corners comply with (8) to establish a classic PnP problem [20]. Hence, the transformation matrix $C_{S_t}^{S_c}$ can be obtained by solving the constructed the P4P problem concerning the four corners of the 2D tag.

The position/posture of the quadcopter can be achieved from the transfer matrix. The quadcopter position in S_t estimated by the AprilTag algorithm is defined as $[x_t \ y_t \ z_t]_{Apr}^T$ and expressed as

$$\begin{bmatrix} x_t \\ y_t \\ z_t \\ 1 \end{bmatrix}_{Apr} = \left[C_{S_t}^{S_c} \right]^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{9}$$

So far, through defining coordinate systems and formula transforming, the two kinds of the flying quadcopter spatial position, $[x_t \ y_t \ z_t]_{gro}^T$ and $[x_t \ y_t \ z_t]_{Apr}^T$, are obtained respectively, but both within the same tag coordinate system. The former is from ground truth, and the latter is from the AprilTag algorithm.

2D Tag Visual Fiducials

2D tags are typically visual cooperative fiducials to facilitate self-positioning of flying vehicles. In the integrated system, the 2D tags are regularly printed on A4 papers. Generally speaking, Tag is a square region with black and white borders for easy detection. The small black and white blocks on the tag are encoded as 1 or 0 respectively. Every tag is generated by a coding system with a unique id. All the

tags are recorded as encoded multi-bit binary numbers—codewords—in the database. For instance, when the 36h11 tag is used in the experiments, the corresponding tag in the database is identified by 001001/101000/000011/100111/010010/110110, as shown in Fig. 1. There are 36 bits with a minimum Hamming distance of 11 in 36h11 tag. That means there are at least 11 bits different between any two codewords. Tags are classified into several patterns, such as 36h11 (6×6), 25h9 (5×5) and 16h5 (4×4). The number of bits and the minimum Hamming distance between any two codewords make differences between different tag groups.

Software Configuration

Actually, the ROS-Gazebo system has two existence forms, in terms of physical system and simulation system. Both of physical and simulation subsystems correspond to each other in every aspect. The simulation system reproduces experimental scenarios, including AR.Drone quadcopter and 2D tag, in Gazebo by applying existing models and building new models. Moreover, Gazebo can provide the direct ground truth of quadcopter flights, while the physical ground truth is usually provided by IMU without vicon-like absolute positioning systems.

Since the localization algorithm is implemented as a ROS package, the software configuration mostly concentrates on the ROS running environment, as shown in Fig. 2. *Roscore* runs as a node master to build a network for communication

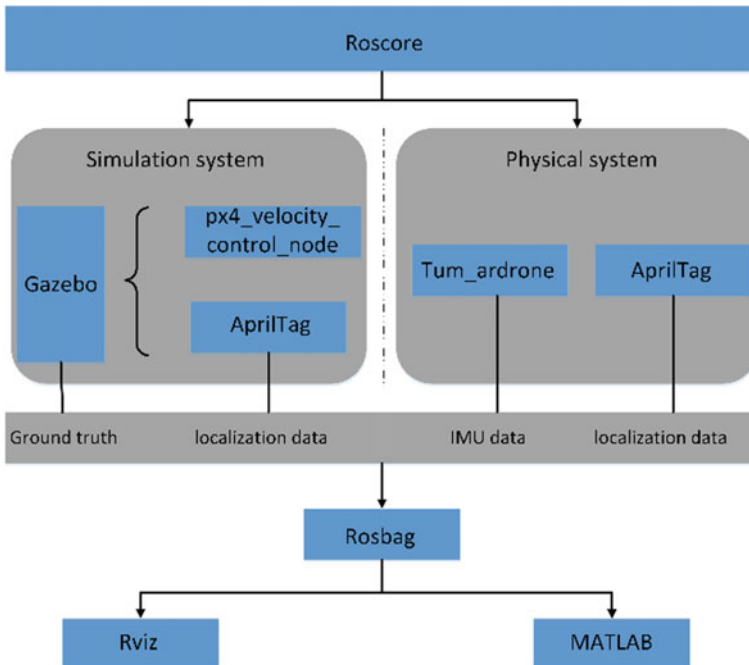


Fig. 2 Illustration of the system software architecture

between all nodes. In the physical system, packages required to complete special features are *tum_ardrone* (http://wiki.ros.org/tum_ardrone) and *apriltag* (http://wiki.ros.org/apriltags_ros). Principles and functions of the two packages will be described in Sect. 3. In the simulation system, packages rely on Gazebo to complete tasks. These packages are *px4_velocity_control_node* (<https://github.com/PX4>) and *apriltag*.

Gazebo can simulate robots, sensors and objects in three-dimensional environments. Furthermore, physical responses for sensors and feedback for objects are realized in Gazebo. Besides many kinds of existing robot models in Gazebo, such as husky, quadcopter and turtlebot, new models are conveniently designed via *urdf/xarco* files. In the running environments, ROS publishes the *tf* transform tree responsible for coordinate transformations by using the *robot_state_publisher*. The relative positions and orientations of the robot or any components of robot to each other and to the world can be easily known by the *tf* transform tree.

Rosbag is also a package that can record experimental data, including ground truth, tag-involved images, vehicle position/posture computed by algorithms, etc. *Rviz* and MATLAB are visualization tools to show 3D trajectories and demonstrate the comparison results.

2.2 System Workflow

The ROS-Gazebo supported system aims to support algorithms demonstration, research and testing. The system workflow is generally summarized as follows. The focused algorithms are understood at first. Implementation of the algorithms is conducted within the ROS platform, such as programming, compiling and debugging. Experimental data are recorded by *rosvbag*. These recorded data then are analyzed in ground based computer, to support future researches.

In this study, the Parrot AR.Drone 2.0 quadcopter is employed as the flying platform, while AprilTag as the testing indoor localization algorithm. Tag-in-loop scenarios are respectively implemented in physical and simulation situations. The position/posture of the visual landmark 2D tag is known and preliminarily set in advance. As shown in Fig. 3, onboard camera captures sequential images containing the 2D tag. AprilTag compute online the S_t position/posture of quadcopter. Moreover, ground truth provides the S_w position/posture of quadcopter. The S_w coordinates are transformed into corresponding S_t by using (7). Comparative analyses are made between the data from ground truth and AprilTag, to evaluate the localization performance. The effectiveness of the developed platform is further demonstrated as well.

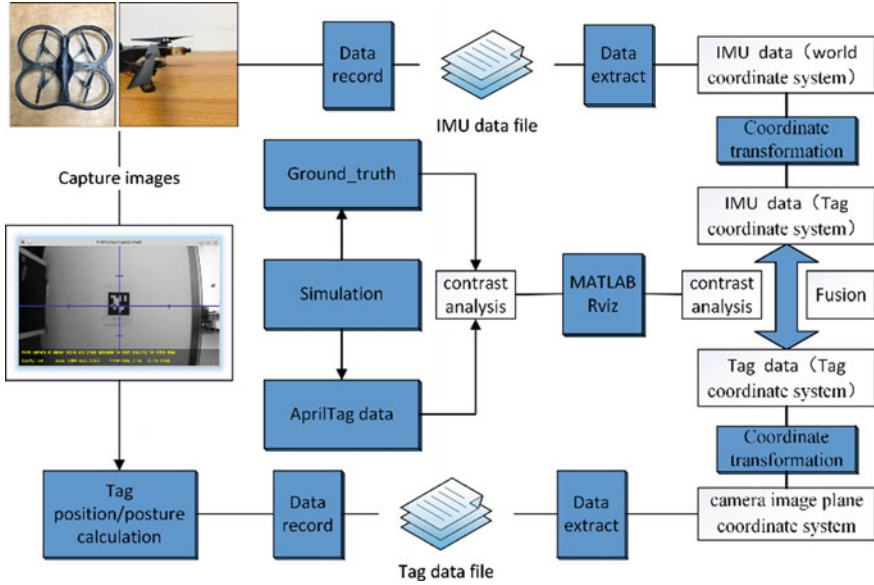


Fig. 3 System workflow of the developed ROS-Gazebo supported system

3 Tag-in-Loop Localization Algorithm and Implementation

3.1 Principle of AprilTag Algorithm

AprilTag is an open-source algorithm for indoor self-positioning by using cooperative visual landmark, 2D coded tags. Hereafter, AprilTag is a typical tag-in-loop localization algorithm.

AprilTag generally consists of four steps. Firstly, onboard camera captures a tag-involved color image, and transforms it to a grayscale image. Secondly, line segments of the tag are detected so that quads are found, through computing the gradient at every pixel and clustering pixels. The black or white block is encoded to 1 or 0 respectively. The multi-bit binary number is the codeword of this tag. Thirdly, the id of the corresponding standard tag is identified from the database by matching the extracted 0-1 sequences to codewords stored in the database. Finally, the positions of the tag region vertices in S_t is known, equations containing transformation matrix between S_c and S_t are established according to the P4P problem. R and T matrices can be solved from corner correspondence between the current tag and the standard tag. Eventually, the position/posture of the quadcopter can be achieved from transformation matrix R and T matrices.

AprilTag algorithm output is the tag's position and orientation in camera image plane coordinate system.

3.2 ROS-Based Implementation

The system architecture is based on Kubuntu 14.04 and ROS Indigo [15] infrastructure, including AprilTag algorithm implementation in ROS, establishment of indoor flight environment containing 2D tags in Gazebo, data record with rosbag, and quadcopter path calculation in *rviz*.

As described in Sect. 2, ROS published packages of *tum_ardrone*, *px4_velocity_control_node* and *apriltag* are involved in the integrated system. Their principles and functions are presented as follows. *Tum_ardrone* package drives and controls the quadcopter directly. The package consists of the driver of AR.Drone 1.0 & 2.0 *ardrone_autonomy*, keyboard control and communication protocol between quadcopter and ground based computer. When the *tum_ardrone* package runs on the computer, friendly interaction serves for operators to control the flying vehicles, while the flight trajectory and captured tag-involved images are displayed on the interface as well. The *tum_ardrone* GUI interface facilitates users to know the states and control the behaviors of the quadcopter. In the simulation system, *px4_velocity_control_node* takes place of *tum_ardrone* for interactions on the quadcopters. Actually, operators smoothly control the quadcopter via keyboard inputs when running this package.

The *apriltag* package computes the tag's position and orientation in the $S_c-O_{cx}y_{cz}z_c$ coordinate (see Fig. 1). The *apriltag* subscribes sequential images via the ROS topic from the on-board cameras. The subscription topic is */ardrone/front/image_raw* in the physical system, or */ardrone/front_cam/camera/image* in the simulation system. In addition, onboard camera calibration is also necessary for this tag-based localization algorithm. Calibration achieves the deformation parameters to improve the positioning accuracy.

Data recording and visualization serves for performance comparison and evaluation of different algorithms or parameters. In the developed system, data recording employs *rosbag* to subscribe the ROS topics of */tag_detections* and

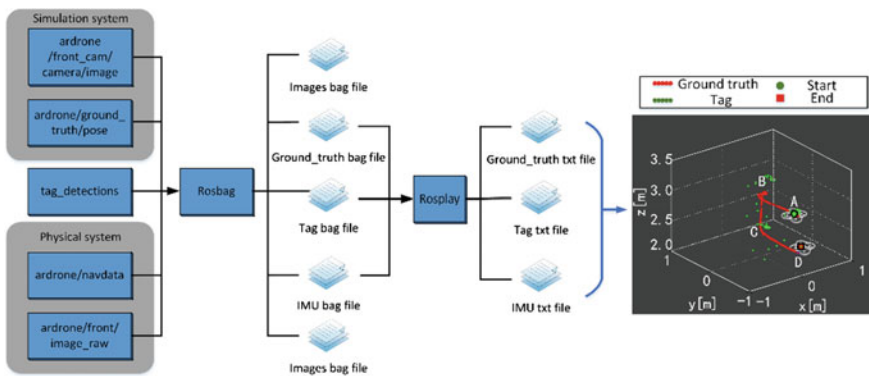


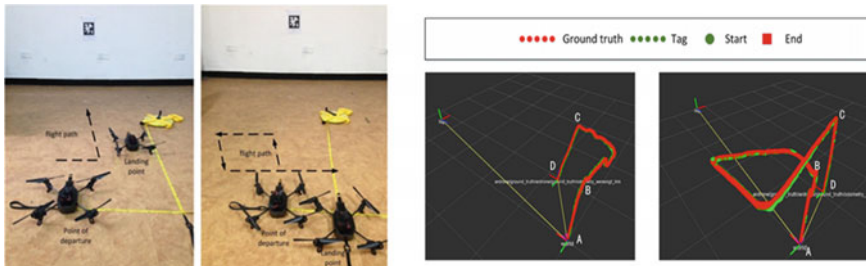
Fig. 4 Data extraction and recording process

/ardrone/navdata. Under such circumstances, both position and posture data of the flying vehicles are simultaneously recorded into bag files. Furthermore, the ROS tool *rxbag* contributes to replay the bag files and reproduce state evolving procedures. As shown in Fig. 4, data visualization and comparisons are off-board implemented by *rviz* or MATLAB on the ground-based computer. Eventually, localization accuracy is evaluated intuitively.

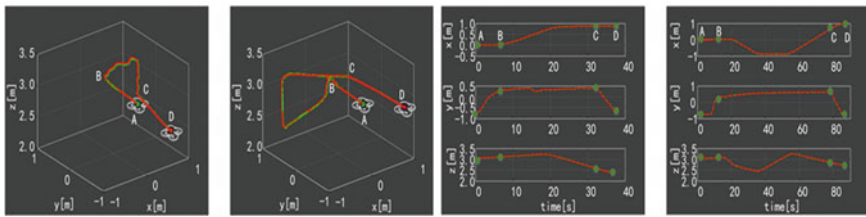
4 System Demonstration

Simulation and experiments of the AR.Drone 2.0 are conducted to demonstrate the system architecture and workflow of the developed tag-in-loop indoor localization research platform. In this study, two flight patterns are considered, in terms of straight-line flight and free flight, as shown in Fig. 5. The straight-line flight denotes taking off and flying in the desired direction for some time and landing. The free flight means that the quadrocopter is manually controlled via the *tum_ardrone* GUI.

In the simulation, quadrocopter flying trajectories are plotted real-timely via the *rviz* tool. As shown in Fig. 5, the red line denotes ground truth provided by Gazebo,



(a) Two flight patterns and Estimated against ground truth trajectories shown in *rviz* environments.



(b) Path in tag coordinate system and Components of the two paths

Fig. 5 Simulation: **a** Estimated trajectory by *apriltag* (green) and ground truth trajectories (red) are plotted real-timely via the *rviz*. **b** Quadrocopter trajectories are unified into the S_i by (7, 9). The visualized three-dimensional comparisons of the two trajectories are plotted respectively, and localization errors are not greater than 5 cm

and the green line represents estimated trajectory by *apriltag*. The point A is the departure point, and the point D is the landing point. A → B is the takeoff stage. B → C is the flying stage. And C → D is the landing stage. Quadcopter trajectories are unified into the S_t coordinate system by (7, 9). The visualized three-dimensional comparisons are shown in Fig. 5a and b, respectively via *rviz* and Matlab.

Within the two flight patterns, the AprilTag algorithm makes the estimated trajectory converge to the ground truth, if there is always a 2D tag within the on-board camera’s field view. In the ROS-Gazebo situation, localization errors are not greater than 5 cm. The on-board mono camera definitely enables quadcopter self-positioning within the simulation environments. It is demonstrated that the ROS-Gazebo integrated system effectively supports the AprilTag algorithm demonstration, research and testing. It is further validated the effectiveness and application potentials of the ROS-Gazebo environment to support quadcopter researches.

Correspondingly, the two flight patterns are experimentally conducted in the developed system as well. The ground truth of the quadcopter flights are set as the on-board IMU S_w coordinates that are converted to S_t by (7). The corresponding results are shown in Fig. 6. Different from the simulations, experiments have lower data sampling frequency, usually 8 Hz that is less than about 30 Hz of the simulations. Therefore, the green trajectories estimated by AprilTag seems sparse to some extent.

Comparisons between the IMU and AprilTag results show that the tag-in-loop localization has relatively lower accuracy, as shown in Fig. 6. Further researches are to be conducted to unveil the inaccuracy in the experiments and the mismatching performance against the simulations. On one hand, experiments take IMU data as the ground truth that is not as stable as the Gazebo references. The inaccuracy caused by the IMU references is remarkable with considerations on indoor short-term navigation. On the other hand, the tag-involved images are captured when the quadcopter is flying with disturbance. The disturbance will cause input images blurred and distortion, especially with a low resolution camera sensor. The

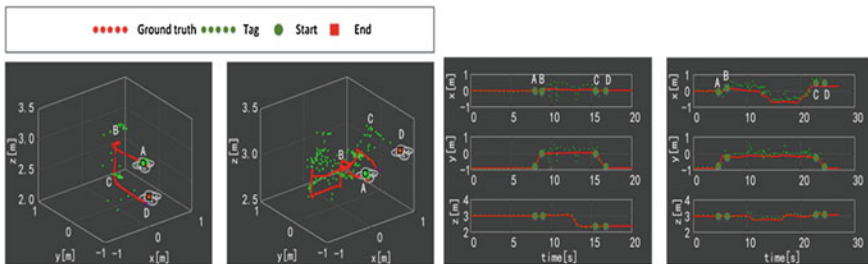


Fig. 6 Real flight: Quadcopter trajectories are unified into the S_t by (7, 9). The visualized three-dimensional comparisons of the two trajectories are plotted respectively. Localization has the same trend, while localization accuracy is lower than simulation

inputted images of *AprilTag* cannot correctly reflect the vehicle state at the current time. That is to say, the inaccurate tag-involved image shall cause localization inaccuracy.

Eventually, as the simulation and experiments demonstrate, the ROS-Gazebo integrated system supports the quadcopter flights to test and evaluate performance of the algorithms. In the following researches, the system has potentials to contribute to workflow improvement and parameter tuning of the indoor localization algorithms.

5 Concluding Remarks

In this paper, the multi-function research platform is conducted for quadcopter indoor self-positioning design and development. The platform adopts ROS and Gazebo to consolidate the system framework. Such an open-source supporting environment has potentials to enhance generality and attract contributions from interested societies. The AR.Drone 2.0 quadcopter demonstrates the system workflow by using *AprilTag* to localize itself inside the laboratory. Under Gazebo scenarios, the 2D tag based position/posture data is automatically generated and compared with the Gazebo ground truth. Under experimental situations, the flying vehicle position/posture is achieved by the on-board imaging and off-board processing. The positioning results can compare to the artificial calibration references as well. The illustrative example of quadcopter indoor flights has validated the effectiveness of the developed ROS-Gazebo supported platform.

In the near future, the platform definitely needs a 3D motion capture system, for instances, vicon or optiTrack, for setting up experimental ground truth. On the other side, the platform has potentials with more applications of indoor localization algorithm design, parameter tuning, performance evaluation, and concept demonstration, etc.

References

1. Das, A.K., et al.: A vision-based formation control framework. *IEEE Trans. Robot. Autom.* **18**(5), 813–825 (2002)
2. Mellinger, D., Michael, N., Kumar, V.: Trajectory generation and control for precise aggressive maneuvers with quadrotors. *Int. J. Robot. Res.* 0278364911434236 (2012)
3. Ritz, R., et al.: Cooperative quadcopter ball throwing and catching. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2012)
4. Grzonka, S., Grisetti, G., Burgard, W.: Towards a navigation system for autonomous indoor flying. In: ICRA'09. IEEE International Conference on Robotics and Automation, 2009. IEEE (2009)
5. Blösch, M., et al.: Vision based MAV navigation in unknown and unstructured environments. In: 2010 IEEE International Conference on Robotics and Automation (ICRA). IEEE (2010)

6. Achtelik, M., et al.: Onboard IMU and monocular vision based control for MAVs in unknown in-and outdoor environments. In: 2011 IEEE International Conference on Robotics and Automation (ICRA). IEEE (2011)
7. Bylow, E., et al.: Real-time camera tracking and 3D reconstruction using signed distance functions. Robotics: In: Science and Systems (RSS) Conference 2013, vol. 9 (2013)
8. Achtelik, M., et al.: Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments. In: SPIE Defense, Security, and Sensing. International Society for Optics and Photonics (2009)
9. Huang, A.S., et al.: Visual odometry and mapping for autonomous flight using an RGB-D camera. In: International Symposium on Robotics Research (ISRR) (2011)
10. Mellinger, D., Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: 2011 IEEE International Conference on Robotics and Automation (ICRA). IEEE (2011)
11. Müller, M., Lupashin, S., D'Andrea, R.: Quadrocopter ball juggling. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2011)
12. Eberli, D., et al.: Vision based position control for MAVs using one single circular landmark. *J. Intell. Robot. Syst.* **61**(1–4), 495–512 (2011)
13. Engel, J., Sturm, J., Cremers, D.: Scale-aware navigation of a low-cost quadcopter with a monocular camera. *Robot. Auton. Syst.* **62**(11), 1646–1656 (2014)
14. Chu, C.-H., Yang, D.-N., Chen, M.-S.: Image stabilization for 2D barcode in handheld devices. In: Proceedings of the 15th International Conference on Multimedia. ACM (2007)
15. Olson, E.: AprilTag: a robust and flexible visual fiducial system. In: 2011 IEEE International Conference on Robotics and Automation (ICRA). IEEE (2011)
16. Wagner, D., et al.: Pose tracking from natural features on mobile phones. In: Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society (2008)
17. Mohan, A., et al.: Bokode: imperceptible visual tags for camera based interaction from a distance. *ACM Trans. Graph. (TOG)* **28**(3), 98 (2009)
18. Dijkshoorn, N.: Simultaneous localization and mapping with the ar. drone. Ph.D. dissertation, Masters thesis, Universiteit van Amsterdam (2012)
19. Martinez, A., Fernández, E.: Learning ROS for Robotics Programming. Packt Publishing Ltd. (2013)
20. Horaud, R., et al.: An analytic solution for the perspective 4-point problem. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1989. Proceedings CVPR'89. IEEE (1989)

The Project PRISMA: Post-Disaster Assessment with UAVs

Carmine Tommaso Recchiuto and Antonio Sgorbissa

Abstract In the context of emergency scenarios, Unmanned Aerial Vehicles (UAVs) are extremely important instruments, in particular during monitoring tasks and in relation to the Post-Disaster assessment phase. The current paper describes a summary of the work performed during PRISMA [1], a project focused on the development and deployment of robots and autonomous systems able to operate in emergency scenarios, with a specific reference to monitoring and real-time intervention. Among other aspects, the investigation of strategies for mapping and for path following, for the implementation of Human-Swarm Interfaces and for the coverage of large areas have been performed, and they will be here summarized.

Keywords UAVs · Multirotors · Post-Disaster Assessment · Image Stitching · Path following · Coverage algorithms · Human-Swarm Interfaces

1 Introduction

In recent years, the area of Unmanned Aerial Vehicles (UAVs) research has shown rapid developments, giving the possibility of deploying these typology of robots in a wide range of domains: agriculture, environmental monitoring, sport events, cinema, Search & Rescue and emergency scenarios.

The usage of UAVs in the Search & Rescue (SAR) domain has widely been investigated: indeed, small aerial vehicles give the possibility to quickly monitor large areas, with a high maneuverability while moving in environments filled with debris and obstacles. For all these reasons, many research teams have already practically tested the deployment of aerial robots in disaster scenarios: the Pratt team [2] implemented small UAVs after the Hurricane Katrina, for inspecting buildings in urban environments, performing omni-directional obstacle avoidance. In the same

C.T. Recchiuto (✉) · A. Sgorbissa
Università degli studi di Genova, DIBRIS Department,
via all'Opera Pia 13, 16145 Genoa, Italy
e-mail: carmine.recchiuto@dibris.unige.it

years, the work of Murphy [3, 4] focused on the analysis of a cooperative strategy between ground vehicles and UAVs, for seawall and bridge inspections. UAVs were also deployed in the context of the L'Aquila earthquake [5] and of the Lushan earthquake [6].

The EU funded projects ICARUS and NIFTi are also worth of mention. The main aim of the project ICARUS [7] was the improvement of crisis management with the use of unmanned SAR robotic systems embedded and integrated into existing infrastructures. The aerial platforms included also solar rotor-wing and fixed-wing UAVs, developed by ETH for sustained flight [8]. The NIFTi project [9] focused on human-robot cooperation, and, in the context of the project, a novel octocopter equipped with a camera and a Kinect-style sensor was designed and developed.

As evident from all these works, the typologies of UAVs adopted in these scenarios are different: however, it can be seen that, while in the earliest approaches the preferred choice was small fixed-wing UAVs and miniature helicopters [4, 10], multirotors have been used in more recent events, given their limited cost and the ability of vertically taking-off and landing [7, 11].

This work started from all these basis, analyzing the principal issues for SAR performed with UAVs (in particular in the Post-Disaster Assessment phase) and proposing some solutions to common problems. This paper summarizes the work performed in the project PRISMA, that focused in particular on the integration of the ETHNOS real-time framework, on the implementation of novel strategies for path following, on the analysis of the performances of a subset of coverage algorithms, and on tests with human subjects aimed at studying Human-Swarm Interfaces. The following paragraphs will describe all these aspects.

2 The Asctec Multirotors and the ETHNOS Framework

As stated in the Introduction, multirotors are nowadays a common choice in works involving UAVs, mainly because of their maneuverability. In the context of the project PRISMA, two multirotors have been used: the Asctec Firefly (esarotor) and Pelican (quadrotor) [12] (Fig. 1).

The two platform are endowed with a Flight Control Unit (FCU) *Autopilot*. The FCU features a complete Inertial Measurement Unit (IMU) as well as two 32 bits, 60 MHz ARM-7 microcontrollers used for flight control and data fusion. The two microcontrollers have different aims: the first, the Low-Level Processor (LLP) is responsible for the IMU sensor data fusion and for the attitude-based controller, while the second, the High-Level Processor (HLP) may be used to implement further custom code. For the computationally more expensive onboard processing tasks (e.g., image processing for localization) there is also an embedded computer, with a Intel Core I7 CPU, a 1 GB RAM, a MicroSD card slot for the operating system.

In the proposed implementation, the HLP was considered as a bridge for the serial communication between the LLP and the on-board CPU, where the control code

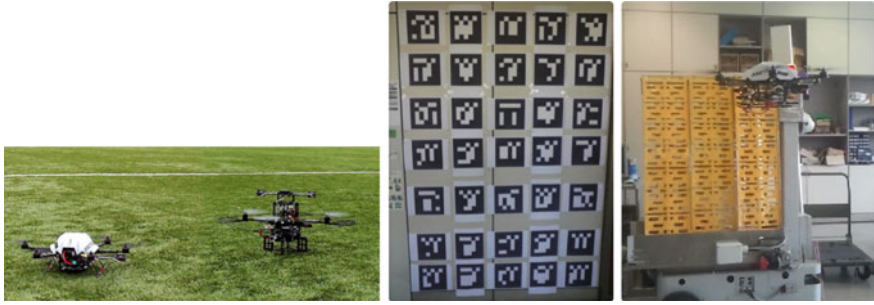


Fig. 1 The Asctec Firefly and Pelican during an experimental session (*left*). The Asctec Firefly during an indoor flight, localized with a board of markers (*right*)

was implemented (Fig. 2). For the control development and for the communication between robots and human operators, the ETHNOS framework [13] was adopted.

ETHNOS (Expert Tribe in a Hybrid Network Operating System) is a programming environment for the design of real-time control systems. From the runtime perspective, ETHNOS provides the possibility of real-time executing periodic and sporadic tasks, a feature that is crucial in aerial vehicles. Indeed, a dedicated network communication protocol, within the distributed operating system, allows a robust communication even in noisy wireless network. The framework achieves soft real-time compliance being a real-time extension to the standard Linux kernel [14]. The reference architecture of the ETHNOS operating system is based on the concept of *expert*, a concurrent agent responsible for a specific deliberative or reactive behavior. The environment handles the concurrent execution and the scheduling of the experts, and their synchronization when accessing the shared resources.

An example of the ETHNOS architecture is the experimental setup developed for indoor localization and autonomous control. The control and indoor localization of the Asctec multirotors have been performed on board using the concurrent structure based on the concept of expert. In this case, the control of the robot has been implemented using five periodic experts:

- *Acquisition*: handles the communication with the HLP (and consequently with the LLP) with a frequency of 100 Hz, periodically reading the data coming from the on-board sensor and sending to the FCU the commands related to thrust and attitude angles, calculated by the expert Control (see below).
- *Localization*: implements an algorithm for indoor localization, based on the artificial vision library ArUco [15]. The principal functionality of the library is to recognize up to 1024 different markers, applying an Adaptive Thresholding and the Otsu’s algorithm. When a marker is recognized, the relative distance and orientation of the camera with respect of the marker is calculated. For improving the accuracy of the localization, a wall of 35 markers has been created (Fig. 1, right) and a custom algorithm has been developed, based on the elimination of the outliers and the estimation of the average value. The expert operates at a frequency of 30 Hz.

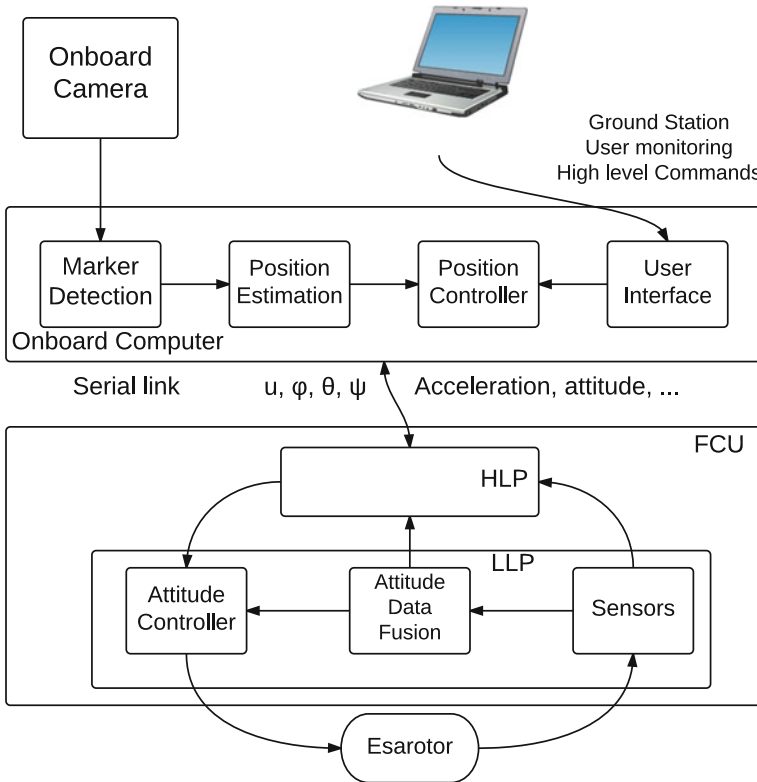


Fig. 2 Architecture of the whole system

- *Interface*: works at 5 Hz, and it offers a simple User Interface integrated in the Unix Shell that allows the user to select the operation to be performed.
- *Planner*: acts as a simple bridge between Interface and Control, in case of complex actions (i.e. waypoint too far, list of commands to be executed), with a frequency of 10 Hz.
- *Control*: sporadic expert, it uses the information deriving from the other experts (i.e. position, attitude, target position, ...) for the control of the robot. Actually, the UAV calculates the error between the target position (a fixed value when hovering, a series of waypoints in more complex cases) and uses the error in space as input of three PID controllers for the three direction in space:

$$\begin{aligned}
 u_x &= K_{Px}(x - x_T) + K_{Dx} \frac{d}{dt}(x - x_T) + K_{Ix} \int (x - x_T) \\
 u_y &= K_{Py}(y - y_T) + K_{Dy} \frac{d}{dt}(y - y_T) + K_{Iy} \int (y - y_T) \\
 u_z &= K_{Pz}(z - z_T) + K_{Dz} \frac{d}{dt}(z - z_T) + K_{Iz} \int (z - z_T)
 \end{aligned}$$

where the K coefficients are the proportional, derivative and integrative proportional terms and x_T , y_T and z_T are the references in position. Starting from the resulting target accelerations u_x , u_y and u_z , these values are used to calculate the reference thrust u and the control angles ϕ_d (pitch) and θ_d (roll), considering the dynamics of the system, the mass m of the UAV and the angle ψ_d (yaw):

$$u = m\sqrt{\mu_x^2 + \mu_y^2 + (\mu_z + g)^2}$$

$$\phi_d = \sin^{-1}\left(m\frac{\mu_x \sin \psi_d - \mu_y \cos \psi_d}{u}\right)$$

$$\theta_d = \tan^{-1}\left(\frac{\mu_x \cos \psi_d - \mu_y \sin \psi_d}{\mu_z + g}\right)$$

The control of the orientation of the multirotor (ψ_d) has been achieved with a proportional controller, directly based on the error between the reference angle and the actual one.

$$\psi_d = K_{P_\psi}(\psi - \psi_T)$$

Finally, u , ϕ_d , θ_d and ψ_d are used as inputs by a inner control loop, implemented in the LLP, that directly handles the rotor velocities.

The real-time scheduling functionalities of ETHNOS are crucial for vertex-to-vertex navigation, achieved as shown for the indoor case. However, given the widespread diffusion of the framework ROS (Robot Operating System), this robotic middleware has also been integrated for some applications; therefore, within the project, a ROS-ETHNOS interface has been developed. The interface is able to convert ROS messages into ETHNOS ones, and vice versa, e.g. converting position commands from ROS topics to ETHNOS messages. An example of this integration will be presented in Sect. 5.

3 SLAM and Image Stitching

Dealing with aerial robots in unknown scenarios, localization and map building are tasks of the utmost importance. While commonly used SLAM techniques are based on Extended Kalman Filters [16] or particle filters [17], the approach used in the project PRISMA focused on the implementation of a Image Stitching algorithm. A detailed description of this typology of algorithms can be found in [18], while in this section, we will just focus on more technical aspects related to the practical implementation.

In details, the practical implementation of the algorithm has seen the adoption of feature-based detection algorithms, such as SIFT and ORB (at the discretion of

the user) that allow reducing computational costs and incrementing execution speed [19]. Both algorithms start from a scan of the selected images, in order to create the descriptors with the related features; after this step, the features are evaluated in subsequent images, and if the number of similar features is too small and below a certain threshold, the images are discarded. In this phase there is also the estimation of the camera parameters: if the intrinsic ones can be considered as a priori known, the extrinsic ones should be evaluated in relation to the received images.

For the Image Alignment process, the algorithm has seen the implementation of Bundle Adjustment techniques (i.e., a photogrammetry technique aimed at combining multiple images related to the same scene) and Feathering Image Blending (i.e., a blending technique based on the softening and on the blurring of the edges of the features) for the final Blending phase. Moreover, the Image Alignment process allows estimating the multirotor position, considering the altitude of the robot and the rototranslation vector of an image with respect of the previous one, strictly connected with the movement of the UAV. The whole code is made available, as a ROS package, at the following link: <https://github.com/CDibris/aerialSLAM>.

The images of the two cameras, and in particular the one directed toward the ground, are streamed by means of ROS topics. The algorithm periodically samples the video stream; each time a new image is saved, the ROS nodes activates a second process, that implements the Image Stitching algorithm. Preliminary tests have been implemented, using first some video datasets available online (Fig. 3) and also directly using the on-board cameras.



Fig. 3 Image Stitching process during a rototranslation of the robot. Dataset taken from http://rpg.ifi.uzh.ch/software_datasets.html

4 Real-Time Path Generation with Obstacle Avoidance

Another fundamental aspect when speaking of SAR with UAVs is related to the ability of the robots of following predefined exploration paths, being also able to modify and update such paths in real-time, based on how they perceive the environment (i.e., recognizing and avoiding obstacles on the path). During the project, a novel algorithm for path generation and obstacle avoidance, using an uncommon technique to describe and modify the path, has been proposed and implemented on the Asctec multirotors.

The algorithm (more details can be found in [20, 21]) is based on:

- *3D Path definition*, through the intersection of two surfaces represented by implicit equations in the form $f_i(x, y, z) = 0$, $i = 1, 2$. The path is given by the solution of the system:

$$\begin{cases} f_1(x, y, z) = 0 \\ f_2(x, y, z) = 0. \end{cases}$$

- *Path modification in presence of obstacles*, i.e. when a robot perceives the presence of any obstacle, the path is deformed, by means of obstacle functions added to one of the surface functions. The system becomes:

$$\begin{cases} f_1'(x, y, z) = f_1(x, y, z) + \sum_{j=1}^N O_j(x, y, z) = 0 \\ f_2(x, y, z) = 0. \end{cases}$$

where a Gaussian function is used as obstacle function:

$$O_j(x, y, z) = A_j e^{-\frac{(x-x_j)^2 + (y-y_j)^2 + (z-z_j)^2}{\sigma^2}}$$

It should be noticed that, by computing the tuning parameters A_j and σ properly, it is possible to guarantee that the deformed path does not collide with obstacles, even in the presence of multiple ones [22].

For the special case of a 2D path, e.g. the robot tries to maintain its flying height and moves on a 2D plane, it is easy to set one of the surfaces as describing a plane at a constant height (e.g., $f_2(x, y, z) = z = H$, with H being the desired height) and choose the other surface according to the path to be followed. Without loss of generality, it can be chosen the zero-height plane as the reference plane. In this case, the function describing the path turns out to contain only 2 variables, and it can be re-written as follows (using the implicit equation of a curve on the plane):

$$f(x, y) = 0$$

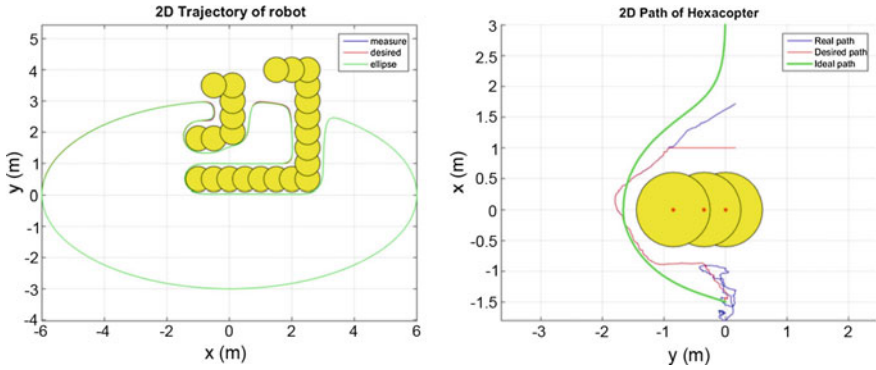


Fig. 4 Simulation of a multicopter following an ellipse with 2 nested horseshoes of obstacles (*left*). Flight of the Asctec Firefly in a straight line with 3 obstacles (*right*)

- *Path generation*, the definition of a velocity vector that should be used as input of the quadrotor, as:

$$V_k(x, y) = -k_g f(x, y) \nabla f(x, y) + \text{tangent}(x, y)$$

where k_g is the gradient vector gain, the product $-f(x, y) \nabla f(x, y)$ drives the robot towards the desired path and $\text{tangent}(x, y)$ is a vector perpendicular to $\nabla f(x, y)$ that drives the robot along the path.

An example of the simulated behaviour and of the real execution of the algorithm with the Asctec Firefly can be seen in Fig. 4.

5 Coverage Algorithms for Search & Rescue

Multicopter UAVs are an extremely interesting technology towards the end of exploring an area and gathering information before first responders come into play. During the project PRISMA, we investigated different strategies for the organization of a team of UAVs implementing autonomous coverage strategies for real-time intervention after earthquakes.

The work was mainly focused in the analysis and comparison of a subset of real-time multi-robot coverage algorithms, i.e. *Node Count* [23], *Edge Counting* [24], *Learning Real-Time A** [23], *PatrolGRAPH** [25], chosen among the most common coverage algorithms in literature, in order to classify them in term of the required coverage time and of the required energy.

For a formal description of the algorithms we refer the readers to the related literature works. In this section we will just recap the most relevant results obtained from

tests in simulation, considering as indicators the length of the longest path among all robots and the overall distance travelled by all robots:

- the *Node Count* algorithm, despite its simplicity, is the most efficient one, in particular increasing the map size and the number of robots;
- the performance of the *Edge Counting* and *PatrolGRAPH** algorithms are noticeably lower;
- however, when a higher minimum number of visits for each node is considered, *PatrolGRAPH** performs better, since it ensures a uniform frequency of visits to all vertices [26].

It is worth of mention that the algorithms were also implemented on actual robots, testing the whole framework in a real scenario. In the current implementation, based on the ROS framework (the software is made available as a ROS package at the following link: <https://github.com/meross/VRepRosQuadSwarm>), a central station is in constant communication with each robot, that has however a local copy of the navigation graph. In case of a temporary lack of global communication, different robots can continue to work, possibly with decreased performance, by using their local copies. The ROS framework and the ETHNOS architecture are integrated using the ROS/ETHNOS interface presented in Sect. 2.

6 Movement in Formation and Human-Swarm Interfaces

Even if many steps forward have been taken towards the fully autonomous control of UAVs, a human pilot is usually in charge of teleoperating the robots during SAR activities. In this context, a swarm of robots is usually deployed, and therefore the teleoperation of UAVs can become a hard task. In order to reduce the effort of the human operator, the organization of the robots in a structured formation allows to:

- increase the field of view, by positioning the robots in a way that a greater section of the surrounding environment can be visualized;
- speed up the learning process of the human operator, that is able to predict the robot motions in a more straightforward way and, consequently, to move the swarm faster;
- avoid collisions between robots;

In relation to the effort of the human operator, also the typology of visual feedback is crucial: different aspects related to the monitoring of the surrounding environment should be taken into account (e.g., the positions of the cameras, the necessity of implementing an egocentric, an exocentric view, or a hybrid combination of both, the quality of the images in relation to the available bandwidth).

In order to investigate the effects of different visual feedbacks on the human operators performances, a custom Human-Swarm Interface (HSI) has been built during the project. The HSI allows human operators to pilot a team of multirotors in environments filled with obstacles, while the robots keep a structured formation, with

a behavior-based policy where the global movement of each robot is calculated as the sum of single concurrent behaviours [27]. More details about the design of the interface can be found in [28].

The interface allows the user to control the swarm both with a camera from above the swarm (exocentric point of view) and with a combination of cameras on board the robots (egocentric point of view). While it is known that an exocentric camera allows to better estimate distances and perceive the environment [29], it is also a matter of

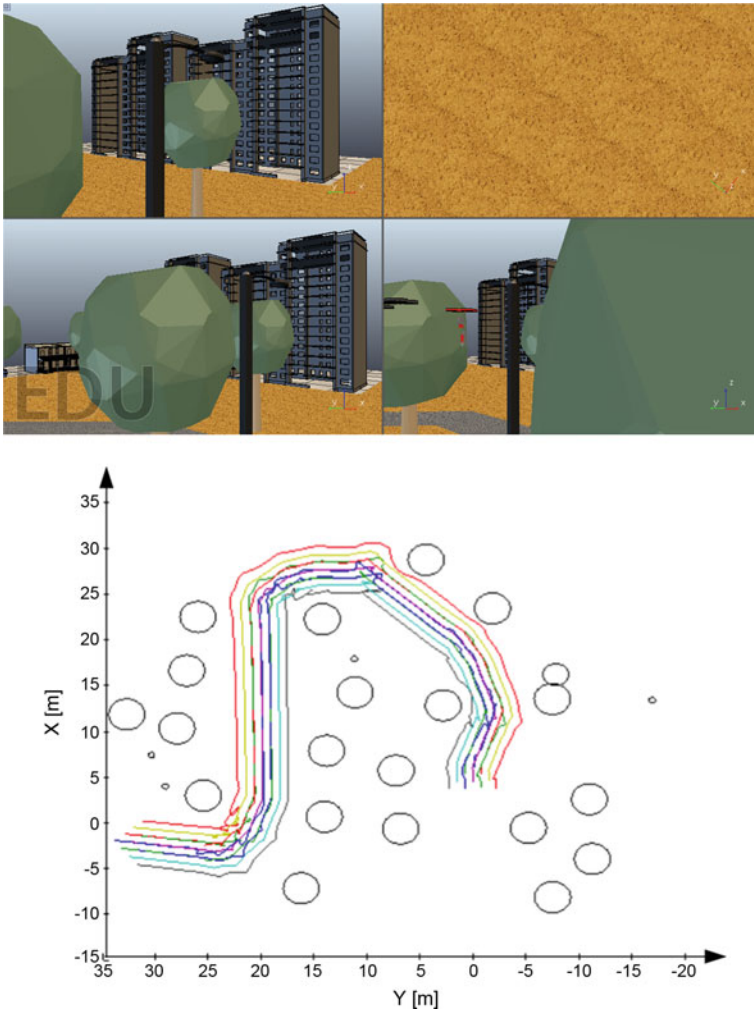


Fig. 5 Swarm in the simulated environment, with an egocentric point of view composed of two cameras on board the robot leader (pointing forward and downward) and on board the wings of the formation (the leftmost and rightmost robots) (up). Paths followed by the robots in a 10-UAVs wedge formation (down)

fact that, in real context, the common way to pilot quadrotors (when a direct line of sight is not possible) is by using the feedback of on-board cameras. Therefore, a series of experiments was aimed at investigating the degradation of performances with the camera in a *first person* configuration. For the aim of the experiments, an environment filled with obstacle was created, as in Fig. 5.

The results show that, as expected, an exocentric point of view allows to generally achieve better performances. However, it can be noticed that when the user is required to accomplish a simple task, that does not require a high situational awareness, the differences between the performances achieved with the *first person* point of view and with the camera from above were not so relevant. On the opposite, during a task consisting in manually avoiding the obstacles (Fig. 5, down), and therefore requiring a higher situational awareness, the gap in performance between the two points of view was much higher.

A more detailed discussion on the results of the experiments can be found in [28, 30]. The experimental environment and the control software is made available, as a ROS package, at the following link: <https://github.com/CDibris/QuadSwarm>.

7 Conclusions

Unmanned Aerial Vehicles and multirotors in particular are receiving in last years a tremendous attention both by the scientific community and by the market, due to their peculiar characteristics: vertical take-off and landing, limited cost, ease of construction, flexibility. For these reasons, and given also the rapid development of UAV technology, in the last decade the scientific community has devoted much interest in the implementation of UAVs swarm in the context of Search & Rescue, and in particular in the Post-Disaster Assessment phase.

This paper presents the work performed in the context of the project PRISMA, related to the development and deployment of aerial robots in emergency scenarios, with a specific reference to monitoring and real-time intervention.

The work has taken into account different aspects related to the utilization of UAVs in an emergency context: multi-robot software architectures, SLAM techniques, path following strategies, analysis of real-time coverage algorithms, development of novel Human-Swarm Interfaces.

The deployment of autonomous aerial robots in emergency situations still presents many complex challenges and open issues. The project PRISMA aims to respond to some of these challenges, by providing some practical solutions, innovative strategies and experimental analysis, using both simulation environments and actual robots.

References

1. Recchiuto, C., Sgorbissa, A., Wanderlingh, F., Zaccaria, R.: UAV Teams in Emergency Scenarios: A Summary of the Work Within the Project PRISMA
2. Pratt, K.S., Murphy, R., Stover, S., Griffin, C.: CONOPS and autonomy recommendations for VTOL small unmanned aerial system based on Hurricane Katrina operations. *J. Field Robot.* **26**(8), 636–650 (2009)
3. Murphy, R.R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., Erkmen, A.M.: Search and rescue robotics. In: *Springer Handbook of Robotics*, pp. 1151–1173. Springer, Berlin, Heidelberg (2008)
4. Murphy, R.R., Steimle, E., Griffin, C., Cullins, C., Hall, M., Pratt, K.: Cooperative use of unmanned sea surface and micro aerial vehicles at Hurricane Wilma. *J. Field Robot.* **25**(3), 164–180 (2008)
5. Quaritsch, M., Kruggl, K., Wischounig-Strucl, D., Bhattacharya, S., Shah, M., Rinner, B.: Networked UAVs as aerial sensor network for disaster management applications. *E & I Elektrotechnik und Informationstechnik* **127**(3), 56–63 (2010)
6. Qi, J., Song, D., Shang, H., Wang, N., Hua, C., Wu, C., Han, J.: Search and rescue rotary-wing UAV and its application to the Lushan Ms 7.0 Earthquake. *J. Field Robot.* (2015)
7. Govindaraj, S., Chintamani, K., Gancet, J., Letier, P., van Lierde, B., Nevatia, Y., Armbrust, C.: The ICARUS project-command, control and intelligence (C2I). In: *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1–4. IEEE (2013)
8. Leutenegger, S., Jabas, M., Siegwart, R.Y.: Solar airplane conceptual design and performance estimation. *J. Intell. Robot. Syst.* **61**(1–4), 545–561 (2011)
9. Kruijff, G.J.M., Kruijff-Korbayov, I., Keshavdas, S., Larochelle, B., Janíček, M., Colas, F., Looije, R.: Designing, developing, and deploying systems to support human-robot teams in disaster response. *Adv. Robot.* **28**(23), 1547–1570 (2014)
10. Murphy, R.R.: *Disaster Robotics*. MIT Press (2014)
11. Michael, N., Shen, S., Mohta, K., Mulgaonkar, Y., Kumar, V., Nagatani, K., Ohno, K.: Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *J. Field Robot.* **29**(5), 832–841 (2012)
12. Ascending Technologies GmbH, website. <http://www.asctec.de>
13. Piaggio, M., Sgorbissa, A., Zaccaria, R.: A programming environment for real-time control of distributed multiple robotic systems. *Adv. Robot.* **14**(1), 75–86 (2000)
14. Sgorbissa, A.: *Multi-Robot Systems and Distributed Intelligence: The ETHNOS Approach to Heterogeneity*. INTECH Open Access Publisher (2006)
15. Munoz-Salinas, R.: *ARUCO: A Minimal Library for Augmented Reality Applications Based on OpenCv*. Universidad de Crdoba (2012)
16. Smith, R.C., Cheeseman, P.: On the representation and estimation of spatial uncertainty. *Int. J. Robot. Res.* **5**(4), 56–68 (1986)
17. Pupilli, M., Calway, A.: Real-time camera tracking using a particle filter. In: *BMVC*, Sept 2005
18. Szeliski, R.: Image alignment and stitching: a tutorial. *Found. Trends Comput. Graph. Vis.* **2**(1), 1–104 (2006)
19. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: *2011 IEEE International Conference on Computer Vision (ICCV)*, Nov 2011, pp. 2564–2571. IEEE (2011)
20. Morro, A., Sgorbissa, A., Zaccaria, R.: Path following for unicycle robots with an arbitrary path curvature. *IEEE Trans. Robot.* **27**(5), 1016–1023 (2011)
21. Sgorbissa, A., Zaccaria, R.: 3D path following with no bounds on the path curvature through surface intersection. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2010, pp. 4029–4035. IEEE (2010)
22. Sgorbissa, A.: *An integrated approach to path following, obstacle avoidance, and motion planning in 2D and 3D workspaces: a case study with mobile, flying, and underwater robots*. Technical Report, DIBRIS. University of Genova, Genova, Italy (2015)

23. Korf, R.E.: Real-time heuristic search. *Artif. Intell.* **42**(2–3), 189–211 (1990)
24. Koenig, S., Simmons, R.G.: Easy and hard testbeds for real-time search algorithms. In: AAAI/IAAI, vol. 1, pp. 279–285, Aug 1996
25. Baglietto, M., Cannata, G., Capezio, F., Grosso, A., Sgorbissa, A., Zaccaria, R.: PatrolGRAPH: a distributed algorithm for multi-robot patrolling. In: IAS10-The 10th International Conference on Intelligent Autonomous Systems, Baden Baden, Germany, pp. 415–424, July 2008
26. Cannata, G., Sgorbissa, A.: A minimalist algorithm for multirobot continuous coverage. *IEEE Trans. Robot.* **27**(2), 297–312 (2011)
27. Balch, T., Arkin, R.C.: Behavior-based formation control for multirobot teams. *IEEE Trans. Robot. Autom.* **14**(6), 926–939 (1998)
28. Recchiuto, C.T., Sgorbissa, A., Zaccaria, R.: Usability evaluation with different viewpoints of a Human-Swarm interface for UAVs control in formation. In: 2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Aug 2015, pp. 523–528. IEEE (2015)
29. Chen, J.Y., Haas, E.C., Barnes, M.J.: Human performance issues and user interface design for teleoperated robots. *IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev.* **37**(6), 1231–1245 (2007)
30. Recchiuto, C.T., Sgorbissa, A., Zaccaria, R.: Visual feedback with multiple cameras in a UAVs Human-Swarm Interface. *Robot. Auton. Syst.* **80**, 43–54 (2016)

An Automatic Collision Avoidance Approach to Assist Remotely Operated Quadrotors

Bruno Giovanini, Hugo A. Oliveira and Paulo F.F. Rosa

Abstract The use of quadrotors to civilian and military missions has been increased and the challenges involved on controlling it, mainly in indoors and restricted environments, has been attracting robotics researchers. Thus, an automatic collision avoidance approach is of utmost importance in this scenario, given the difficulty of control and the risk of accidents involved in the use of these vehicles. This paper presents an approach for obstacle avoidance of a manually controlled quadrotor automatically, allowing the operator to keep focus on the overall mission. The method is based on constantly estimating its future path considering its dynamics, current status, current control and distances measured by four on-board sonar sensors. Simultaneously, the pose is estimated based on the quadrotor odometry and an occupation grid representation of the nearby environment is constructed using the sonar sensors measurements. All that information is used to determine an imminent collision and overrides the user control, if necessary, keeping its last safe position. All the solution was evaluated in a simulator, the real quadrotor's and sonars sensors were characterized to be embedded in the quadrotor through a computer-on-module and controlled over wireless network communication.

Keywords Indoor environment · Collision avoidance · Trajectory estimation · Mapping · PID control

1 Introduction

The use of quadrotors to civilian and military missions has been increased and its related challenges have been attracting robotics researchers in a variety of areas such as surveillance and rescue in remote areas.

B. Giovanini · H.A. Oliveira · P.F.F. Rosa (✉)
Instituto Militar de Engenharia, Rio de Janeiro, Brazil
e-mail: rpaulo@ime.eb.br

B. Giovanini
e-mail: bsgiovanini@gmail.com

The usual platform with four rotors has the suitable ability to perform vertical take off and landing, hovering and performing fast maneuvers in different environment conditions. However, controlling a quadrotor is not a simple task when a safe flight is desired. Even extensively trained pilots may face situations where it is hard to guarantee a safe flight due to, for instance, loss of visual contact. So, an automatic collision avoidance method to assist the operation of the platform is required.

Our solution consists in an algorithm to assist the operation of a quadrotor avoiding obstacles based on the Simultaneous Localization and Mapping approach (SLAM), with the global position being calculated continuously during the flight through its odometry data while on-board sonar sensors measure the relative distances to nearby obstacles and map the environment as an occupancy grid map [1]. Then, the user input is overridden if a possible collision is detected in a near future based on its calculated future trajectory and obstacle positions. A Proportional-Integral-Derivative controller (PID) is used to override the user input while the platform is in the imminence of a collision with an obstacle.

Specifically, our method continuously estimates the future trajectory of the vehicle given its dynamics, its current state, and the current operators control input. On the other hand, the environment is kept mapped using the on-board sonar sensors. The method consists in testing whether a collision between the quadrotor and an obstacle might occur. This is basically done by comparing the future location during the trajectory with a possible obstacle position within a preset time horizon. If a collision is imminent, the user command is replaced by a automatic PID control which keeps the last known vehicle's safe location. If no collision is imminent, our method does not override the operators command.

The solution is firstly implemented in a simulator called TUM Simulator, a quadrotor simulator developed by the Technical University of Munich, over the Robotics Operational System (ROS) and the Gazebo simulation platform. The quadrotor's and sonars sensors were characterized to, afterward, embed all solution though a computer-on-module on the platform and control it with a joystick.

2 Related Work and State-of-the-Art

Recently, unmanned aerial vehicles have received greater attention from the robotics community. Many authors have focused on the modelling and control of these vehicles [2], particularly, on quadrotors [3]. In this context, comprehensive studies about the platform configuration, modelling methodologies, comprehensive nonlinear modelling, aerodynamic effects, identification and simulation were performed [4]. For being considered a nonlinear complex system, techniques for quadrotor control also have been widely studied. [5] developed a PID control for a quadrotor to stabilize it in flight. On the other hand, [6] proposed an adaptive neural network to stabilize the quadrotor taking into account wind modelling errors and disturbances.

Several applications are under study on the quadrotor platform. For the majority (e.g., autonomous navigation, multi-agent tasks and obstacle avoidance), estimating the vehicle state is an well established challenge [7]. Regarding to indoor environments, computer vision, which makes use of cameras to image processing, is one of the researching area about the problem [8]. Another area is about using stereoceptive sensors for this task, such as sonars [9] and laser [10]. For outdoor environments, however, Global Positioning System (GPS) is available and some applications using this technology have been developed, such as [11] and [12].

Specifically, with regards to collision avoidance problem, [13] conducted an approach to head quadrotor off the obstacle to a safe position. However, the referred method considers the environment and obstacle positions previously known and being already geometrically preprogrammed, not using real time measurements obtained from on-board sensors. [10] also developed a fully autonomous quadrotor in an indoor environment using, for obstacle avoidance, a laser scanning sensor miniature. [14] used five ultrasonic sensors for that, but both [10] and [14] used a very well-known platform developed and studied in laboratory for years. [9] proposed a real time method, using fast simultaneous localization and mapping (FastSLAM) with occupancy grid mapping [1] to localize the quadrotor and map the environment in a simultaneous way, however, using an off-board approach and validating it in a simulator.

3 Collision Avoidance: Our Approach

Indeed, there are many steps in implementing our proposed method. An overview of its entire architecture is presented in Fig. 1. Firstly, the user's desired control is inputted to the quadrotor, which provides its attitude and velocity through an embedded Attitude Heading Reference System (AHRS). That information is used by both the trajectory estimator and SLAM modules that output the future trajectory and an occupation grid representation of the environment, respectively. Both information are confronted each other by a collision decision block which detects an imminent collision. If negative, the system does nothing. Otherwise, a position controller determines the automatic control necessary to keep the last safe position. Finally, the user control is overridden and the automatic control is passed to the quadrotor. Following are the topics supporting this architecture.

3.1 *The Quadrotor Platform*

A quadrotor is classified as an under-actuated system. It receives four inputs, given by the thrusts generated vertically by its four rotors, and produces six outputs, represented by the three angular movements roll (ϕ), pitch (θ) and yaw (ψ) and the three linear movements (x, y, z) [15]. The movements are generated combining the vari-

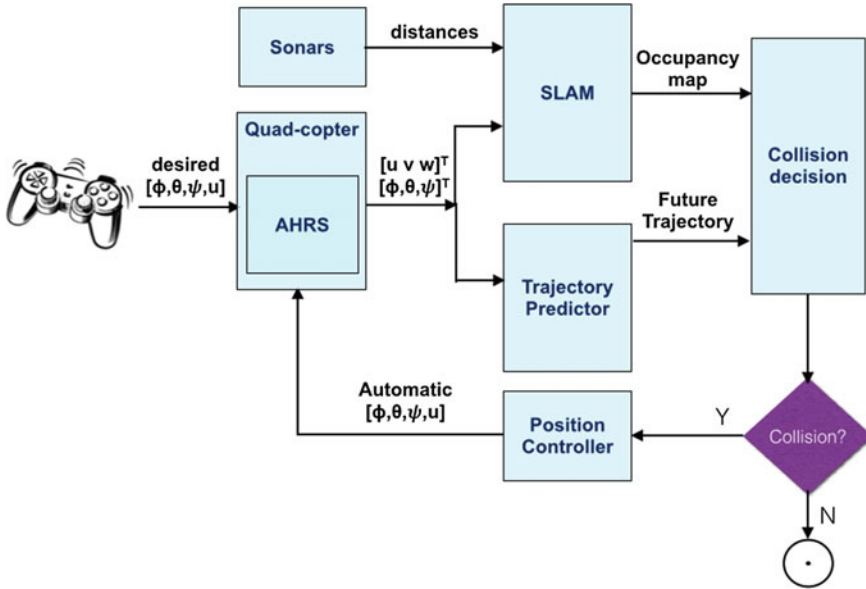


Fig. 1 An overall representation of our proposed architecture

ation of each rotor's thrust, with one pair of opposed rotors rotating clockwise and the other one counter-clockwise to balance the total torque of the system [16]. Those characteristics are shown in Fig. 2.

In order to develop our method, it was used a robust small quadrotor with on-board electronic stabilization, equipped with a front-camera and a down-looking camera, making use of the latter to improve stability. Furthermore, it is equipped with a sonar pointing downwards to measure altitude and an AHRS measuring (ϕ, θ, ψ) and linear velocities along all axes. The vehicle is controlled by sending commands over a WI-FI connection and outputs its navigation data, including acceleration, velocity and attitude at frequency of 200 Hz [17].

3.2 Future Trajectory Estimation

The quadrotor dynamics supports this context. Considering that we are working with a quadrotor that gives its acceleration, velocity and attitude over the time, the source of that data is neglected and only its kinematics is considered.

The transformations related to the study of kinematics involve one translation and successive rotations from the inertial referential \mathcal{F}^i until the body-fixed referential \mathcal{F}^b . Figure 3 shows the transformations step-by-step from \mathcal{F}^i , passing by three inter-

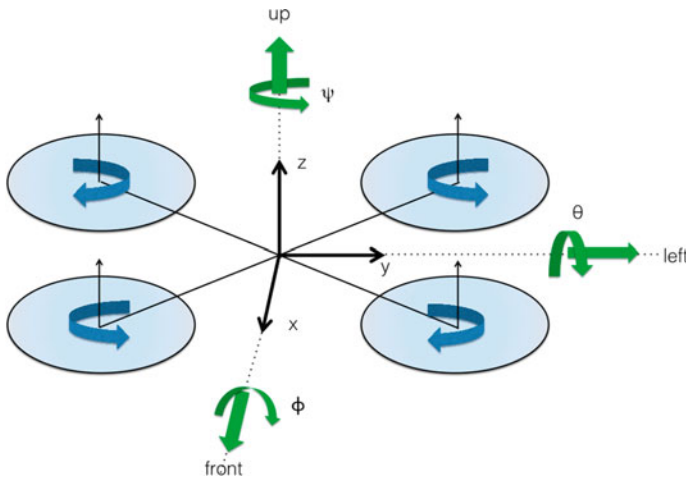


Fig. 2 A simplified quadrotor model with *arrows* indicating the rotors' directions in *blue* and the angular (ϕ, θ, ψ) and linear (x, y, z) movements in *green*

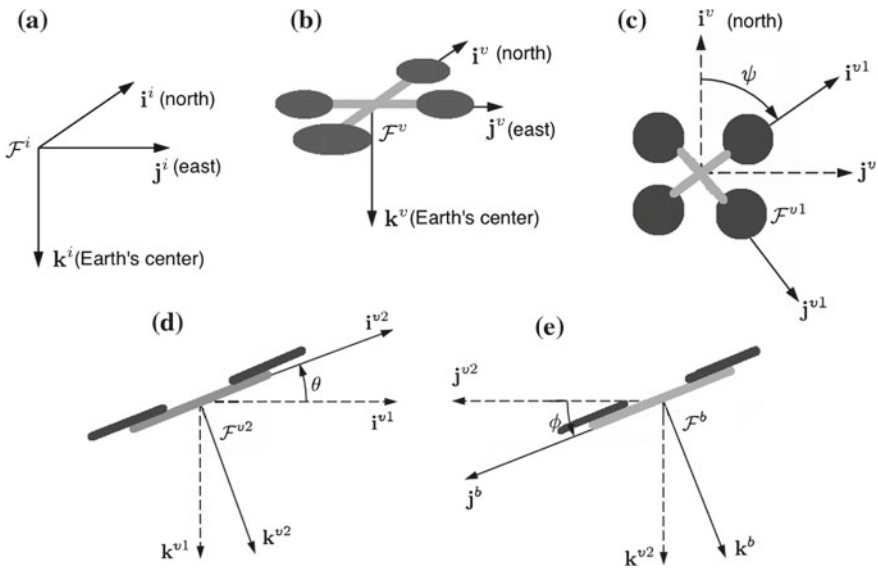


Fig. 3 Referentials related with the study of the quadrotor dynamics from the inertial referential \mathcal{F}^i (a), with a translation to \mathcal{F}^v (b) and three successive rotations ψ, θ, ϕ to \mathcal{F}^{v1} (c), \mathcal{F}^{v2} (d) and the body-fixed referential \mathcal{F}^b (e), respectively

mediaries referentials, including one translation from \mathcal{F}^i to \mathcal{F}^v and three rotations ψ - θ - ϕ for the sequence \mathcal{F}^v - \mathcal{F}^{v1} - \mathcal{F}^{v2} - \mathcal{F}^b [18].

The rotation matrix $R_v^b(\phi, \theta, \psi) = R_{v2}^b(\phi)R_{v1}^{v2}(\theta)R_v^{v1}(\psi)$ represents the transformation from \mathcal{F}^v to \mathcal{F}^b and the matrix W describes the relation between the vehicle's

angular velocity ω and Euler's Angles variation $\dot{\theta} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]$ [18]. Given the global quadrotor's current position \mathbf{x} and its current measurements: linear acceleration $\ddot{\mathbf{x}}$, linear velocity \mathbf{v} , attitude θ , angular velocity ω and angular acceleration $\dot{\omega}$, the trajectory is calculated by an integration over time, considering the previous quantities as initial conditions. Both accelerations $\ddot{\mathbf{x}}$ and $\dot{\omega}$ are constant during the trajectory. That is reasonable given the short period of time for trajectory estimation. One step of the prediction is presented in Eq. (1) (where dt is a constant representing the duration time between two steps).

$$\begin{aligned}
 \omega_t &= \omega_{t-1} + dt * R_b^v * \dot{\omega} \\
 \dot{\theta}_t &= W^{-1} * \omega_t \\
 \theta_t &= \theta_{t-1} + dt * \dot{\theta}_t \\
 \dot{x}_t &= \dot{x}_{t-1} + dt * R_b^v * \ddot{\mathbf{x}} \\
 x_t &= x_{t-1} + dt * \dot{x}_t
 \end{aligned} \tag{1}$$

3.3 Simultaneous Localization and Mapping

Our collision avoidance problem considers the fact that we are in an unknown environment and under sonar measurements uncertainly. The objective of SLAM is to estimate position of a robot and its nearby map simultaneously in a probabilistic fashion (Eq. 2). Let $x_{1:t}$ denote the path of the robot, m the map, $z_{1:t}$ all measurements and $u_{1:t}$ all control inputs where $1 : t$ represents the time step from 1 to t . The conditional independence propriety of the SLAM problem [1] allows us to perform a factorization that decomposes SLAM into a independent localization estimation and mapping problem (Eq. 3). Our approach is simpler and only considers odometry data to calculate the localization and is showed in Eq. 4.

$$p(x_{1:t}, m | z_{1:t}, u_{1:t}) \tag{2}$$

$$p(x_{1:t} | z_{1:t}, u_{1:t}) p(m | z_{1:t}, x_{1:t}) \tag{3}$$

$$p(m | z_{1:t}, x_{1:t}) \tag{4}$$

Getting the data given by the quadrotor at a frequency of 200 Hz as shown in Sect. 3.1, it is possible to integrate over the time and get its global location. Given x_{t-1} the position in $t - 1$, v_t the velocity and a_t the acceleration provided by the quadrotor, the position x_t is calculated as demonstrated in the Eq. 5.

$$x_t = x_{t-1} + v_t * \Delta t + a_t * 0.5t^2 \tag{5}$$

In general, the imprecision and noise of sensors' data in calculating distances, including sonars, lead us to adopt probabilistic approach for mapping. Occupancy

grid maps address the problem of generating consistent maps from noisy and uncertain measurement data, given the robot pose [1].

We adopted the Octomap to generate volumetric 3D environment models, based on octrees and probabilistic occupancy estimation. It explicitly represents not only occupied space, but also free and unknown areas. Its central property is its efficiency and the ability of promoting probabilistic updates of occupied and free space while keeping memory consumption at a minimum [19].

3.4 Our Approach for Collision Detection

A possible collision is verified based on an intersection between the region occupied by the obstacle and the quadrotor. Here, the quadrotor's region is simplified as a sphere and the obstacle's region includes unknown areas and the area increased by the quadrotor sphere radius. During the flight, the scanning collision area is constrained as the possible affordable positions in the interval $t \in [0, \tau]$ while the future trajectory is constantly calculated. Both are based on the quadrotor dynamics and state, including position and velocity.

Given the vehicle position $\mathbf{x} = (x, y, z)$, the scanning collision area is delimited as follows. Initially, the affordable distance from its position \mathbf{x} , denoted by Δ_x , is defined by the Eq. 6 where v_{max} is a fixed value representing the vehicle maximum affordable velocity and τ is the configured forwarding time. Afterwards, all the map grid cells are covered in a loop through the volume formed by the interval $[\mathbf{x} - \Delta_x/2, \mathbf{x} + \Delta_x/2]$. Furthermore, each future trajectory step is evaluated against the current grid cell, looking for an occupied area indicating a possible collision with the quadrotor area. The Fig. 4 illustrates our approach. The environment is depicted as a grid. Region **A** represents the quadrotor's sphere with radius r and **CB** is the obstacle region. The collision scanning area is in yellow and its intersection with **CB** is in orange showing all possible positions for a potential collision. A future trajectory is represented by the blue arrow leading to the future position p at time τ , which demonstrates an imminent collision.

$$\Delta_x = v_{max} * \tau \quad (6)$$

3.5 Controlling to Safe Position

The last known safe position is determined when a future trajectory is estimated and a colliding path is found. In this case, an automatic control is enabled and overrides the user input for 3 (three) seconds. This approach helps to maintain the user's experience letting the quadrotor stop while keeping its attitude and position.

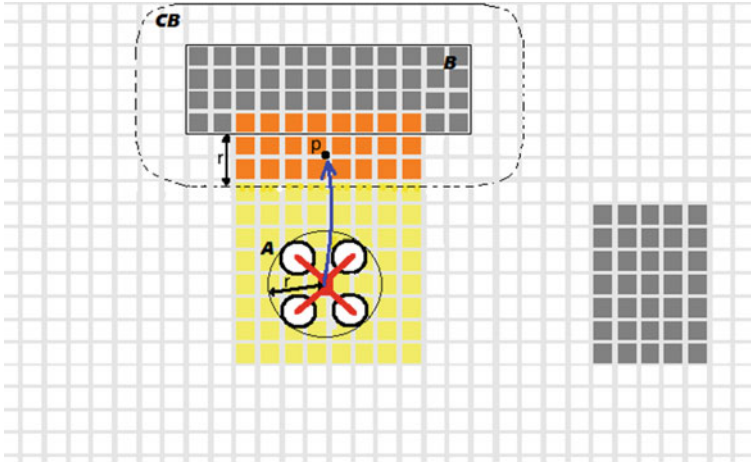


Fig. 4 Illustration of the intersection (in *orange*) between the collision scanning area (in *yellow*) and the obstacle region (CB) during flight. The quadrotor is represented by the sphere region A with radius r

Four PID Controllers were implemented as a combination in order to control the quadrotor global position (x, y, z) and yaw (ψ) [3]. Figure 5 shows the controller scheme. Error inputs are calculated regarding to the difference between the desired safe position and ψ (i.e. desired $[x, y, z, \psi]^T$) and its current data obtained from a global position estimator using the quadrotor AHRS measurements. Then, they are mapped to velocity commands accordingly to the range of acceptable vehicle commands. The control is calculated at the same frequency that the platform sends its navigational data (200 Hz) during 3 (three) seconds. The proportional, integrated and derivative constants k_p , k_i and k_d were measured empirically.

4 Experimental Setup and Results

The solution was implemented in C++ language. An Octomap cell resolution of 0.1 m was chosen allowing to have a good environment representation regarding to the solution performance. Some tests were executed to determine efficiency and its correctness. All data used in our experiments was obtained through the TUM Simulator over the Robotics Operation System (ROS), which models the quadrotor with fidelity and allow us to communicate with the vehicle over WI-FI and to get its navigation data in real time. Figure 6 shows three different simulator's screen-shots running our solution evidencing the scanning collision areas being mapped (in yellow) over the time (t_k) while the quadrotor trajectory towards to the obstacle (arrows in green). In an imminent collision (orange area), an automatic control is generated (arrow in red) and stops the vehicle safely.

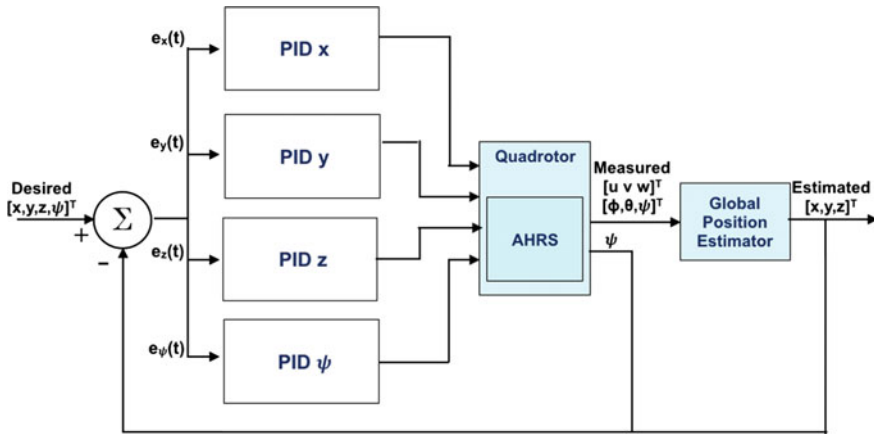


Fig. 5 PID controller scheme to lead the quadrotor to its last safe pose. The platform is controlled by four PID controllers for x , y , z and ψ , which receive feedback from the global position estimator

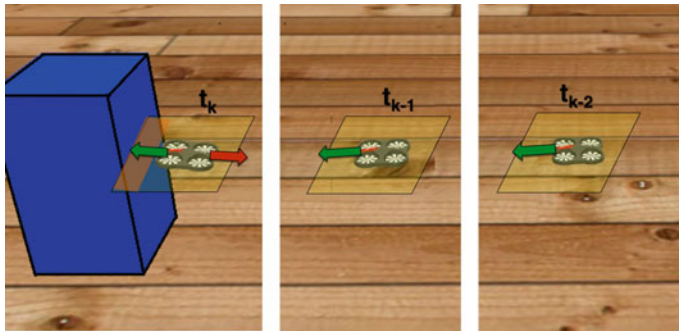


Fig. 6 Three different simulator’s screen-shots taken while the quadrotor’s trajectory towards to the obstacle and scanning collision areas are mapped

In order to suit our demand, an off-the-shelf platform (i.e., Parrot AR.Drone) was chosen. Besides the characteristics observed in Sect. 3.1, it is worth highlighting that it presents flight autonomy about 15 (fifteen) minutes, maximum payload of 100 (one hundred) grams and dimensions of 0.51m^2 , which are sufficient to embed and evaluate our solution. A study about its data accuracy, given by its standard development kit (SDK), was realized to understand the uncertainty generated when calculating its global position. One thousand samples of the Parrot AR.Drone’s attitude were obtained at 200 Hz while it was in idle state. The average ($\bar{\theta}$) and standard deviation (σ_{θ}) were calculated and shown in Table 1. That information is relevant when defining a minimal security distance to obstacles. The measurements are in degrees.

In order to measure the distance to obstacles and map the nearby environment, four sonar sensors Maxbotics I2CXL-MaxSonar-EZ (S1, S2, S3, S4) were evaluated.

Table 1 Parrot AR.Drone attitude average and standard deviation in degrees

	Roll (ϕ)	Pitch (θ)	Yaw (ψ)
$\bar{\theta}$	-0.1643	0.2958	5.2059
σ_{θ}	0.0643	0.0363	0.0339

Their range measurements are distances from 0.2 to 2.55 m at 10 Hz and are modeled centered above the quadrotor pointing horizontally in the forward direction. A study about their imprecision was done by getting six hundreds samples from each one of the distances 0.5, 1.0, 1.5 and 2.0 m. Table 2 gives their average (\bar{X}) and standard deviation σ in meters, after eliminating outliers obtained by Boxplot Method [20]. The number of eliminated outliers, in each case, is also shown in Table 2.

The forwarding time τ has influence in the solution behavior as a configuration parameter. The smaller is τ , the shorter will be the scan collision area and closer to the obstacle the quadrotor will stop. Table 3 gives the distances obtained when τ equals to 0.5, 1.0, 1.5 and 2.0 s. Due to quadrotor's sensors and sonars imprecisions, a safe stop distance to obstacle was considered (10 cm) setting τ equals 1.0 s.

The automatic control was calibrated by tuning empirically the four PID controllers' constants k_p , k_d and k_i . An experiment, with the quadrotor in idle state at position $x = 0$, $y = 0$, $z = 0$ and $\psi = 0$, was conducted to lead it to the position $x = 0.5$, $y = 0.5$, $z = 0.5$ and $\psi = 1$ rad. Four combinations of k_p , k_d and k_i were evaluated to determine which one has major influence on the automatic control behavior over time. The faster is the stabilization time combined with a lower oscillation, the better is the automatic control response time with minor risk to assume an unknown position. For both x , y and yaw , $k_p = 2.0$, $k_d = 0.0$ and $k_i = 0.0$ stabilized first with a lower oscillation. In case of z , $k_p = 2.0$, $k_d = 2.0$ and $k_i = 0.0$ presented a better result.

In our experiment, the program managed to fulfill its objective and avoid the crash despite the orders from the user to continue its trajectory towards the obstacle. The quadrotor was represented as sphere model with radius of 0.26 m and a security distance of 0.1 m from the obstacle was considered. Figure 7 presents the satisfactory behavior of our solution when the vehicle is headed toward the obstacle while the controller is managed at full throttle. The distance is represented in blue (Fig. 7a) and keep decreasing until our security distance of 0.1 m from the obstacle (in orange). From that moment, the distance is kept constant even when the maximum velocity command is applied. Figure 7b shows the velocity command magnitude of approximately 1.1 m/s being applied to the quadrotor during the trajectory practically constant over the time. This, fortunately, is our desired accomplishment.

The obtained results, even considering some imprecision from the quadrotor's sensors and sonars, match our expectations and can be embedded on the platform to complete the implementation.

Table 2 Maxbotics I2CXL-MaxSonar-EZ measurements average and standard deviation in meters and the number of outliers

Sonar	0.5 m			1.0 m			1.5 m			2.0 m		
	\bar{X} (m)	σ (m)	Outliers	\bar{X} (m)	σ (m)	Outliers	\bar{X} (m)	σ (m)	Outliers	\bar{X} (m)	σ (m)	Outliers
S1	0.50	0.0000	27	1.00	0.0048	28	1.49	0.0048	113	1.99	0.0033	131
S2	0.50	0.0008	30	1.00	0.0050	29	1.49	0.0017	133	2.00	0.0009	121
S3	0.50	0.0008	34	1.00	0.0025	41	1.49	0.0018	135	2.00	0.0010	141
S4	0.50	0.0032	28	1.00	0.0015	34	1.50	0.0026	139	2.00	0.0033	123

Table 3 Distance to obstacle when the quadrotor stops considering different forwarding times τ

τ (s)	Distance to obstacle (m)
0.5	0.031915
1.0	0.104508
1.5	0.518959
2.0	0.907943

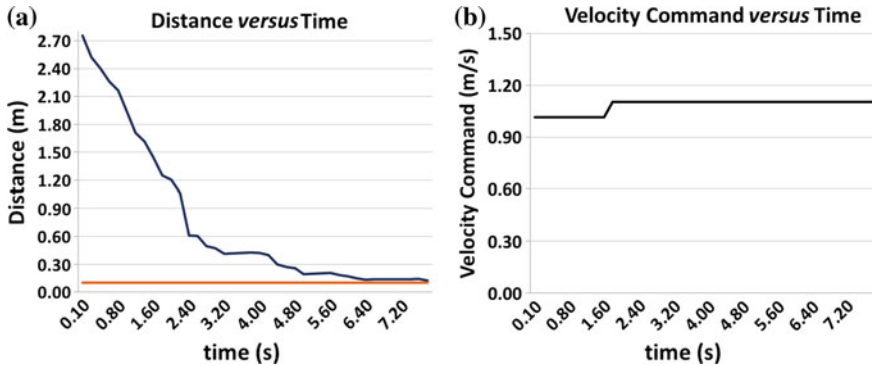


Fig. 7 Experiment showing the system behavior when the quadrotor is toward the obstacle

5 Conclusion

In this paper, we presented an approach to avoid collision automatically assisting the manual control of a quadrotor. Our experiments in a simulated environment accomplished the objective being capable to avoid collisions in totally unknown environments even when the user’s commands were contrary or unable to do so. When at risk, the inputs are overridden and an automatic control assumes for some time keeping the quadrotor in its last safe known position. The solution took into account the uncertainty from real quadrotor’s and sonar sensors, being ready to be embedded on the quadrotor through a computer-on-module. As for future work, we are currently working at embedding our solution on the quadrotor, enabling it suitably to complete missions in a much safer manner.

Acknowledgements This research is partially funded by CAPES (Brazilian Research Agency) through Edital Pro-Estrategia N^o 050/2011 and Demanda Social and Petrobras SA.

References

1. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2005)
2. Ross, S., Melik-Barkhudarov, N., Shankar, K.S., Wendel, A., Dey, D., Bagnell, J.A., Hebert, M.: Learning monocular reactive UAV control in cluttered natural environments. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 1765–1772. IEEE (2013)
3. Lugo, J.J., Zell, A.: Framework for autonomous on-board navigation with the AR.Drone. *J. Intell. Robot. Syst.* **73**(1–4), 401–412 (2014)
4. Zhang, X., Li, X., Wang, K., Lu, Y.: A survey of modelling and identification of quadrotor robot. In: Abstract and Applied Analysis, vol. 2014. Hindawi Publishing Corporation (2014)
5. Salih, A.L., Moghavvemi, M., Mohamed, H.A., Gaeid, K.S.: Flight PID controller design for a UAV quadrotor. *Sci. Res. Essays* **5**(23), 3660–3667 (2010)
6. Nicol, C., Macnab, C., Ramirez-Serrano, A.: Robust neural network control of a quadrotor helicopter. In: Proceedings of the Canadian Conference on Electrical and Computer Engineering, 1233–1237 (2008)
7. Ito, K., Han, J., Ohya, A.: Localization using uniaxial laser rangefinder and IMU for MAV. In: 2014 IEEE/SICE International Symposium on System Integration (SII), pp. 712–717. IEEE (2014)
8. Shen, S., Mulgaonkar, Y., Michael, N., Kumar, V.: Vision-based state estimation for autonomous rotorcraft MAVs in complex environments. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 1758–1764. IEEE (2013)
9. Mendes, J., Ventura, R.: Assisted teleoperation of quadcopters using obstacle avoidance. *J. Autom. Mobile Robot. Intell. Syst.* **7**(1), 54–58 (2013)
10. Grzonka, S., Grisetti, G., Burgard, W.: A fully autonomous indoor quadrotor. *IEEE Trans. Robot.* **28**(1), 90–100 (2012)
11. Kerns, A.J., Shepard, D.P., Bhatti, J.A., Humphreys, T.E.: Unmanned aircraft capture and control via GPS spoofing. *J. Field Robot.* **31**(4), 617–636 (2014)
12. Roberts, A., Tayebi, A.: A new position regulation strategy for VTOL UAVs using IMU and GPS measurements. *Automatica* **49**(2), 434–440 (2013)
13. Israelsen, J., Beall, M., Bareiss, D., Stuart, D., Keeney, E., van den Berg, J.: Automatic collision avoidance for manually tele-operated unmanned aerial vehicles. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 6638–6643 (2014)
14. Becker, M., Sampaio, R.C.B., Bouabdallah, S.: Perrot, Vincent de Siegwart, R.: In-flight collision avoidance controller based only on OS4 embedded sensors. *J. Braz. Soc. Mech. Sci. Eng.* **34**(3), 294–307 (2012)
15. Gupte, S., Mohandas, P.I.T., Conrad, J.M.: A survey of quadrotor unmanned aerial vehicles. In: 2012 Proceedings of IEEE SoutheastCon, pp. 1–6. IEEE (2012)
16. Mian, A.A., Daobo, W.: Modeling and backstepping-based nonlinear control strategy for a 6 DOF quadrotor helicopter. *Chin. J. Aeronaut.* **21**(3), 261–268 (2008)
17. Dijkshoorn, N., Visser, A.: Integrating sensor and motion models to localize an autonomous AR.Drone. *Int. J. Micro Air Veh.* **3**(4), 183–200 (2011)
18. Beard, R.W., McLain, T.W.: Small Unmanned Aircraft: Theory and Practice. Princeton University Press, Princeton (2012)
19. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Auton. Robots* **34**(3), 189–206 (2013)
20. Dawson, R.: How significant is a boxplot outlier. *J. Stat. Educ.* **19**(2), 1–12 (2011)

Part IV
Hand

Shared Control with Flexible Obstacle Avoidance for Manipulator

Zhixuan Wei, Weidong Chen and Hesheng Wang

Abstract This paper presents a new control method that enable the robot pushes the obstacle while it moves to the target. In this paper, the shared control combines the user command—always directed at the target, and the machine command—to control the robot to avoid obstacles. And traditional obstacle avoidance requires to find a collision free path, which will find no solution in clutter environment. With pushing obstacles, the robot can solve a problem which was unsolvable, or find a more optimal solution (such as less motion for all joints). However, how to ensure the safety of pushing is a challenge. In this paper, an enhanced artificial potential field obstacle avoidance method is proposed, and a potential field for guaranteeing the safety of pushing is defined. The proposed method is test in simulation system, and the result shows that the proposed method is effective.

Keywords Motion control · Shared control · Obstacle avoidance

1 Introduction

Consider the motion of a human when he wants to get an object behind obstacles. In addition to avoid the obstacles, the human will also push some obstacles, in order to get a more optimal motion, such as less path length. However, the robot will avoid

Z. Wei (✉) · W. Chen (✉) · H. Wang

Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China
e-mail: zhixuan.wei@sjtu.edu.cn; zhixuan.wei@gmail.com

W. Chen

e-mail: wdchen@sjtu.edu.cn

H. Wang

e-mail: wanghesheng@sjtu.edu.cn

Z. Wei · W. Chen · H. Wang

State Key Laboratory of Robotics and System (HIT), Harbin 150001, China

© Springer International Publishing AG 2017

W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_17

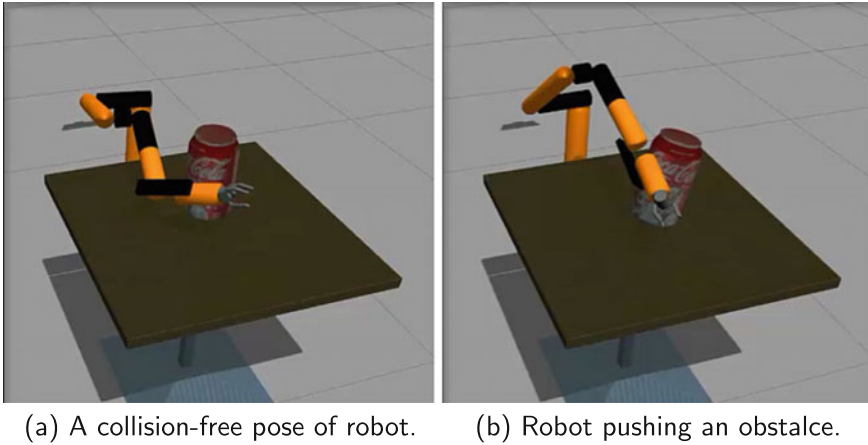


Fig. 1 Comparing the pose of robot

any collision with obstacles during the motion. The reason is that human has the knowledge of the obstacles about how to contact the obstacle in safe.

An example is shown in Fig. 1. Figure 1a shows a robot controlled by a traditional controller will find a collision free pose. However, if the robot pushes obstacle like Fig. 1b, it is a more nature pose.

Obstacle avoidance is one of the fundamental problem in shared control. The traditional obstacle avoidance, such as MVFH&VFF used in [1, 2], requires to find a collision free path, which will find no solution in clutter environment.

In this paper, we consider the motion controller by which the robot can push the obstacle to change the environment. There are several advantages by changing environment: solve the problem which was no solution; get a more optimal solution; and so on.

Several researchers worked on Navigation Among Movable Obstacles (NAMO) [3, 4], which is a method for guiding heuristic search in the computationally challenging domain of motion planning among movable obstacles. In NAMO, moving obstacles and moving to the goal are separated steps. However, in our method, the robot pushes the obstacles and moves to the goal at the same time. Hence, NAMO didn't consider the pushing problem.

The flexible obstacle avoidance is defined as the robot can push some obstacles during the motion under safe. Since the robot pushes the obstacle and moves to the goal at the same time, the motion of pushing is part of motion toward to the goal. Hence it is more difficult than moving the obstacles alone.

In paper [5, 6], Miyazawa proposed a planning method, which plan the motion to the push the object to go to a goal pose and avoid obstacles with multi fingers of robot hand. Different with above mentioned pushing method, we don't consider the accurate force analysis between the robot and the obstacle, since in our method, the contact part between robot and obstacle is not limited. Instead of analysis of forces

of obstacle, a cost space is build to represent the risk while the robot interacts with obstacle.

This paper is organized as follow: In Sect. 2, we introduce the idea of flexible obstacle avoidance. The definition of flexible potential field is given and flexible potential field estimation method is presented. As our previous work [7], how to use semantic mapping to build the environment potential field map is described in Sect. 3. In Sect. 4, the shared controller using environment potential field map to adjust user's input command is proposed. A simulation result of proposed method is given in Sect. 5. Finally, this paper is concluded in Sect. 6.

2 Flexible Obstacle Avoidance

Obstacle avoidance is a fundamental problem in robotics. Obstacle avoidance usually find collision free path in order to ensure safe of robot and other object. However, although avoiding contact between robot and obstacle is simple and effective way to ensure safe, sometime proper contact is also safe while more effective.

Pushing object has many types: pushing the object sliding on the support plane or pivoting [8] move the position of object; tilting only change the orientation of object, and when the robot leave the object, the object will restore the original pose. All above belong to grasplless manipulation.

Flexible obstacle avoidance is defined as the robot contact with obstacles by any part of the robot and do some grasplless manipulation, while the robot moves to the target.

An advantage of flexible obstacle avoidance is that since pushing obstacles will change environment, controller or planner with flexible obstacle avoidance may solve a problem which was unsolvable, or find a more optimal solution.

Flexible obstacle avoidance also has a lot of challenge. First, grasplless manipulation refers to contact dynamics between object and robot, as well as between object and support plane, which is difficult to accurately modeling. Second, the environment is dynamic since pushing obstacles will change environment. Third, the environment is unpredictable due to the motion of obstacle is difficult to model.

2.1 Flexible Potential Field

In order to respond the challenge mentioned in the last section, the flexible potential field is proposed. The proposed method is inspired by obstacle avoidance algorithm based on artificial potential field. The flexible potential field is defined to describe the cost produced during the motion of pushing the obstacle. Usually, to avoid collision between robot and obstacle, the artificial potential field is defined around the obstacle. However, in order to control the robot to push obstacle in a certain extent,

the flexible potential field is inside the obstacle. The detail of flexible potential field is described in following section.

In this paper, flexible cost is defined as the cost produced during the motion of pushing the obstacle. The bigger the cost, indicating that the larger the displacement of obstacle due to being pushed by robot. The danger behavior, such as pushing over the obstacle, will generate a cost large enough which will exceed the threshold.

Flexible potential field consists of a scalar field and a vector field. In scalar field, the value of each point represents the cost while robot move over this point. In vector field, the value and direction of each point represents the resistance when robot moves over this point.

2.2 Flexible Space

Flexible space is a flexible potential field distributed in the space occupied by the obstacle. The value of each point relates to the level of risk when the obstacle was pushed by the robot.

Usually, the level of risk when the obstacle was pushed by the robot can be represented by the inclination of the obstacle. The larger the inclination, the greater the risk. If the inclination is larger than some threshold, the obstacle will fall. However, it's hard to predict the inclination of obstacle in motion planning.

The depth of penetration between robot and obstacle is easy to estimation by planer. And the depth relates to the inclination of the obstacle pushed by the robot at the same position. Hence the depth can be used to represent the level of risk when the obstacle was pushed by the robot.

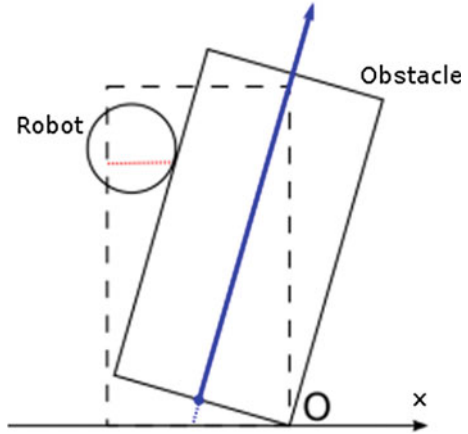
To guarantee the safe of obstacle, just like the inclination must smaller than a threshold, the depth of penetration is also limited. The surface composed by the threshold of depth is defined as boundary surface. For the scalar flexible potential field, the value of the point is defined as: it is infinity inside the boundary surface; from boundary surface to the edge of the obstacle, the value decreases; outside the edge of the obstacle, it is zero. The vector field is the gradient of the scalar field.

2.3 Flexible Potential Field Estimation

In this section, a cylinder object with uniform density is considered.

Since the cylinder is the axis of symmetric, the motion of a longitudinal section is analyzed. As show in Fig. 2, considering a static situation, the object will fall only when the projection of the center of mass of the object is out of the bottom of the object. The critical state is the center of mass is just above the supporting point of object. And at the critical state, the boundary b is a part of boundary surface. Since the cylinder is the axis of symmetric, rotating the boundary b along the axis of cylinder will generate a surface like cone. The cone is a coarse boundary surface.

Fig. 2 Quasi-static analysis



The lower part of the cone is incomplete. Considering the size of robot, the lower part of flexible space is ignored. The higher part of cone is crossed. In order to simplify the motion planning, the higher part is modified into a cylinder around the center axis with a certain radius. Finally, the boundary surface is like a flask as shown in Fig. 3.

Then the flexible space is defined. The scale field is shown in Fig. 4. As previously described, the value inside the boundary surface should be infinity. In practice, a value bigger than a threshold is enough. The value outside the boundary surface distributes as a function $f(r_p, h)$, in which r_p is the distance between the point to the center axis, and h is the height of the point. $f(r_p, h)$ satisfies the following conditions:

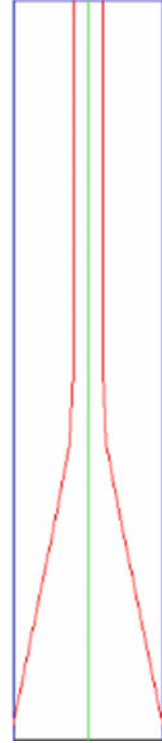
$$\begin{cases} f(B_s(h), h) = V_{in} \\ f(B_c(h), h) = 0 \\ f'(r_p, h) < 0, & B_s(h) < r_p < B_c(h) \\ f''(r_p, h) < 0, & B_s(h) < r_p < B_c(h) \end{cases} \quad (1)$$

in which $B_s(h)$ is radius the boundary surface at h , $B_c(h)$ is radius the cylinder at h , and V_{in} is the value inside the boundary surface.

The reasons why $f(r_p, h)$ is designed to be so are: first two lines of Eq. 1 determines the range of $f(r_p, h)$; third line of Eq. 1 ensure the smaller the r_p , the greater the $f(r_p, h)$, which means the deeper the robot pushing the object, the bigger the cost; fourth line of Eq. 1 ensure the smaller the r_p , the greater the $f'(r_p, h)$, which means the deeper the robot pushing the object, the speed of robot should be slower.

And the vector field is shown in Fig. 5. As previously described, the vector field is the gradient of the scalar field. The color represents the value of resistance, and the resistance is maximum when robot moves along the arrow.

Fig. 3 Boundary surface of a cylinder object. *Blue line* is the outer edge of cylinder. *Red line* is the boundary surface of the cylinder



3 Semantic Mapping

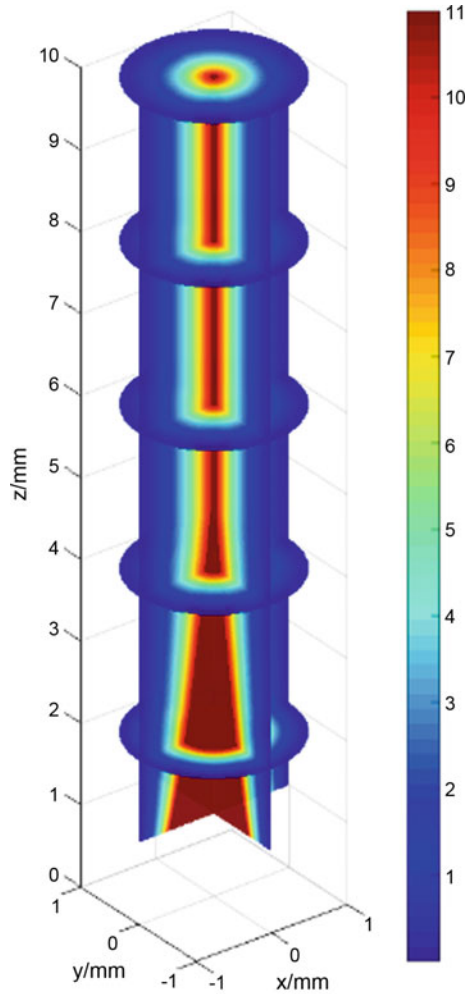
Semantic map provides the semantic and geometrical information of recognized objects. Semantic information includes category, size, color and other properties, as well as the flexible potential field of this object.

Our semantic mapping method mixed rule-based reasoning and recognition. For room level structure extraction, the object larger than the view range of the sensor, the observation is incomplete and only captures a part of object. This object usually lacks of feature for recognition, such as a wall. However, this object usually conforms to special shape rules, and can easily be extracted by the reasoning-based method.

Reasoning-based method has two steps. First, the input 3D point cloud is segment into candidate units by shape. Usually, for the room level structure such as wall, floor, cell, door, candidate units are plane. Then, features of candidate units are extracted. For plane as example, the features include area, size, position. And features of relationship between two planes are also used in reasoning. The features of relationship include connectivity and relative position between two planes. A forward chaining is used to reason the category and other semantic information of object.

For the detail of this method, please refer to our previous works [2, 7].

Fig. 4 Scale flexible potential field of a cylinder object



4 Shared Control

The system architecture is shown in Fig. 6. The potential field of environment is build real time from sensor data. The shared controller combines potential field and robot pose information to estimate the collision situation between obstacle and robot, and adjusts the user input to avoid dangers.

In the last sections, we have got the semantic map and the potential field of obstacles. In the semantic map, it contents obstacle position and category, then combining above information and the potential field of the obstacles is the potential field map of environment.

Fig. 5 Vector flexible potential field of a cylinder object

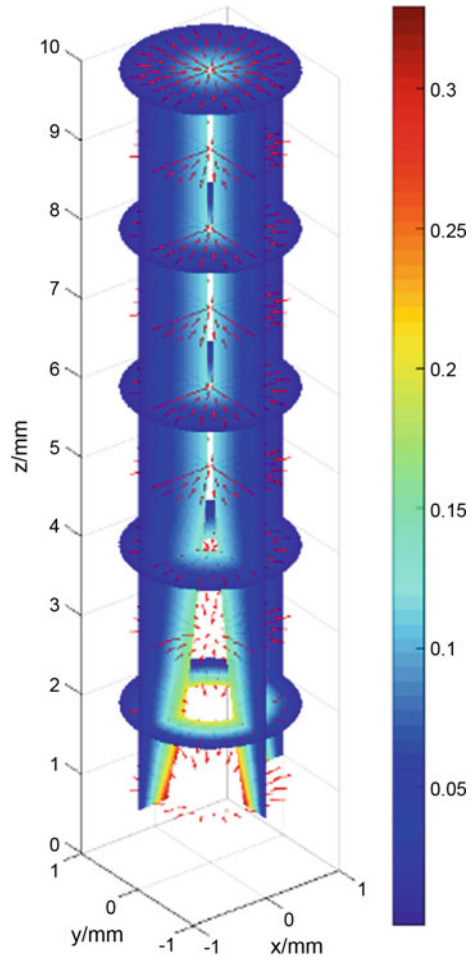
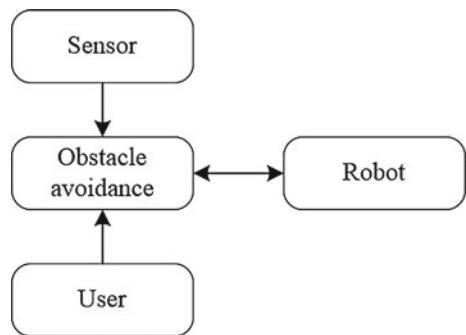


Fig. 6 System architecture



The potential field map describes the distribution of obstacles and how the robot should reflect. The main idea is sum two vectors which are user input and potential force of obstacle. x is the position of robot, $p_i(x)$ is the potential force of obstacle i at x , $\max p_i$ is the maximum force of the potential force of obstacle i , \dot{x}_u is the user command and \dot{x}_c is the final command sent to robot.

$$p'_i(x) = \frac{|\dot{x}_u|}{|\max p_i|} p_i(x) \quad (2)$$

As shown in Eq. 2, the $p(x)$ is limited by user command. The limitation ensures $p'_i(x) \leq \dot{x}_u$. Therefore, $p'_i(x) = 0$ when $\dot{x}_u = 0$.

$$\dot{x}_c = \dot{x}_u + \sum_i p'_i(x) \quad (3)$$

As shown in Eq. 3, the final command \dot{x}_c is the vector sum of user command input and potential forces of all collision obstacles.

5 Simulation

5.1 Simulator

In this approach, the control software is based on Robot Operation System (ROS) [9] and Point Cloud Library (PCL) [10]. Gazebo is used as simulator. The control processes and user interface is programmed in ROS. The semantic mapping is realized with PCL. Gazebo can simulate the physical world with collision and force between different objects. Sensor also can be simulated in Gazebo.

In order to simulate the user input, a fake teleoperation panel¹ included by rviz (a visualization of ROS) is used, by which the user can use mouse to send velocity command in $\dot{x}_u = (v_x, v_y)$.

The simulated environment is shown in Fig. 7. A two degrees of freedom robot is used (the left part of Fig. 7 which is in orange and black). The two joints are along vertical axis and the end effector of robot can move in horizontal plane. The workspace of the end effector is a circle which radius is 0.9 m. Two cans are used to simulate the obstacles pushed by robot. The can is about 0.3 m height and radius is 0.05 m. The center of mass of the can is the same with the geometry center. Cans are put on a table to simulate the friction force which is necessary to test the pushing motion. And a simulated Kinect is used as sensor to obtain the environment information (on the top right corner of Fig. 7).

¹http://docs.ros.org/indigo/api/rviz_plugin_tutorials/html/panel_plugin_tutorial.html.

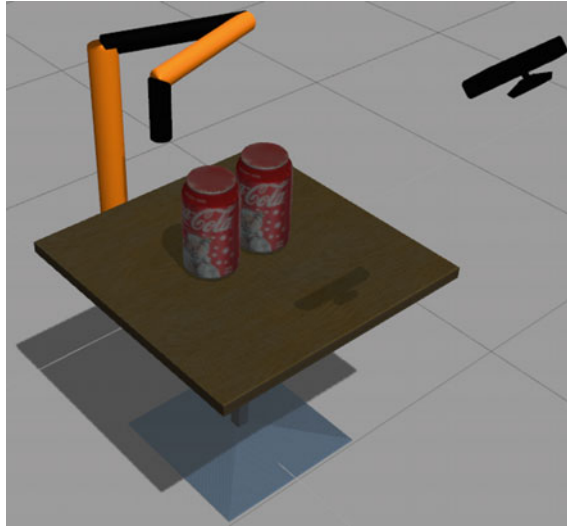
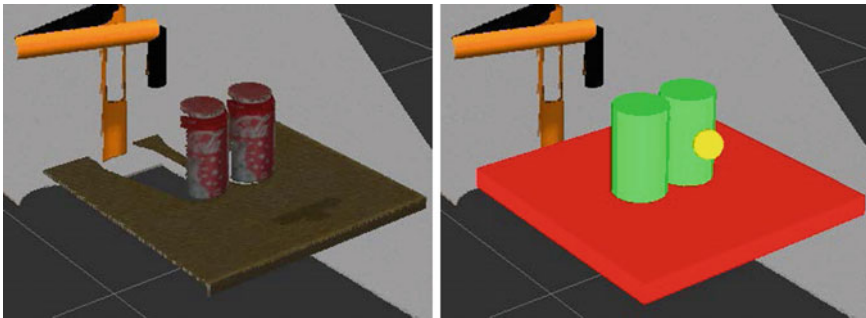


Fig. 7 Simulated environment



(a) Point cloud obtained by Kinect.

(b) The obstacles recognized by semantic map.

Fig. 8 Semantic mapping result

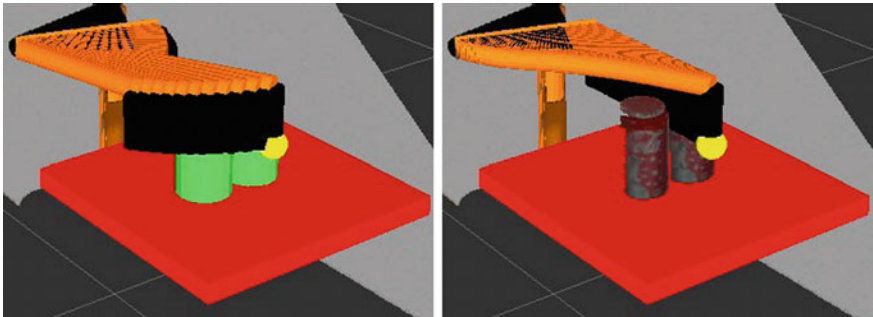
5.2 Result

The semantic mapping result is shown in Fig. 8. The Fig. 8a shows the 3D point cloud obtained by Kinect. Although the point cloud is not complete and miss some parts because of the view point of sensor and the occlusion among different objects, the semantic mapping result still describes the environment accurately.

As shown in Fig. 8b, the objects in workspace are represented by semantic map. The red box and green cylinders are table and cans. The yellow ball represents the target of task. The target is changing in different tasks.

Three comparison experiments in different task targets are tested. The first target is behind the middle of two cans, and other two targets are on the left and back of the first target. These target cover different distance and different position relating to obstacles.

Each experiment of each task is tested in ten times. The user uses the teleoperation panel to send command, which always control the robot move directly to the target. One of the comparison experiment (using traditional obstacle avoidance) result is shown in Fig. 9. Obviously, since the target is behind the middle of two cans, the



(a) Controlled by traditional obstacle avoidance.

(b) Controlled by proposed method.

Fig. 9 Comparison experiment result

Fig. 10 Robot pushes the obstacle

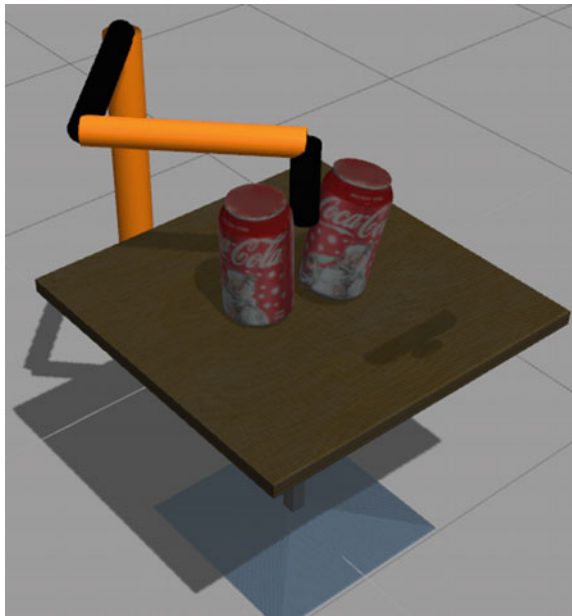
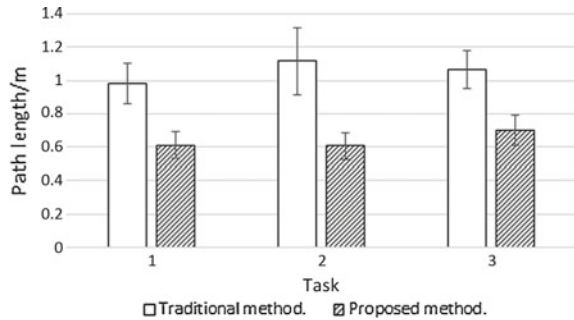


Fig. 11 Statistical result of experiment



robot controlled by traditional obstacle avoidance has to turn around the two cans, but the robot controlled by proposed method can go through from the middle of two cans.

Figure 10 shows that the robot is pushing a obstacle. The obstacle is tilted but not pushed over. It is shown the proposed controller can control the robot to push the obstacle appropriately in order to guarantee the safety.

Figure 11 shows the statistical data of experiment. The path length is the moving distance of end effector of the robot. The histogram and error bars are average and standard deviation of ten times of experiments. It shows that the path length of proposed method is shorter than the traditional one.

6 Conclusion

This paper proposed a new shared control method enable the robot pushes the obstacle while it moves to the target. An enhanced artificial potential field obstacle avoidance method is proposed, and a potential field for guaranteeing the safety of pushing is defined. The simulation shows under the control of the proposed method, the robot can push the obstacle safely and the path length is shorter than traditional method.

Acknowledgements This work is partially supported by the China Domestic Research Project for the International Thermonuclear Experimental Reactor (ITER) under Grant 2012GB102001, the Natural Science Foundation of China under Grants 61175088, 61221003 and 61573243.

References

1. Li, Q., Chen, W., Wang, J.: Dynamic shared control for human-wheelchair cooperation. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 4278–4283. IEEE (2011)
2. Wei, Z., Chen, W., Wang, J.: 3d semantic map-based shared control for smart wheelchair. In: Intelligent Robotics and Applications, pp. 41–51. Springer (2012)

3. Stilman, M., Kuffner, J.: Planning among movable obstacles with artificial constraints. *Int. J. Robot. Res.* **27**(11–12), 1295–1307 (2008)
4. Stilman, M., Schamburek, J.-U., Kuffner, J., Asfour, T.: Manipulation planning among movable obstacles. In: 2007 IEEE International Conference on Robotics and Automation, April 2007, pp. 3327–3332 (2007)
5. Miyazawa, K., Maeda, Y., Arai, T.: Planning of graspless manipulation based on rapidly-exploring random trees. In: The 6th IEEE International Symposium on Assembly and Task Planning: From Nano to Macro Assembly and Manufacturing (ISATP 2005), July 2005, pp. 7–12 (2005)
6. Makita, S., Maeda, Y.: Joint torque optimization for quasi-static graspless manipulation. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), May 2013, pp. 3730–3735 (2013)
7. Wei, Z., Chen, W., Wang, J.: Semantic mapping for smart wheelchairs using rgb-d camera. *J. Med. Imaging Health Inf.* **3**(1), 94–100 (2013)
8. Aiyama, Y., Inaba, M., Inoue, H.: Pivoting: A new method of graspless manipulation of object by robot fingers. In: Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems '93, IROS '93., Jul 1993, vol. 1, pp. 136–143 (1993)
9. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA workshop on open source software, vol. 3, no. 3.2, p. 5 (2009)
10. Munaro, M., Rusu, R.B., Menegatti, E.: 3d robot perception with point cloud library. In: Robotics and Autonomous Systems (2016). <http://www.sciencedirect.com/science/article/pii/S0921889015003176>

An RGB-D Visual Application for Error Detection in Robot Grasping Tasks

Ester Martinez-Martin, David Fischinger, Markus Vincze
and Angel P. del Pobil

Abstract The ability to grasp is a fundamental requirement for service robots in order to perform meaningful tasks in ordinary environments. However, its robustness can be compromised by the inaccuracy (or lack) of tactile and proprioceptive sensing, especially in the presence of unforeseen slippage. As a solution, vision can be instrumental in detecting grasp errors. In this paper, we present an RGB-D visual application for discerning the success or failure in robot grasping of unknown objects, when a poor proprioceptive information and/or a deformable gripper without tactile information is used. The proposed application is divided into two stages: the visual gripper detection and recognition, and the grasping assessment (i.e. checking whether a grasping error has occurred). For that, three different visual cues are combined: colour, depth and edges. This development is supported by the experimental results on the Hobbit robot which is provided with an elastically deformable gripper.

Keywords Service robotics · Grasping · Computer vision

E. Martinez-Martin (✉) · A.P. del Pobil
Robotic Intelligence Lab, Universitat Jaume-I (UJI),
Avda. Sos Baynat s/n, 12071 Castellón, Spain
e-mail: emartine@uji.es

A.P. del Pobil
e-mail: pobil@uji.es

D. Fischinger · M. Vincze
Department of Electrical Engineering, Institut für Automatisierungs-
und Regelungstechnik (ACIN), Vienna University of Technology (TU Wien),
Gusshausstrasse 27-29, 1040 Vienna, Austria
e-mail: fischinger@acin.tuwien.ac.at

M. Vincze
e-mail: Vincze@acin.tuwien.ac.at

1 Introduction

In the last decades, Robotics research has focused on the development of autonomous systems able to perform different services useful to the well-being of humans such as domestic tasks, healthcare services, entertainment, and education. In such services, object grasping and manipulation are fundamental motor skills since they allow humanoid robots to interact with their environment and properly perform those meaningful, physical tasks.

From a biological point of view, object grasping and manipulation is critically dependent on the senses of touch and proprioception because they provide information about the target object (e.g. size, shape, and texture) and indicate when it slides [1]. On the one hand, tactile data allows the system to easily perceive the contact interactions of the gripper, such as the slip of the held object [2]. For that reason, a wide variety of tactile sensors for robot hands have been developed [3]. Nevertheless, despite being a very active field, still tactile sensors present a number of problems. First, most such sensors are too bulky to be used without sacrificing the system dexterity. Another reason is because they are too expensive, slow, fragile or widely unavailable. They may also lack elasticity, mechanical flexibility or robustness. Some other problems relevant to tactile sensing in a general mechatronic systems are discussed in [4, 5]. This issue was apparent in the recent Amazon Picking Challenge held at ICRA 2015, in which many of the participants were apparently not using tactile sensing or any alternative approach, resulting in absurd behaviours as a consequence of not detecting grasp errors. It is therefore important to count with an alternative or complementary sensing approach to robustly detect errors in object grasping. Such a combination has been successfully applied in a framework for robot physical interaction [6].

On the other hand, proprioceptive data such as joint angles could be considered. In this approach, the location and geometry of the gripper is inferred from the kinematic information provided by sensing the joint angles and the known kinematics of the robot hand [7, 8]. In this way, the system is able to carry out a set out actions resulting in a target configuration that leads to a successful grasp (e.g. [9, 10]). However, an accurate kinematic model of a robot hand with several degrees of freedom can be difficult to obtain due to aspects such as elasticity or deformability that are hard to model, or changes that might occur over time such as an unalignment of a joint rotation axis. In other cases, the information provided is limited (e.g. open/close instead of the exact grip aperture). These drawbacks limit the suitability of this approach for service robots.

As an alternative, we propose here the use of vision to estimate the robot hand configuration when the joints do not provide any feedback, or it is not accurate enough. However, despite the fact that vision is present in most robot systems and used for object recognition and pose estimation for grasping ([11–14]), very few approaches have considered the visual input for tracking the robot hand. For instance, Sorribes et al. [15] presented a model-based approach to visually track the hand pose for grasping based on *virtual visual servoing*. Although the results are very promising,

it presents some drawbacks that considerably limit the system in terms of the hand visual features, the initialization of the tracking method, the speed of the hand, or the availability of a precise 3D model. Sanchez-Lopez et al. [16] proposed a three-stage method for visually tracking a 7-DoF end-effector under a binocular stand-alone configuration (note that only monocular data is used). In this case, a planar patch is placed at the end-effector and this patch is tracked over time. Experimental results highlighted that visual detection fails when the end-effector is too far from the camera or under certain illumination conditions.

As already mentioned, sensor integration is a promising path [6]. Li et al. [17] use vision to extract the object pose, contact positions are obtained from proprioceptive feedback and forward kinematics, whereas tactile sensing estimates contact forces. They use a BCH-code-based marker attached to the object to easily estimate its 6D pose, which limits the approach. Also, the accuracy of touch and proprioceptive data are a crucial issue that suggests the role of vision should be enhanced.

Discerning the success or failure of a grasping action is of paramount importance since undetected errors resulting in absurd behaviours are costly, and unacceptable for the user. Given the mentioned limitations of current approaches based on tactile sensing or proprioception, we propose an RGB-D visual application for identifying the success/failure condition of a robot grasp. For that, during the grasp execution, the fingers are visually monitored to check if an object is held or not. This problem is particularly challenging due to a number of issues: the high-dimensional configuration of current robot hands, the change in the gripper appearance when performing the grasping task, the wide range of graspable objects (and their features) in the real world that could be possibly unseen beforehand, noise in the visual input, different illumination conditions, heavily cluttered daily scenarios, variability in the grasping pose, occlusions, low visibility conditions, real-time constraints, etc. Thus, to overcome all these issues, the implemented visual application combines depth cues, edge detection and colour cues. First, the robot hand is segmented by means of the edge information together with a colour cue processing, and then, its contour sets the framework for determining if an object is being held by the fingers or not. In this case, the available edge map is properly combined with depth information in order to obtain a robust evaluation of the performance of the grasping task. The proposed approach has been validated by the Hobbit robot, a system based on the FESTO *Fin Ray Effect* [18], which elastically deforms and adapts its shape to the grasped object without any additional actuation. Although this feature makes the robot more versatile, life-like, and compatible with human interaction, the complexity of its visual detection is considerably increased.

This paper is organized as follows. Section 2 gives an overview of the visual application. Then, Sects. 3 and 4 describe in detail the new developed approach for identifying errors in robot grasping. Section 5 presents and analyses the experimental results carried out for evaluating the performance of the system.

2 Application Architecture

A key point to visually evaluate the success in the grasping task is to detect the robot hand and, in particular, its fingers. The reason lies in the fact that the robot fingers determine the condition for an object for being held or not in the sense that an object is grasped if it is located between the robot fingers. So, from this starting point, the designed visual application is divided into two stages (Fig. 1):

- *Gripper Recognition.* This stage is in charge of detecting and recognizing the robot gripper (and its fingers) from the visual input. For that, the colour cue is used for distinguishing among the objects that could potentially be the robot gripper and those belonging to the scene background. Then, an edge detection technique based on depth information allows the robot to recover the gripper shape
- *Grasping Check.* Its goal is to check if the gripper is holding an object after performing the grasping task or, on the contrary, if there was an error during its execution (e.g. the object slipped, the grasp planning was inaccurate and the gripper was closed out of the object, etc.). To do that, depth data is combined with edge information for detecting the contact points with the gripper fingers

Note that, during the application execution, two control statements have been introduced with the purpose of both obtaining more robust results and avoiding running unnecessary instructions. In this way the execution time is reduced (a crucial aspect for Robotics). Thus, the first check corresponds to the gripper status, provided by the proprioceptive sense, such that only in the case that the gripper is closed, the visual application is performed. Similarly, an error is returned when the gripper is not found in the visual scene.

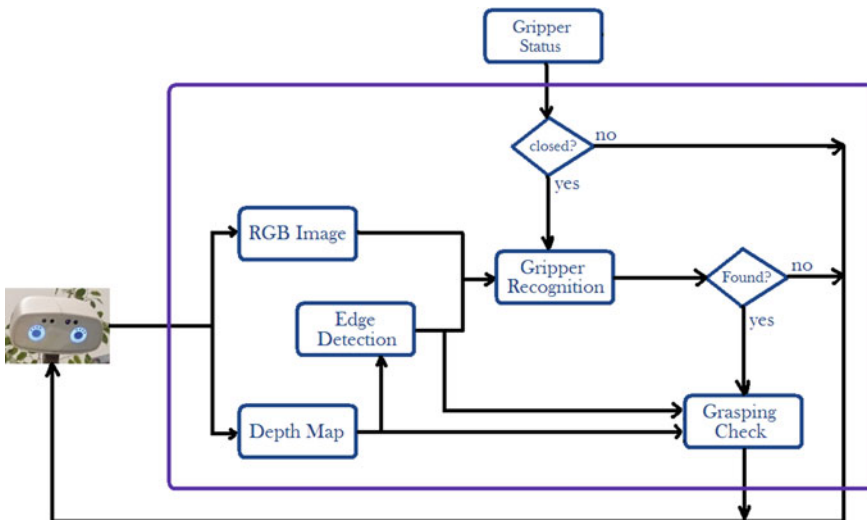


Fig. 1 Flowchart of the implemented RGB-D visual application

3 Gripper Recognition

The first issue to be solved refers to gripper detection and recognition. A straightforward method would be to use a 3D model of the gripper in order to recognize it by using any of the existing techniques (e.g. [19, 20]). However, the gripper elastic deformations for adapting to the shape of the object would make this kind of approaches miserably fail, since its appearance is continuously changing as shown in Fig. 2. Thus, with the aim to properly detect and recognize the robot gripper, different approaches were investigated.

A wide research on articulated, rigid, or even deformable object detection and recognition can be found in the literature. Model-based approaches are the most common way to recognize objects and shapes (e.g. [21, 22]). Basically, these techniques learn the object model in advance from a large set of training images which show the object from different viewpoints and in different poses. The features extracted from the objects in a scene are matched against features of the previously stored object models. Note that the chosen features must be invariant with respect to various transformations (e.g. changes in illumination, scale and/or view direction) and be also robustly extracted, what cannot be guaranteed in real scenarios and for all the existing robot grippers. In addition, despite being a good procedure for some kind of objects, it is difficult to learn models of objects with a high dimensionality and/or with a rich variability or deformability. Moreover, autonomy is a requirement for service robots and, consequently, no constraints about the gripper appearance can be established. Similarly, shape-based approaches were discarded since they depend on the gripper shape at each time and, consequently, suffer from the same drawbacks.

As an alternative, the Robotics community has widely used marker-based systems since they make feature extraction more robust (e.g. [23–25]). Although simple, relying on fiducial markers such as LEDs, Augmented Reality or coloured spheres tags has several disadvantages. First of all, the mobility of the robot hand is restricted to have at least one marker always in view. In addition, every trackable joint should contain at least one precisely located marker. For these reasons, marker-less approaches as those based on local features have been developed. In this case, objects are represented via their edges, colour or corner cues [26]; steerable filters [27]; haar-like features [28]; or scale-invariant descriptors (SURF [29], SIFT [30]). These approaches are very popular for their computational efficiency, simplicity and/or invariance to



Fig. 2 Appearance of the robot gripper in different poses

affine transformations. Nevertheless, their accuracy is closely linked to the considered number of features for describing an object. Furthermore, achieving reliable segmentation for robustly extracting object features is especially difficult with unconstrained environments. Moreover, object features are only relatively well behaved under small affine transformations, a condition that could not be always satisfied in the real-world.

Given that the gripper is moving during the execution of the grasping task, the motion cue could also be considered. This topic has been extensively studied because of its importance in natural and computer vision, as it is often required as an early step in a large variety of visual robotics applications. However, motion extraction from visual input is not straightforward because it could be due to different factors: movement of the imaged objects (targets and/or vacillating background elements), movement of the observer, motion of the light sources or a combination of them. In addition, noise in images can considerably affect the detection. So, despite research on this topic has taken a number of forms (see [31, 32] for a review), the images provided by our sensor—Microsoft Kinect RGB—are too noisy with segmentation thresholds that highly depend on the environmental conditions and, as a consequence, it is inappropriate for our goal.

Alternatively, taking into account that the robot vision system is capturing RGB-images along with per-pixel depth information with a reasonably accurate mid-resolution, the depth cue could be exploited. Nonetheless, the restricted range of depth estimation and the gripper features can lead to a *vanishing* gripper as depicted in Fig. 3, making this approach also inadequate for the task at hand.

Instead, colour could be a good cue since it provides powerful information for object detection and recognition. A simple and effective recognition scheme is representing and matching images on the basis of colour histograms. However, when the illumination circumstances are not the same, the accuracy in object recognition

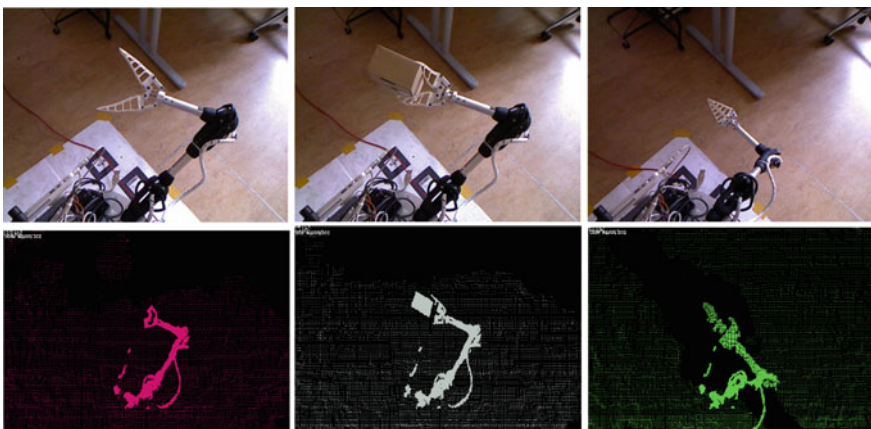


Fig. 3 Some examples where the gripper is visible in the RGB image (*top row*), but missing in the obtained depth maps (*bottom row*)

degrades significantly. Alternatively, colour image derivatives can be used. In this case, the channel derivatives are separately estimated and added to produce the final colour gradient. The main shortcoming of this approach is that derivatives of a colour edge might be in opposite directions for the separate colour channels. So, with the aim of achieving a robust gripper detection and recognition, different colour models have been investigated in terms of sensitivity to image parameters [33]. From this study, the $L^*a^*b^*$ colour space is the best alternative due to its invariance under different conditions. Nevertheless, this is a rather poor gripper description since the robot workspace could be full of objects with the same (or similar) colour as that of the gripper. Therefore, other cues must be used in conjunction to obtain satisfactory results.

Another approach is edge detection. Actually, edge detection is one of the primary low-level vision operations for object detection and recognition since it allows the system to extract geometric information from the visual input. Traditional approaches can be divided into Gradient- and Laplacian-based techniques. However, as stated in [34], Gradient-based algorithms have major drawbacks in their sensitivity to noise and also they cannot be adapted to a given image (their kernel filter dimensions and coefficients are static); while Laplacian-based approaches suffer from mismapping some of the lines. More recent approaches use a variety of features as input, including brightness, colour and texture gradients computed over multiple scales (e.g. [35, 36]). However, since visually salient edges correspond to a variety of visual phenomena, finding a unified approach to edge detection is difficult.

Motivated by all this research, we have developed a new approach for gripper detection and recognition that can be summarized as follows (see Fig. 4):

1. Image segmentation. This stage is composed of two concurrent processes:
 - Image segmentation based on colour cue. In this step, the capture RGB image is transformed into an *Lab* image and segmented by using its colour components. So, the segmentation thresholds have been experimentally set to $L \in [130, 140]$, $a \in [120, 150]$ and $b \geq 120$ (note that these values should be modified to properly detect a different gripper)
 - Image segmentation based on edge detection. This process uses the depth map estimated by the Microsoft Kinect camera and, from that, generates an edge map. That is, a pixel is classified as an edge if there is a leap between the

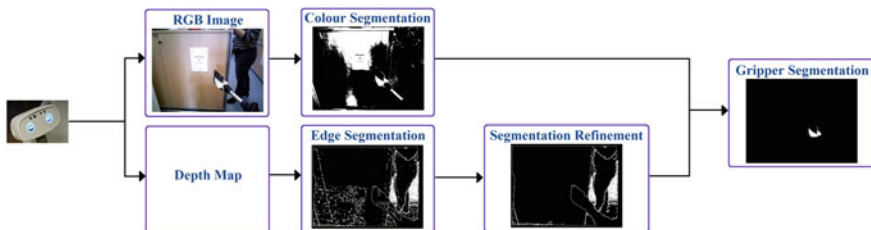


Fig. 4 Approach flowchart for robust robot gripper detection

depth information for that pixel and one of its neighbours. In our case, that jump was limited to 0.01 depth units. In addition, given the possibility of lack of depth information in some pixels (e.g. Fig. 3), border pixels in terms of presence/absence of information are also considered as edges

2. Segmentation refinement. Before combining the results from colour and edge segmentations, the robot architecture is used to refine the edge detection. Given that the vision system is always located at the top of the robot platform and looking ahead, the robot arm contour arises from the bottom part of the image. Thus, all the edges that are not connected in some way with the bottom image are discarded, as shown in Fig. 4
3. Gripper segmentation. The last step is to combine the two processed cues such that, firstly, the colour map acts as a mask for the refined edge detection, by isolating the gripper contour. Then, only the inner pixels labelled as foreground in the colour map are kept. In this way, the gripper fingers are robustly identified.

4 Grasp Check

Once the robot gripper has been detected in the image, the second stage corresponds to the grasp check itself. Considering that an object is grasped when the gripper contour is in touch with the object contour, the designed application searches for those contact points in order to determine whether the gripper is or not holding an object. For that, contour contact is determined by the contact points in the inner rectangle enclosing the gripper. However, there are some cases in which there are some configurations leading to a false contact detection, as shown in Fig. 5.

As a solution, we have integrated depth information. That is, from the position of the gripper in the 3D space, a 3D cube is defined and only the objects located in this cube are studied. Therefore, if there is a contact between the contours of the object



Fig. 5 Some cases in which colour-edge combination is not enough for properly evaluating the grasping task such that the blue line corresponds to the detected contact points

and the gripper fingers, then it is assumed that the object is held by the gripper and the grasping task was successfully performed. Otherwise, the grasping task failed at any point and the application notifies this situation.

5 Experimental Results

Our experimental platform is the *Hobbit* robot, a socially assistive robot aimed at helping seniors and elderly people at home. This robot is equipped with a multitude of sensors, providing the robot with perceptual data, and actuators, allowing the system to perform its tasks. The visual application presented in this paper has been tested by using the pan-tilt RGB-D camera (Microsoft Kinect) mounted in the *head* of the robot (approximately at a height of 130 cm) and the gripper which is based on the FESTO *Fin Ray Effect*. In this way, the system performance is evaluated in a challenging condition, while the gripper is continuously changing its appearance.

With the purpose of robustly analysing the application's performance, the robot was located at different scenarios and, at the same time, the pose of both the robot head and its gripper was continuously changed. Thus, with these changes in the configuration, performance was tested on a total of 940 images, with a resolution of 640×480 pixels, taken in five different scenarios, four robot head poses, three gripper poses and using ten different objects with distinct visual features (i.e. a toy car, a tissue box, keys, a biscuits box, a wallet, a pack of gum, a box of aspirin, a CD sleeve, a headphone's bag, and a box staple). Some of the obtained experimental results are depicted in Fig. 6. As it can be observed, the implemented application is able to overcome those situations in which the gripper is near the ground and its visibility is poor. In addition, the false contact positives when some objects are located below the closed gripper have been also properly dealt with, isolating the gripper from the remaining elements within the scene and robustly determining that no object is being grasped (i.e. a black image is obtained). As a particularity, note that the application only provides the robot with a partial image of the grasped object. The reason lies in the fact that the application is aimed at evaluating the grasping task based on the contour contact points. Consequently, only the object part in contact with the gripper is recovered. This makes the approach faster, running at a speed of 100 ms per image on the on-board PC. Thus, a real-time performance is obtained, which is crucial for service robots.

The performance analysis under different poses and environmental conditions (including cluttered scenes) has been carried out by means of a comparison between the system answer in terms of grasping or not an object, and the images manually labelled. Using this measurement, the application successfully evaluates the grasping task in 96 % of the studied images. However, thin objects or small ones like keys can make the system fail under certain configurations.

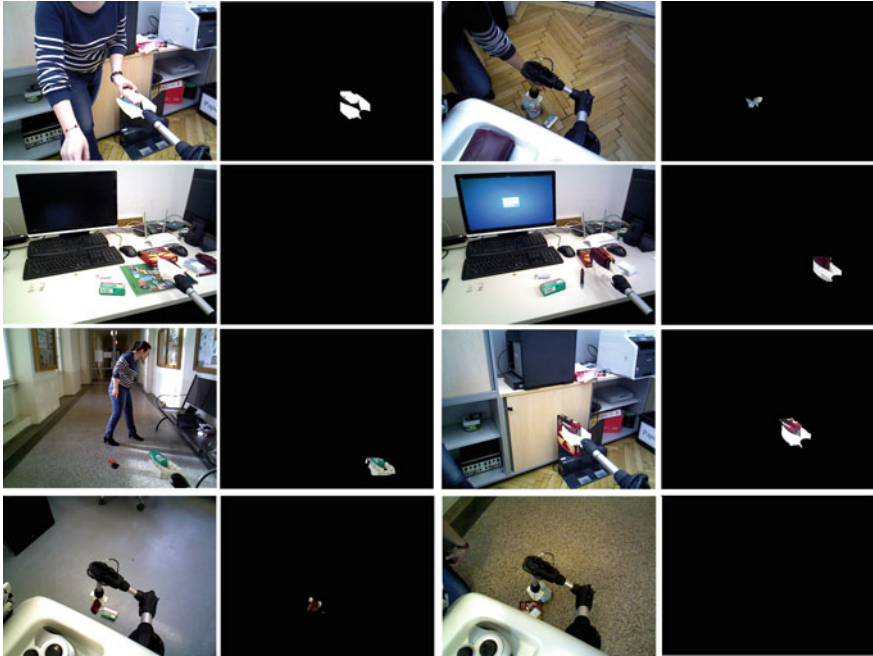


Fig. 6 Some of the experimental results obtained by the presented application on different scenarios, head poses and gripper poses

6 Conclusions

Grasping is a crucial issue for service robots. Indeed, the ability to perform autonomous dexterous operations is envisioned as an essential element for an effective physical interaction with the environment and humans. Discerning immediately the status of a grasping task, in terms of its success or failure, is of paramount importance since undetected errors result in absurd behaviours instead of recovery actions, which is unacceptable for the users. Despite the wide research on tactile and proprioceptive sensing, this is still a challenge in the robotics community. It is therefore crucial to count with an alternative or complementary sensing approach to robustly detect errors in object grasping. We present a visual application that uses only proprioceptive data in terms of gripper opened and closed. The method is divided into two different stages. The first one is in charge of detecting and recognizing the robot gripper from a visual input. For that, a combination of colour cue and edge detection based on depth information are used. This combination is required because each cue separately is not robust enough to achieve a satisfactory gripper recognition. Thus, after the gripper identification, the second stage evaluates the performance of the robot grasping task by means of depth information and edge detection based on that depth data.

This approach was implemented as a ROS service and was tested on a large number of images taken in different scenarios and changing the poses of both the robot head and gripper. Moreover, objects of ten different types were used. On this basis, the application was successful in 96 % of the cases, running at a speed of 100 ms per 640×480 image. The failure rate mainly results from some object features such as thinness or tiny size, which make the application misbehave. Consequently, as future work, we are planning to refine the proposed technique in order to properly deal with those features. With that aim, we are exploring other visual features that can improve the grasping task evaluation without constraining the system autonomy.

Acknowledgements This work has been partially funded by by Ministerio de Economía y Competitividad (DPI2015-69041-R), Generalitat Valenciana (PROMETEOII/2014/028), and by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 288146 (Hobbit).

References

1. Johansson, R.S., Flanagan, J.R.: Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nat. Rev. Neurosci.* **10**, 345–359 (2009)
2. Srinivasan, M.A., Whitehouse, J.M., LaMotte, R.H.: Tactile detection of slip: surface micro-geometry and peripheral neural codes. *J Neurophysiol.* **63**(6), 1323–1332 (1990)
3. Cutkosky, M.R., Howe, R.D., Provancher, W.R.: Force and tactile sensors. In: Siciliano, B., Khatib, O. (eds.) *Springer Handbook of Robotics*. Springer, New York (2008)
4. Dahiya, R.S., Metta, G., Valle, M., Sandini, G.: Tactile sensing—from humans to humanoids. *IEEE Trans. Robot.* **26**(1) (2010)
5. Lee, M.H.: Tactile sensing: new directions, new challenges. *Int. J. Robot. Res.* **19**(7), 636–643 (2000)
6. Prats, M., del Pobil, A.P., Sanz, P.J.: *Robot Physical Interaction Through the Combination of Vision, Tactile and Force Feedback*. Springer Tracts in Advanced Robotics, vol. 84(2013)
7. Haidacher, S., Hirzinger, G.: Contact Point Identification in Multi-Fingered Grasps Exploiting Kinematic Constraints, pp. 1597–1603. *ICRA* (2002)
8. Dupont, P., Schulteis, T., Millman, P., Howe, R.D.: Automatic identification of environment haptic properties. *Presence: Teleoperators Virtual Environ.* **8**(4), 392–409 (1999)
9. Deckers, P., Dollar, A.M., Howe, R.D.: Guiding grasping with proprioception and Markov models. In: *Workshop on Robot Manipulation: Sensing and Adapting to the Real World* (2007)
10. Johnsson, M., Balkenius, C.: Experiments with proprioception in a self-organizing system for haptic perception. *TAROS* 239–245 (2007)
11. Wang, W., Li, S., Chen, L., Chen, D., Kühnlenz, K.: Fast Object Recognition and 6D Pose Estimation using Viewpoint Oriented Color-Shape Histogram. *IICME* (2013)
12. Uckermann, A., Haschke, R., Ritter, H.: Real-time 3D segmentation of cluttered scenes for robot grasping. *Humanoids* 198–203 (2012)
13. Rusu, R.B., Bradski, G., Thibaux, R., Hsu, J.: Fast 3D recognition and pose using the viewpoint feature histogram. *IROS* 2155–2162 (2010)
14. Kuehnle, J., Verl, A., Zhixing, X., Ruehl, S., Zoellner, J.M., Dillmann, R., Grundmann, T., Eidenberger, R., Zoellner, R.D.: 6D Object localization and obstacle detection for collision-free manipulation with a mobile service robot. *ICAR* 1–6 (2009)
15. Sorribes, J.J., Prats, M., Morales, A.: Visual tracking of a jaw gripper based on articulated 3D models for grasping. *ICRA* 2302–2307 (2010)

16. Sanchez-Lopez, J.R., Marin-Hernandez, A., Palacios-Hernandez, E.R.: Visual Detection, Tracking and Pose Estimation of a Robotic Arm End Effector, pp. 41–48. ROSSUM (2011)
17. Li, Q., Elbrechter, C., Haschke, R., Ritter, H.: Integrating vision, haptics and proprioception into a feedback controller for in-hand manipulation of unknown objects. IROS 2466–2471 (2013)
18. Festo Fin Ray.: http://www.festo.com/net/SupportPortal/Files/53886/AdaptiveGripper_DHDG_en.pdf
19. Wohlkinger, W., Aldoma, A., Rusu, R.B., Vincze, M.: 3DNet: largescale object class recognition from CAD models. ICRA 5384–5391 (2012)
20. Daras, P., Axenopoulos, A.: A 3D shape retrieval framework supporting multimodal queries. *Int. J. Comput. Vis.* **89**, 229–247 (2010)
21. Lee, S., Lee, J., Moon, D., Kim, E., Seo, J.: Robust recognition and pose estimation of 3D objects based on evidence fusion in a sequence of images. ICRA 3773–3779 (2007)
22. Sian, N., Sakaguchi, T., Yokoi, K., Kawai, Y., Maruyama, K.: Operating humanoid robots in human environments. In: RSS Workshop: Manipulation for Human Environments (2006)
23. Wuthrich, M., Pastor, P., Kalakrishnan, M., Bohg, J., Schaal, S.: Probabilistic object tracking using a depth camera. IROS (2013)
24. Gratal, X., Bohg, J., Bjrkmann, M., Kragic, D.: Scene representation and object grasping using active vision. In: IROS Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics (2010)
25. Vahrenkamp, N., Wieland, S., Azad, P., Gonzalez, D., Asfour, T., Dillmann, R.: Visual servoing for humanoid grasping and manipulation tasks. *Humanoids* 406–412 (2008)
26. Urdiales, C., Dominguez, M., de Trazegnies, C., Sandoval, F.: A new pyramid-based color image representation for visual localization. *Image Vis. Comput.* **28**(1), 78–91 (2010)
27. Villamizar, M., Sanfeliu, A., Andrade-Cetto, J.: Computation of rotation local invariant features using the integral image for real time object detection. ICPR **4**, 81–85 (2006)
28. Wilson, P., Fernandez, J.: Facial feature detection using haar classifiers. *J Comput. Sci. Coll.* **21**(4), 127–133 (2006)
29. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Surf: speeded up robust features. *Comput. Vis. Image Underst.* **110**(3), 346–359 (2008)
30. Lowe, D.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
31. Martínez-Martín, E., del Pobil, A.P.: *Robust Motion Detection in Real-Life Scenarios*. Springer (2012)
32. Brutzer, S., Hoferlin, B., Heideman, G.: Evaluation of background subtraction techniques for video surveillance. CVPR 1937–1944 (2011)
33. Martínez-Martín, E., del Pobil, A.P.: Visual object recognition for robot tasks in real-life scenarios. URAI 644–651 (2013)
34. Shrivakshan, G.T., Chandrasekar, C.: A comparison of various edge detection techniques used in image processing. *Int. J. Comput. Sci.* 9(5), no. 1, 269–276 (2012)
35. Lim, J., Zitnick, C.L., Dollar, P.: Sketch tokens: a learned mid-level representation for contour and object detection. CVPR 3158–3165 (2013)
36. Arbelæz, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(5), 898–916 (2011)

Sensorless In-Hand Caging Manipulation

Yusuke Maeda and Tomohiro Asamura

Abstract In this paper, we study a method to manipulate objects in position-controlled robot hands: in-hand caging manipulation. In this method, an object is caged by a hand throughout manipulation and located around a goal as a result of the deformation of the cage without sensing the object configuration. A motion planning algorithm for planar in-hand caging manipulation of a circular object is proposed. Planned motions by the algorithm are successfully performed by a two-fingered robot hand. Additionally some examples of 3D in-hand caging manipulation by a four-fingered robot hand are presented.

Keywords Caging · Robot hand · Manipulation planning

1 Introduction

In-hand manipulation by robot hands has been studied widely aiming at human-like dexterity. Most of such studies adopted force sensing and/or force control to achieve in-hand manipulation (e.g., [1] and [2]).

In contrast, caging by robot hands [3] is a method for robotic manipulation without force sensing or force control. In robotic caging, an object is captured geometrically by the cage that is formed by a robot hand. Once the object is caged, it can be manipulated by the position control of the hand because it cannot escape from the cage.

The authors applied the caging method to robotic in-hand manipulation, which enabled position-control-based in-hand manipulation [4]. In our method, in-hand manipulation of a caged object is achieved by changing the shape of the cage. We

Y. Maeda (✉)

Faculty of Engineering, Yokohama National University,
79-5 Tokiwadai, Hodogaya-ku, Yokohama 240-8501, Japan
e-mail: maeda@ynu.ac.jp

T. Asamura

Mitsubishi Electric Corp., Tokyo, Japan

refer to this approach as “in-hand caging manipulation.” In-hand caging manipulation is a kind of sensorless manipulation [5] because it can be performed without sensing the configuration of the object. The sensorless nature and maintenance of caging throughout manipulation make in-hand caging manipulation robust to disturbances.

Our in-hand caging manipulation can be executed based on shape information of objects and position control. In other words, it does not explicitly use force sensing, force control, or mechanical information such as friction. The geometrical nature of in-hand caging manipulation may impose restriction on applicable manipulation, but it suits for today’s robots and would provide an alternative for robotic manipulation.

This paper explains in-hand caging manipulation in detail, while only a very brief overview was given in [4]. First, we formulate our in-hand caging manipulation. Then we propose an algorithm to plan 2D in-hand caging manipulation by a two-fingered robot hand. Some experimental results of execution of planned in-hand manipulation are shown. Finally, we demonstrate some examples of 3D in-hand caging manipulation by a four-fingered hand.

2 Related Works

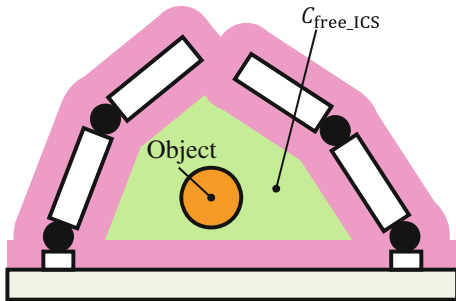
Most of the previous studies on robotic caging assume that the robot formation is fixed once caging is achieved. As one of a few exceptions, Sudsang et al. studied manipulation of polygonal objects by multiple circular robots [6]. In the manipulation, the robot formation was variable. Yamawaki and Yashima studied whole arm manipulation, where caging was used partially [7]. The authors’ group studied robotic caging with walls [8, 9]; objects are caged by not only mobile robots but also immobile walls.

3 Conditions for In-Hand Caging Manipulation

In [8] and [9], robotic caging in which an object is caged not only by robots but also by passive entities such as walls was investigated. Using passive walls can reduce the necessary number of robots for caging, but raises a manipulability problem; that is, the object is caged by mobile robots and immobile walls and therefore simple movements of the robots with a fixed formation may not be able to maintain caging and achieve desired manipulation.

In [8] and [9], we studied this problem and formulated conditions for caging and manipulability. In this section, we apply the conditions to in-hand caging manipulation, where mobile fingers and immobile palms exist instead of robots and walls.

Fig. 1 Caging by fingers and a palm



3.1 Conditions for Caging

Let us consider a robot hand and an object as shown in Fig. 1. We use the following notations similar to those used in [10]:

- C : The configuration space of the object.
- \mathcal{A}_{obj} : The volume occupied by the object in the workspace.
- \mathcal{A}_{fin} : The volume occupied by the fingers in the workspace.
- $\mathcal{A}_{\text{palm}}$: The volume occupied by the palm in the workspace.
- \mathbf{q}_{obj} : The configuration of the object. Typically, it is the position and orientation of the object.
- \mathbf{q}_{fin} : The configuration of the fingers. Typically, it is the joint angles of the fingers.

The free configuration space of the object in which the object is free from interference with the robot hand can be written as follows:

$$C_{\text{free}} := \{ \mathbf{q} \in C \mid \mathcal{A}_{\text{obj}}(\mathbf{q}_{\text{obj}}) \cap (\mathcal{A}_{\text{fin}}(\mathbf{q}_{\text{fin}}) \cup \mathcal{A}_{\text{palm}}) = \emptyset \}. \quad (1)$$

When the object is caged by the rigid parts of the object, C_{free} can be separated into two parts: an inescapable configuration space (ICS) [11] and an escapable configuration space (ECS). If the former space does not exist, the object cannot be caged. In the latter space, the object can escape in the distance. Here we define the former space $C_{\text{free_ICS}}$ and the latter space $C_{\text{free_ECS}}$ as follows:

$$C_{\text{free_ICS}} := C_{\text{free}} \setminus C_{\text{free_ECS}} \quad (2)$$

$$C_{\text{free_ECS}} := \bigcup_{\mathbf{q}_{\text{obj}} \in \mathcal{Q}_{\text{dist}}} C_{\text{free_max}}(\mathbf{q}_{\text{obj}}), \quad (3)$$

where $\mathcal{Q}_{\text{dist}} (\subset C)$ is a set of object configurations that can be regarded as “distant” or unquestionably escaped; $C_{\text{free_max}}(\mathbf{q}_{\text{obj}})$ is the maximal connected subset of C_{free} that includes \mathbf{q}_{obj} . Now the caging condition can be written as follows:

$$\begin{cases} C_{\text{free_ICS}} \neq \emptyset \\ \mathbf{q}_{\text{obj}} \in C_{\text{free_ICS}}. \end{cases} \quad (4)$$

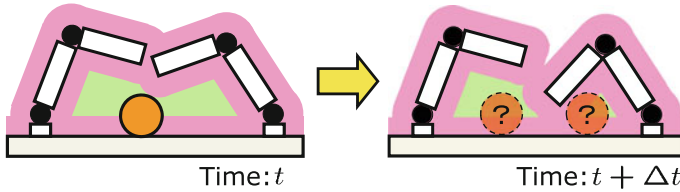


Fig. 2 Split of ICS

3.2 Caging Manipulability

Our previous papers [8, 9] studied the manipulability problem in caging by robots and walls. The derived condition for caging manipulation can be written as follows:

$$\lim_{\Delta t \rightarrow +0} (C_{\text{free_ICS}}(t) \cap C_{\text{free_ICS}}(t + \Delta t)) = C_{\text{free_ICS}}(t), \quad (5)$$

where $C_{\text{free_ICS}}(t)$ denotes $C_{\text{free_ICS}}$ at time t . The above condition states that the ICS neither shrinks discontinuously nor splits into two or more distinct subsets (Fig. 2). Such discontinuous shrinkage or split can occur even when robots move continuously, which may cause manipulation failure. Note that:

- (5) holds in conventional caging where robot formation is fixed;
- discontinuous expansion of the ICS is allowed.

In fact, this formulation does not directly deal with the manipulability of the object because it is difficult for sensorless manipulation. Instead, the changeability of the ICS, which indirectly corresponds to the manipulability of the object, is formulated.

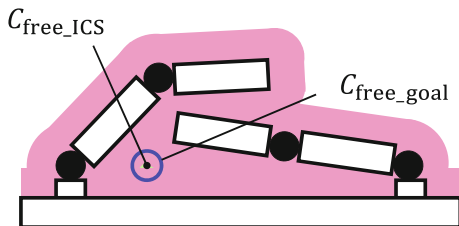
The analysis above is also applicable to in-hand caging manipulation. As a result, if (4) and (5) hold, in-hand caging manipulation is possible based on position control of robot fingers. Sensing of the configuration of the object is not necessary.

4 Planning of In-Hand Caging Manipulation

4.1 Problem Statement

Here we discuss how to plan in-hand caging manipulation based on the conditions presented in the previous section. The planning problem is described as follows:

- Input: the initial finger configuration ($\mathbf{q}_{\text{fin}0}$) and the goal region for the object ($C_{\text{free_goal}}$)
- Output: a path of the finger configuration (\mathbf{q}_{fin}) to manipulate the object to $C_{\text{free_goal}}$.

Fig. 3 Goal condition

We assume that the object is caged initially and the ICS in which the object exists is known, but the exact configuration of the object is unknown. Note that we can calculate the ICS even if we cannot localize the object.

The goal condition can be written as follows:

$$C_{\text{free_ICS}} \subseteq C_{\text{free_goal}}. \quad (6)$$

Typically, $C_{\text{free_goal}}$ is given as a small region (Fig. 3). If we define $C_{\text{free_goal}}$ as a single configuration, the object would be in form closure in the goal.

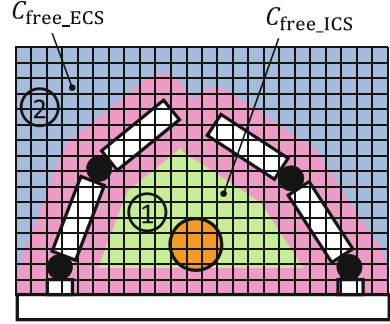
4.2 Planning Algorithm

We developed a planning algorithm for planar in-hand caging manipulation of circular objects by a two-fingered hand. Each finger of the hand has two joints. As a result, the configuration space of the object is two-dimensional and that of the hand is four-dimensional.

The planning algorithm is based on RRT (Rapidly-exploring Random Trees) [12] as follows:

1. Add the initial configuration of the fingers $\mathbf{q}_{\text{fin}0}$ to the RRT.
2. Sample a configuration of the fingers $\mathbf{q}_{\text{fin_sample}}$ randomly and find its nearest neighbor in the RRT: $\mathbf{q}_{\text{fin_nearest}}$.
3. Consider a branch of length Δl from $\mathbf{q}_{\text{fin_nearest}}$ to $\mathbf{q}_{\text{fin_new}}$ in the direction of $\mathbf{q}_{\text{fin_sample}}$.
4. If $\mathbf{q}_{\text{fin_new}}$ is not collision-free, discard it and go back to step 2.
5. If $\mathbf{q}_{\text{fin_new}}$ does not pass the caging test (4) and the manipulability test (5), discard it and go back to step 2.
6. Add the branch from $\mathbf{q}_{\text{fin_nearest}}$ to $\mathbf{q}_{\text{fin_new}}$ to the RRT.
7. Continue extending the branch of length Δl in the direction of $\mathbf{q}_{\text{fin_sample}}$, as far as (4) and (5) hold.
8. Repeat steps 2 through 7 until $\mathbf{q}_{\text{fin_new}}$ satisfies the goal condition (6).

Fig. 4 Labeling process



4.3 Tests for In-Hand Caging Manipulation

In order to test the conditions for in-hand caging manipulation ((4) and (5)), we adopted grid-based representation of the configuration space of the object; possible locations of the center of the object are discretized as grid points on a plane. Collision-free grid points denote C_{free} approximately. In our current implementation for planar manipulation of circular objects, 400×400 grid points are used for their 2D configuration space.

C_{free} may consist of multiple distinct regions and the object should be in one of them. For sensorless manipulation, we perform labeling to find where the object is (Fig. 4). $C_{\text{free_ICS}}(t + \Delta t)$ ($t \geq 0$) can be identified as the connected region that has an intersection with $C_{\text{free_ICS}}(t)$. If $C_{\text{free_ICS}}(t) \cap Q_{\text{dist}} = \emptyset$, the object is caged. In this case, Q_{dist} is a set of the boundary grids.

As for caging manipulability test, discontinuous shrinkage of the ICS rarely occurs in this problem. Thus we focused on split of the ICS (Fig. 2). If there are two or more distinct ICS regions that have an intersection with $C_{\text{free_ICS}}(t)$ at time $t + \Delta t$, we judge that caging manipulation is impossible due to split of the ICS.

4.4 Heuristics to Avoid Jamming

The manipulation plans generated by the above algorithm often failed in execution due to jamming [6]. Jamming is a mechanical phenomenon and therefore it is difficult to eliminate it completely in our geometry-based sensorless scheme. However, to avoid it as possible, the following heuristics was used in planning:

$$\begin{cases} w(t + \Delta t)/w(t) > b \\ h(t + \Delta t)/h(t) > b, \end{cases} \quad (7)$$

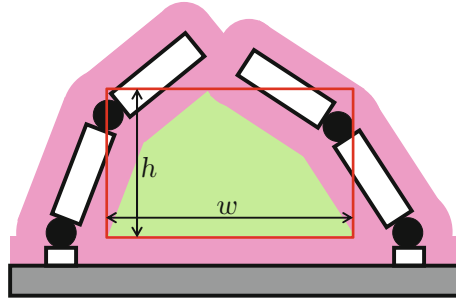


Fig. 5 Bounding box of ICS

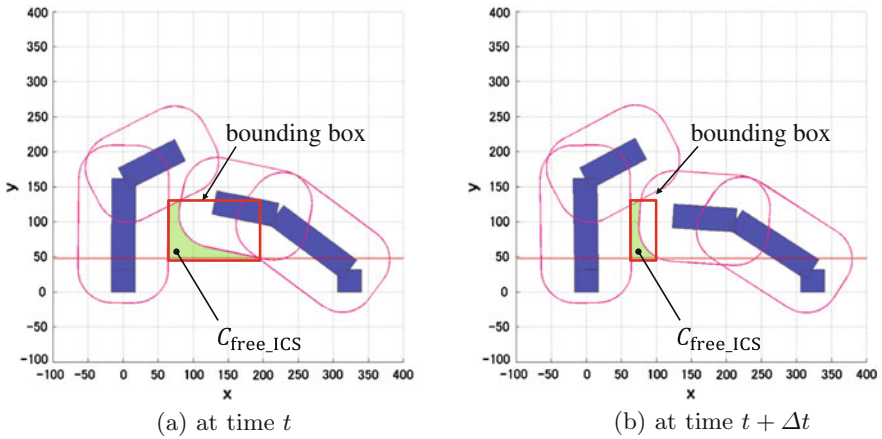


Fig. 6 A case in which jamming often occurs

where w and h is the width and the height of the bounding box of C_{free_ICS} , respectively (Fig. 5); b is a threshold and 0.5 in our current implementation. If (7) does not hold, C_{free_ICS} shrinks rapidly from time t to time $t + \Delta t$, as in the case of Fig. 6. In such cases, jamming often occurs. Thus we extend branches of the RRT only when (7) also holds.

4.5 Execution of Manipulation Plans

Here we show some examples of execution of planned in-hand caging manipulation. We used an experimental setup shown in Fig. 7. The object is a circular wood plate. The robot hand has four servomotors (Futaba RS405CB) to drive its joints. The links of the hand are made of acrylic resin and their inner sides are covered with low-friction fluoroplastic tapes. The hand was placed on a horizontal plane and therefore gravity effect was negligible.

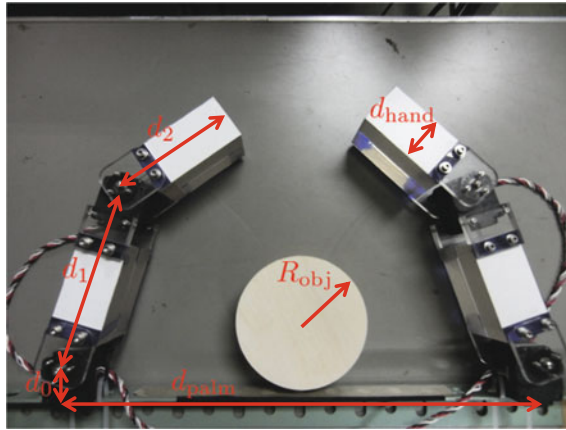


Fig. 7 A two-fingered hand

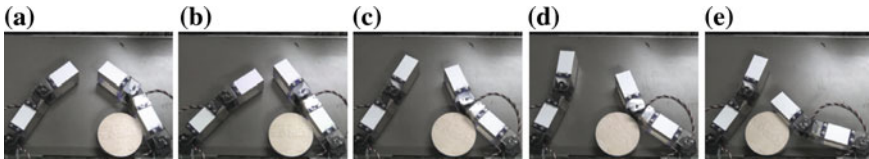


Fig. 8 In-hand caging manipulation (*right-to-left*)

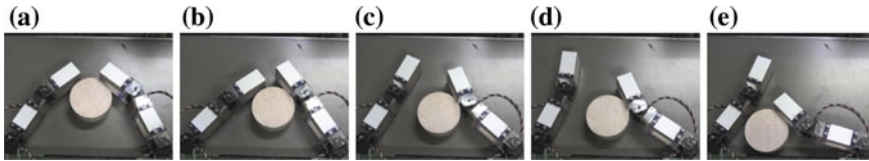


Fig. 9 In-hand caging manipulation (*top-to-left*)

Figures 8 and 9 show carrying the object to the bottom-left corner of the hand. The goal region was identical in the two cases, but the initial configuration of the object was different. In both cases, according to a single motion plan, the object was moved to the goal region successfully regardless of the difference in the initial object configuration, without sensing the object configuration. The time required for planning was 2760 [CPU s] on a PC with Core i7 3930K CPU.

Figures 10 and 11 also show planned in-hand caging manipulation. The object was carried to the center of the hand regardless of the difference in the initial object configuration in a sensorless manner. The time required for planning was 10856 [CPU s].

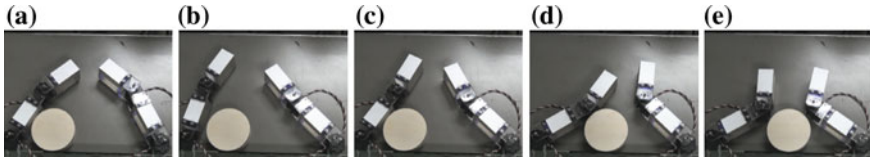


Fig. 10 In-hand caging manipulation (*left-to-center*)

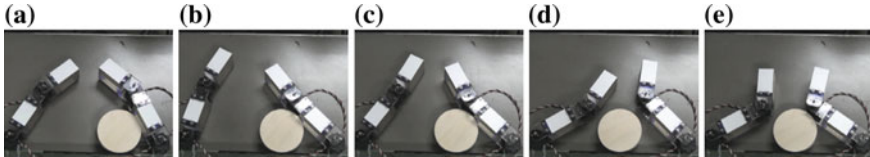


Fig. 11 In-hand caging manipulation (*right-to-center*)

The heuristics to avoid jamming (Sect. 4.4) worked well in many cases, but not always. The failure highly depends on the initial configuration of the object and therefore it is difficult to analyze the failure rate. How to escape from jamming during manipulation should be tackled in future work.

5 3D In-Hand Caging Manipulation

5.1 Overview of 3D In-Hand Caging Manipulation

We also conducted some experiments of in-hand caging manipulation in 3D space to test its feasibility. Because the planning algorithm for 3D in-hand caging manipulation is under development, we executed manipulation plans that were generated manually.

5.2 Experimental Setup

We prepared an experimental setup shown in Fig. 12. A six-degree-of-freedom industrial manipulator (Fanuc LR Mate 200i) with a four-fingered hand was used. Each of the fingers has three joints and attached to the planar palm with circular symmetry. Each joint is driven by a servomotor (Futaba RS405CB).

A spherical object with a 40 (mm) radius (Fig. 13) and a cylindrical object with a 47.5 (mm) radius and a 40 (mm) height (Fig. 14) were used in the experiments. These have circular sections and therefore it is relatively straightforward to apply the results of 2D in-hand caging manipulation to 3D cases.

Fig. 12 A four-fingered hand

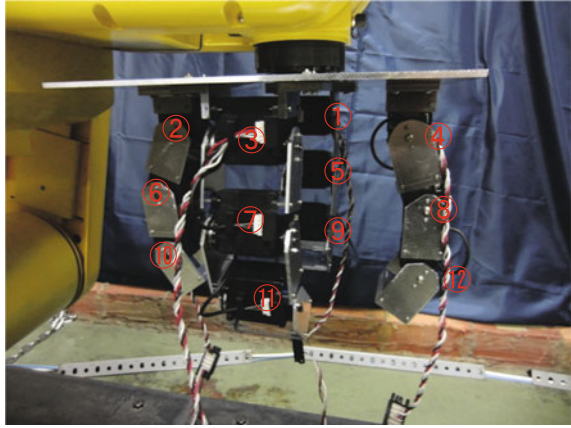


Fig. 13 Sphere

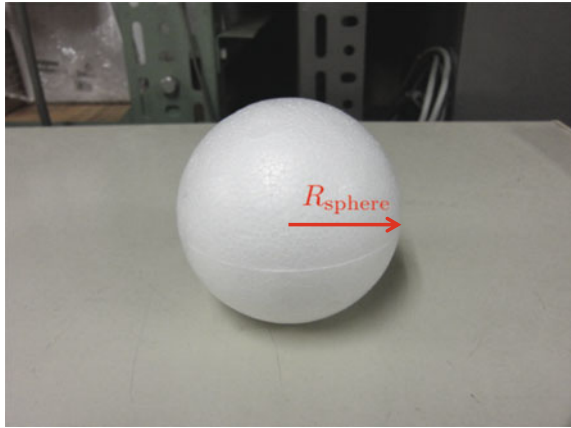


Fig. 14 Cylinder



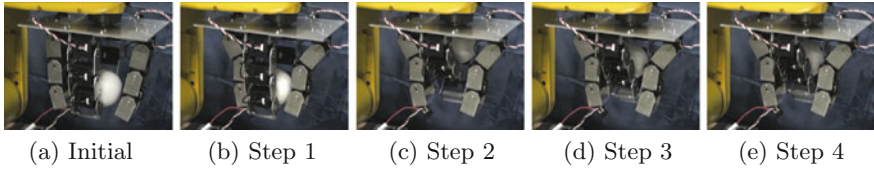


Fig. 15 In-hand manipulation of a sphere (to rear side)

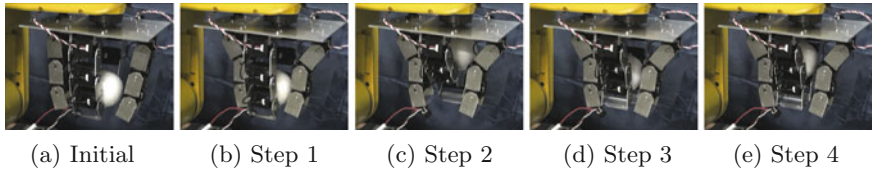


Fig. 16 In-hand manipulation of a sphere (to center)

Table 1 Joint angles [deg] (to rear side)

	Initial	Step 1	Step 2	Step 3	Step 4
θ_1	15	←	←	←	←
θ_2	10	30	←	←	←
θ_3	15	←	50	←	60
θ_4	10	30	←	←	←
θ_5	10	←	←	←	←
θ_6	0	-35	←	←	←
θ_7	10	←	←	-10	←
θ_8	0	-35	←	←	←
θ_9	20	←	←	←	←
θ_{10}	15	←	←	←	←
θ_{11}	20	←	←	-10	←
θ_{12}	15	←	←	←	←

5.3 Experimental Results

Scenes in experiments of 3D in-hand caging manipulation of the sphere are shown in Figs. 15 and 16. The nominal joint angles in the experiments for each case is shown in Tables 1 and 2, respectively. In the case of Fig. 15, the spherical object was carried to the rear side in the hand. In the case of Fig. 16, the spherical object was carried to the palm-side center in the hand. As the fingers move, the ICS was narrowed gradually and manipulation was achieved in these cases.

Step 1 almost corresponds to making the ICS compact and Steps 2 through 4 correspond to moving the object to the goal region.

Table 2 Joint angles [deg] (to center)

	Initial	Step 1	Step 2	Step 3	Step 4
θ_1	15	←	←	←	50
θ_2	10	30	←	←	←
θ_3	15	←	50	←	←
θ_4	10	30	←	←	←
θ_5	10	←	←	←	-20
θ_6	0	-35	←	←	←
θ_7	10	←	←	-20	←
θ_8	0	-35	←	←	←
θ_9	20	←	←	←	-20
θ_{10}	15	←	←	←	←
θ_{11}	20	←	←	-20	←
θ_{12}	15	←	←	←	←

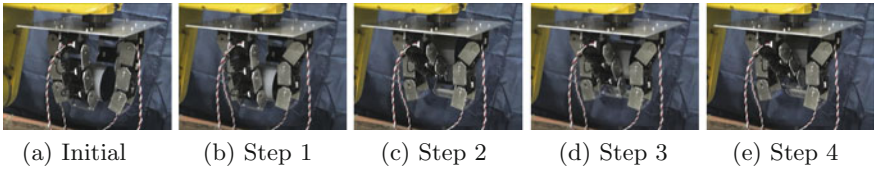


Fig. 17 In-hand manipulation of a cylinder (to rear side)

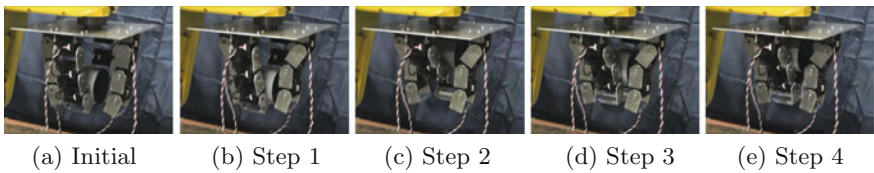


Fig. 18 In-hand manipulation of a cylinder (to center)

Similarly, scenes in experiments of 3D in-hand caging manipulation of the cylinder are shown in Figs. 17 and 18. The same nominal joint angles in the sphere experiments were used (Tables 1 and 2, respectively). The cylindrical object was manipulated similarly to the goal region.

6 Conclusion

In this paper, we formulated in-hand caging manipulation. Then we presented a manipulation planning algorithm for 2D in-hand caging manipulation of circular objects by a two-fingered hand. The planned manipulation was tested on an actual

robot hand. Additionally, we presented some results of 3D in-hand caging manipulation by an actual four-fingered hand.

Further work should include:

- effective jamming avoidance in planning and execution phases
- acceleration of motion planning
- wider variety of objects and robot hands [4].

Acknowledgements This work was supported by JSPS KAKENHI Grant Number JP22700200 and JKA's promotion funds from KEIRIN RACE (24-83).

References

1. Rus, D.: In-hand dexterous manipulation of piecewise-smooth 3-D objects. *Int. J. Robot. Res.* **18**(4), 355–391 (1999)
2. Kondo, M., Ueda, J., Ogasawara, T.: Recognition of in-hand manipulation using contact state transition for multifingered robot hand control. *Robot. Auton. Syst.* **56**(1), 66–81 (2008)
3. Rimon, E., Blake, A.: Caging planar bodies by one-parameter two-fingered gripping systems. *Int. J. Robot. Res.* **18**(3), 299–318 (1999)
4. Maeda, Y., Asamura, T., Egawa, T., Kurata, Y.: Geometry-based manipulation through robotic caging. In: *IEEE/RSJ IROS 2014 Workshop on Robot Manipulation: What has been Achieved and What Remains to be Done?* (2014)
5. Erdmann, M.A., Mason, M.T.: An exploration of sensorless manipulation. *IEEE J. Robot. Autom.* **4**(4), 369–379 (1988)
6. Sudsang, A., Rothganger, F., Ponce, J.: Motion planning for disc-shaped robots pushing a polygonal object in the plane. *IEEE Trans. Robot. Autom.* **18**(4), 550–562 (2002)
7. Yamawaki, T., Yashima, M.: Randomized planning and control strategy for whole-arm manipulation of a slippery polygonal object. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2485–2492 (2013)
8. Yokoi, R., Kobayashi, T., Maeda, Y.: 2D caging manipulation by robots and walls. In: *Proceedings of IEEE International Symposium on Assembly and Manufacturing*, pp. 16–21 (2009)
9. Yokoi, R., Maeda, Y., Kobayashi, T.: Formulation and planning of planar caging manipulation by robots and walls. *Trans. Jpn Soc. Mech. Eng. Series C* **76**(770), 2671–2677 (2010), (in Japanese)
10. Maeda, Y., Kodera, N., Egawa, T.: Caging-based grasping by a robot hand with rigid and soft parts. In: *Proceedings of 2012 IEEE International Conference on Robotics and Automation*, pp. 5150–5155 (2012)
11. Sudsang, A., Ponce, J., Srinivasa, N.: Grasping and in-hand manipulation: experiments with a reconfigurable gripper. *Adv. Robot.* **12**(5), 509–533 (1998)
12. LaValle, S.M., Kuffner, J.J.: Rapidly-exploring random trees: progress and prospects. In: Donald, B., Lynch, K., Rus, D. (eds.) *Algorithmic and Computational Robotics: New Directions*, pp. 293–308. A. K. Peters, Wellesley, U.S.A. (2001)

Development of New Cosmetic Gloves for Myoelectric Prosthetic Hand by Using Thermoplastic Styrene Elastomer

Yoshiko Yabuki, Kazumasa Tanahashi, Suguru Hoshikawa,
Tatsuhiko Nakamura, Ryu Kato, Yinlai Jiang and Hiroshi Yokoi

Abstract This paper reports on design and development of new cosmetic gloves for Myoelectric Prosthetic Hand which provides a realistic appearance and flexible motion of robot hands. The main design issues are divided into five as followings; appearance, gripping performance, durability, texture, flexibility. The appearance includes the shape, wrinkles, finger mark, nail, and color of the hand; the aim is to make these properties of the prosthetic hand as similar to those of the human hand as possible. The durability is evaluated by adaptabilities for daily living, and flexible materials without prevention from finger motion. Furthermore, the gripping performance is improved by the thickness map of palm which is well fit to the gripping object. The experiment shows the results of the performance test applied to the pick-and-place task by using powered prosthetic hand in order to evaluate total properties of the developed cosmetic gloves.

Keywords Cosmetic gloves • Myoelectric prosthetic hand • Gripping performance • Texture of the skin

Y. Yabuki (✉) · S. Hoshikawa · T. Nakamura · Y. Jiang · H. Yokoi
The University of Electro-Communications, Cho-fu, Tokyo, Japan
e-mail: yabuki@hi.mce.uec.ac.jp

S. Hoshikawa
e-mail: hoshikawa@hi.mce.uec.ac.jp

T. Nakamura
e-mail: tatsuhiko@hi.mce.uec.ac.jp

Y. Jiang
e-mail: jiang@hi.mce.uec.ac.jp

H. Yokoi
e-mail: yokoi@hi.mce.uec.ac.jp

K. Tanahashi
Tanac Co. Ltd, Gifu, Japan
e-mail: s-tanahashi@k-tanac.co.jp

R. Kato
Yokohama National University, Yokohama, Japan
e-mail: Kato-ryu-cy@ynu.ac.jp

1 Introduction

The electromyogram (EMG) prosthetic hand is one of the engineering technologies of powered artificial hands that is used for amputees; it is controlled using myoelectric signals (EMG) detected in the amputee's arm. The overall system of the EMG prosthetic consists of a powered hand, EMG controller, battery, below-elbow socket, and cosmetic glove. The development of an EMG prosthetic hand with five fingers has been highly successful, except for the problem of total weight and size. It covers almost all functions of the human hand, including finger motion and wrist motion [1]. However, the attachment part of the amputee arm needs to support all parts of the EMG prosthetic hand system by using a cylindrical socket; therefore, the system must be lightweight. Thus, the total weight must be less than 300 g. This weight condition restricts the number and weight of motors and batteries; therefore, the cosmetic glove should be designed to minimize motor power loss.

This paper reports the results of the research and development of a cosmetic glove based on a hyperelastic material. The desired functions of the cosmetic glove and experimental results for developed products are described. The products are designed based on existing research on prosthetics as well as new material. The design concept and requirements are described in Sect. 2, and the experimental results and evaluations are presented in Sects. 3 and 4, respectively.

2 Design Concept and Requirements

The history of the cosmetic glove started with the use of polyvinyl chloride (PVC) in the 1960s; a realistic appearance similar to a human hand was realized with 3D copy technology based on a mold. The 3D copy technology was developed to realize the natural form of fingers and the precise texture of skin [2]. The next step was to improve the colors of skin by using silicon-based material. To this end, prosthetists and orthoptists established painting techniques in order to mimic human skin colors accurately. However, even with significant developments in powered prosthetic hand technology, emulating finger motion has remained challenging. Thus, the technology is currently at the stage of recreating finger motion and handling functions [3].

This section presents the design requirements of the cosmetic glove and proposes a design method considering the following aspects: (2.1) appearance, (2.2) function of grasping and handling, (2.3) durability, (2.4) bio-texturized feeling, and (2.5) flexibility.

2.1 Appearance

The human hand consists of five fingers and the palm. The skin of the palm has thin tissue around the outer part and thick tissue including muscles in the inner part. The finger consists of three cylindrical sections of skin tissue that cover distal phalange, the intermediate phalanx, and the proximal phalanx. The distal phalanx of the fingertip includes a hemi-spherical fingertip with a stiff, thin nail, and it has a specific texture characterized by the fingerprint. Existing research and development have already realized a method of molding using die-casting material to mimic the natural shape and texture of a finger. Therefore, this study sets a design requirement of replicating finger texture with fingerprint and wrinkles on the finger joint, and for this purpose, we utilized a plaster-cast model of the human hand and obtained the precise texture of fingerprint and wrinkles by referring to existing research.

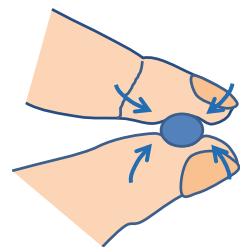
Three glove sizes have been developed: adult male, adult female, and child size. The child-size plaster cast model was developed with data from one hundred elementary-school children with the permission of the ethics committee of the University of Electro-Communication (permit No. 10006). The adult sizes were based on the author's hands.

2.2 Function of Grasping and Handling (Dynamic Viscoelasticity and Shape of Fingertip)

The primary functions of grasping and handling using the prosthetic hand are classified into eight types: lateral pinch, hook grip, tip pinch, tripod grip, spherical grasp, cylindrical grasp, wrist flexion/extension, and supination/pronation.

A study by Arimoto and Hirai mathematically analyzed the stability of object handling using a robotic finger with a flexible fingertip [4, 5]. The result shows that hemispheric fingertips with flexible material property have a function which guides the objects to the stable position automatically. Such function is determined as passive stability which is able to be achieved under the conditions of high viscosity and adequate elasticity, which enables the distribution of local pressure of the contact point between the fingertip and object (Fig. 1). Meanwhile, the joint part of

Fig. 1 Flexible fingertip



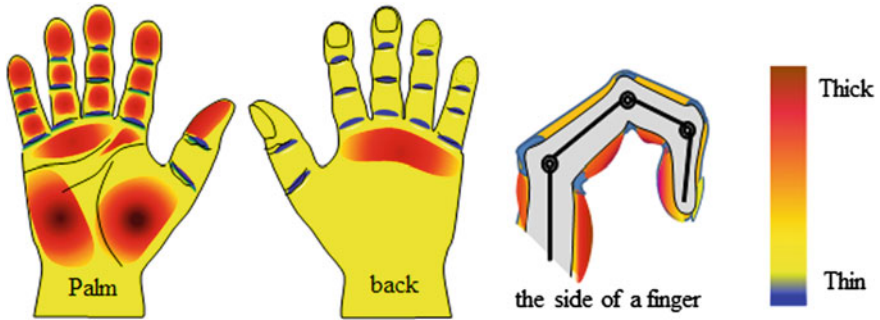


Fig. 2 The thickness map of the glove

the glove needs to follow the joint motion with expansion and shrinkage, and its viscosity factor should be low as possible.

Therefore, in order to achieve stable grasping capability, the thickness map of the glove has been newly designed, as shown in Fig. 2, where the palm side of the glove is made thin and the dorsal side and fingertip are made thick.

2.3 Durability

Commercial glove products are made of two materials, PVC and silicon. PVC offers high durability and shock resistance, and many users rely on cosmetic gloves made of PVC or silicon for daily use of their prosthetic hands. Furthermore, PVC and silicon are good materials for natural coloring and shaping; therefore, it is easy to obtain a natural appearance with these materials. However, both materials have problems: the color of PVC changes to black over time, and silicon can easily be broken. Public financial support has been provided in Japan for the exchange of cosmetic gloves every 2 years; therefore, the material should be sufficiently durable to last longer than 2 years. Furthermore, both materials have high elasticity and low viscosity factors; therefore, the dynamic movement of a prosthetic hand, which requires high energy consumption, is hampered. Moreover, the product is required to be waterproof and abrasion resistant because the glove may be used to touch or pick up food. For the purposes of durability and abrasion resistance, thermoplastic styrene elastomer (TSE) has been selected.

2.4 Bio-texturized Feeling

The mechanical characteristics of finger pulp have been studied by AIST group members. Maeno [6] reported on the elasticity of finger pulp, and reported that the

average ratio of elasticity among the outer layer of skin, inner skin, and subcutaneous tissue is approximately 8:5:2. The Young's moduli of these three parts are 1.36×10^{-1} (MPa), 8.0×10^{-2} (MPa), and 3.4×10^{-2} (MPa) respectively. Viscoelasticity in the shear direction has been reported by Nakazawa [7]; the elasticity is 1.7×10^{-3} (MPa), and the viscosity is 1×10^{-5} (MPa). Therefore, we need a material that has the approximate characteristics of finger pulp and rheology measurement has thus been adopted for detecting material characteristics and determining their adjustment conditions in order to compare TSE and finger pulp.

2.5 Flexibility

A large deformation of skin is induced at the finger joint by wrinkling according to the flexibility and extension of skin. Therefore, the skin deformation needs to absorb differences in extension and contraction. As mentioned previously, PVC has been used for a long time in cosmetic gloves because it has the advantages of durability and shock resistance. However, a disadvantage of PVC is its high elasticity, which prevents flexible finger motion. In contrast, silicon has the advantages of providing an authentic flexible feeling with low elasticity, but it can easily be broken by friction in daily use [8, 9]. Furthermore, the finger is an important interface for communication among humans. Therefore, the glove material should have a human-like touch, external appearance, and durability. Therefore, in this study, the injection forming method is applied to produce thin wrinkles at finger joints by using an internal die-casting and outer die-casting. The clearance of the internal die-casting and outer die-casting is controlled to produce a TSE glove with thickness ranging from 1.5 to 5 mm (Fig. 3).

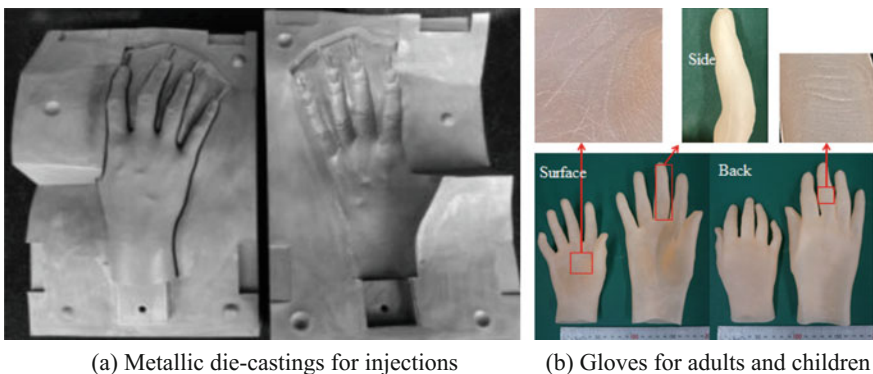


Fig. 3 Gloves which made with injection molding

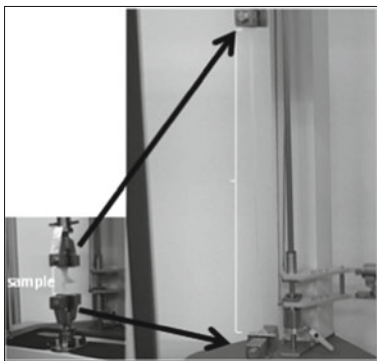
3 Experimental Result of Functional Evaluation

This section describes an experiment comprising of a functional evaluation of the cosmetic glove made of TSE. The experiment involved four kinds of evaluation: (3.1) tear-strength comparison, (3.2) fitting-time comparison, (3.3) electric current of EMG prosthetic hand motion comparison, and (3.4) flexibility comparison.

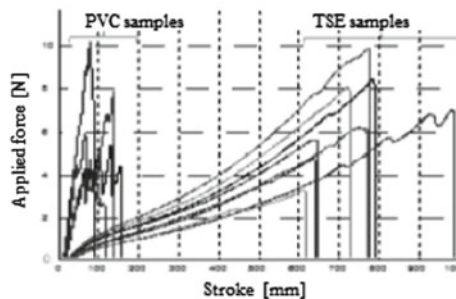
3.1 Tear-Strength Comparison Test

In order to test the capabilities of extension and contraction as well as tear resistance, a tear-strength test was designed using the universal tester autograph (AGS-100NX, Shimadzu co. Ltd) with approximation test JIS K 6252. The experiment compares two sample tips in four kinds of TSE gloves and two kinds of PVC gloves. The sample tips are rectangular with a size of 80×20 mm, and have a 40-mm notch in the narrow side. The end points of 10-mm depth are clamped at the chuck of the autograph tester. Stress is applied with a speed of 300 mm/min until breakage (Fig. 4a). The time series data for both the strain and applied force are indicated in (Fig. 4b); they are analyzed using the thickness of samples D (mm), applied force at the break point F [N], strain, and tear strength $P = F/D$ (kN/m).

As shown in (Fig. 4b), both the TSE and PVC samples have the same tear strength. Although the strain until breakage is 600–1000 mm for TSE, it is 130–160 mm for PVC. Therefore, TSE is superior to PVC. Furthermore, the TSE sample was able to maintain extension capability until breakage, whereas the PVC sample was easily broken without extension.



(a) Attach of sample to autograph



(b) Change of a stroke and applied force tester

Fig. 4 Tear-strength comparison test

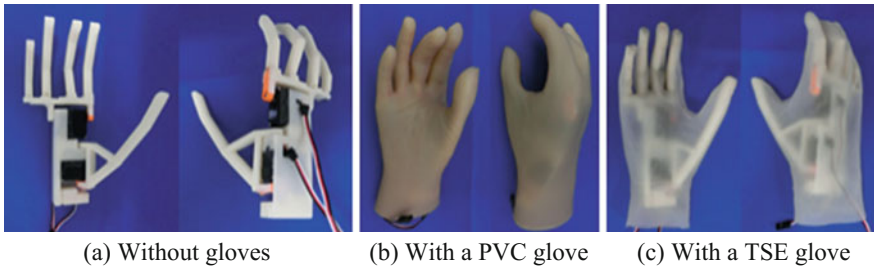


Fig. 5 Prosthetic hand and gloves

3.2 Fitting-Time Comparison

This section presents the result of the evaluation of flexibility when fitting a glove on a prosthetic hand by comparing the times required for glove fitting. The fitting time of the experimental glove to the prosthetic hand by an operator familiar with the structure of the prosthetic hand is compared with the fitting time of a PVC glove for a myoelectric prosthetic hand. The prosthetic hand is fitted with two motors at the CM joint of the thumb and the MP joint of the other fingers. The thumb CM joint is able to rotate by 90° from an opposing position to a parallel position. The other four fingers are able to rotate from extension at 30° to flexion at 90° . The bone parts are produced by a 3D printer by using ABS plastic material (Fig. 5a). The PVC glove and the TSE glove weigh 70 g and 60 g respectively (Fig. 5b, c). The experiments are repeated five times for each glove fitting trial, and the time taken is measured for each trial. By measuring the fitting time, the efficiency and adaptability of the glove are evaluated.

We found that all gloves could be fitted to the prosthetic hand in all experiments. The fitting time, average, and standard deviations (SD) are listed in Table 1. The fitting time of the PVC glove is 170 (s) on average, compared to 63 (s) on average for the TSE glove. The SDs are sufficiently small (PVC: 5.4 (s), TSE: 7.3 (s)). These results can be attributed to the difficulty of fitting PVC due to its low flexibility. The flexible TSE facilitated easy fitting to achieve fast functioning.

Table 1 Results of glove fitting time

Fitting time (s)	PVC glove (s)	TSE glove (s)
Trial 1	178	76
Trial 2	173	64
Trial 3	169	58
Trial 4	162	60
Trial 5	169	63
Average	170	63
Standard deviation	5.4	7.3

3.3 Electric Current During EMG Prosthetic Hand Motion

In order to detect and compare the amount of loading in the PVC glove and TSE glove, the electric current is measured at the time of opening hand, grasping hand, bending both the index finger and middle finger, and opposing motion of the thumb (Fig. 6). The electric-current sensor (AS-AM, Asakusa Giken) is connected to the energy supply line and measured using an AD converter (ADA16-8/2(CB)L, Contec) with 1.6 (kHz) sampling 16 (bit). A smoothing filter is applied to the measured data with a frame width of 100 points. Next, the average electric current during the stable state of the prosthetic hand posture and the maximum electrical current during the moving state are obtained. The experiments are repeated five times both with and without each glove, and the average and SD are calculated.

The result showed no differences between the cases with and without the TSE glove; the average currents are 50 mA each. However, in the case of the PVC glove, the averages ranged from 150 to 250 mA. This result is three to five times greater than that in the case of the TSE glove. This result indicates that the load of the TSE glove is lower than the measurement limit. Furthermore, the increase of load with the use of the PVC glove is significant and approximately four times greater than the increase with the use of the TSE glove.

In the next experiment, the maximum electric current at the state of changing hand posture is measured. The result shows that the PVC glove needs a significantly higher current than the TSE glove, except in the state of the hand posture changing from open to closed. Furthermore, the maximum current measured in the TSE glove case is the same as that measured in the case without a glove, except in the state of grasping hand.

These results show that the TSE glove successfully reduces energy consumption for many cases of hand movement.

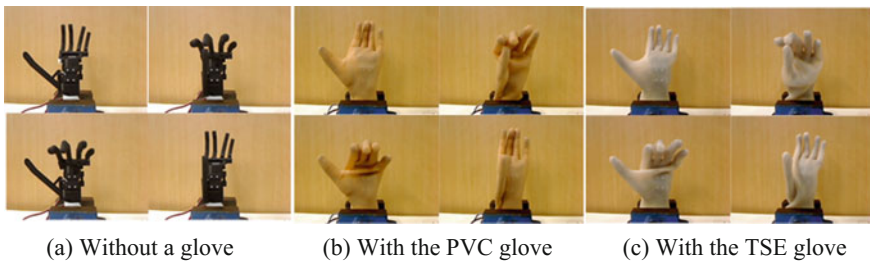


Fig. 6 The statue of 4 kind of motion (Open, Grasp, Bend fingers, and Oppose motion of thumb)

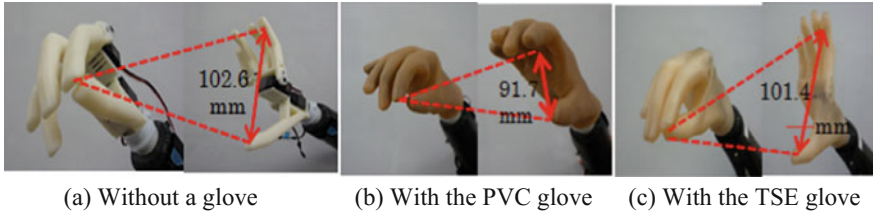


Fig. 7 The width of opening motion of the index finger



Fig. 8 The width of the opening motion of the thumb

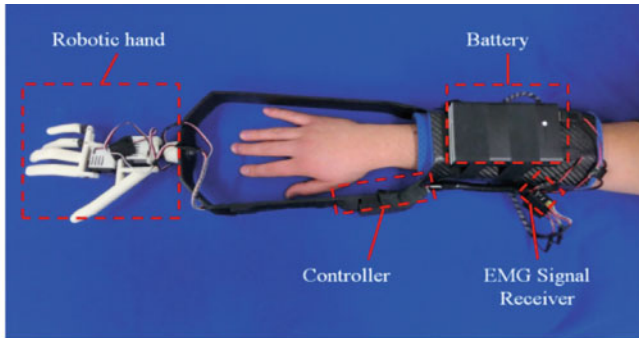
3.4 Flexibility

In order to evaluate the flexibility of the produced glove, two patterns of opening widths of hand are measured. The width between the top of the thumb and the index finger at the limit of the opening motion of the index finger from the precision tip pinch posture is shown in (Fig. 7). The width between the top of the thumb and the index finger at the limit of the opening motion of the thumb from the precision tip pinch posture is shown in (Fig. 8). The result shows that the PVC glove case strongly restricts the width of the opening hand. However, the differences between the TSE glove case and the case without a glove are very small; therefore, the TSE glove is able to realize the same range of motion as in the case without a glove.

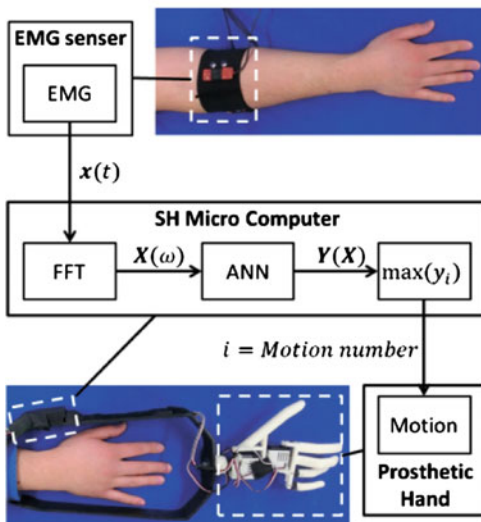
4 Pick-and-Place Experiment

This chapter describes an experiment for evaluating the overall performance of the produced TSE glove by using the pick-and-place test that makes comparisons between a prosthetic hand without a glove case, with a PVC glove case, and with a TSE glove case. The performance test is applied to the produced EMG prosthetic hand system for healthy subject shown in (Fig. 9a).

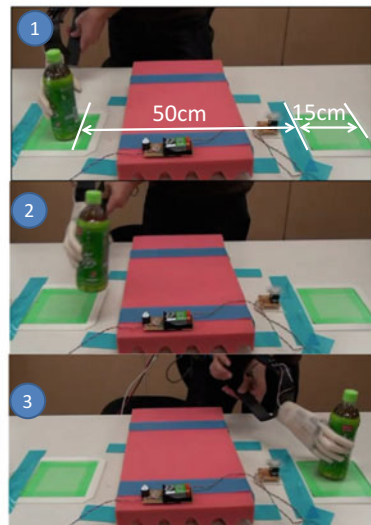
The EMG prosthetic hand system contains a robotic hand, three EMG sensors that are attached on the forearm, an EMG controller that discriminates EMG signals by using fast Fourier transformation as the input for an artificial neural network with



(a) EMG prosthetic hand system



(b) System Algorithm



(c) Pick-and-Place test

Fig. 9 Experiment system using EMG prosthetic hand and glove

a back-propagation algorithm, and a labeling learning algorithm (Fig. 9b). The EMG controller is able to distinguish up to 15 patterns of finger motion of a healthy subject, amputee, and a subject with a congenital defect.

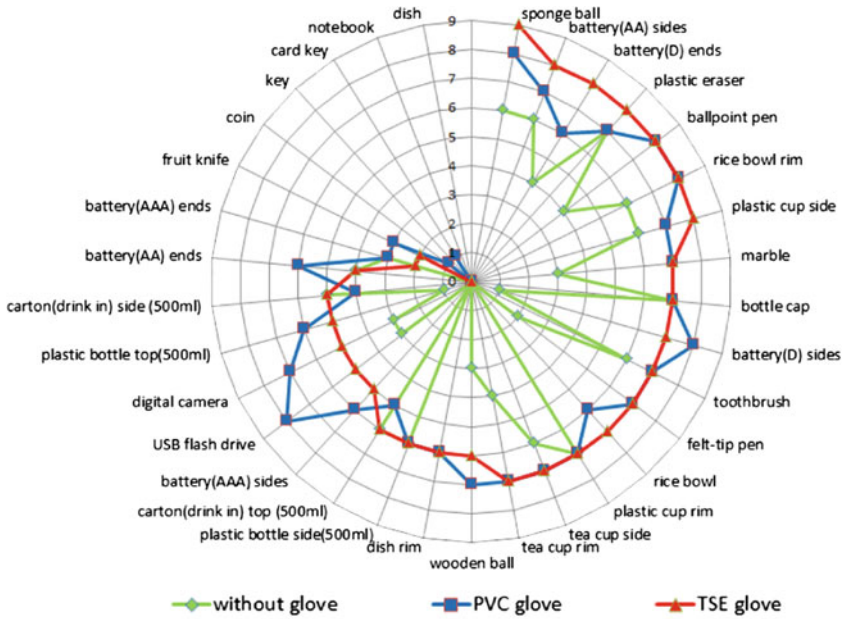
The pick-and-place test consists of three phases using an EMG prosthetic hand. In the first phase, an object is picked up from a 15 cm rectangular area. In the second phase, the object is carried from the first area to another rectangular area located at a distance of 50 cm. In the final phase, the object is put down, and the counter switch is pushed. The performance was evaluated by counting the number of switches within 30 s (Fig. 9c).

Table 2 Objects used in pick and place experiment

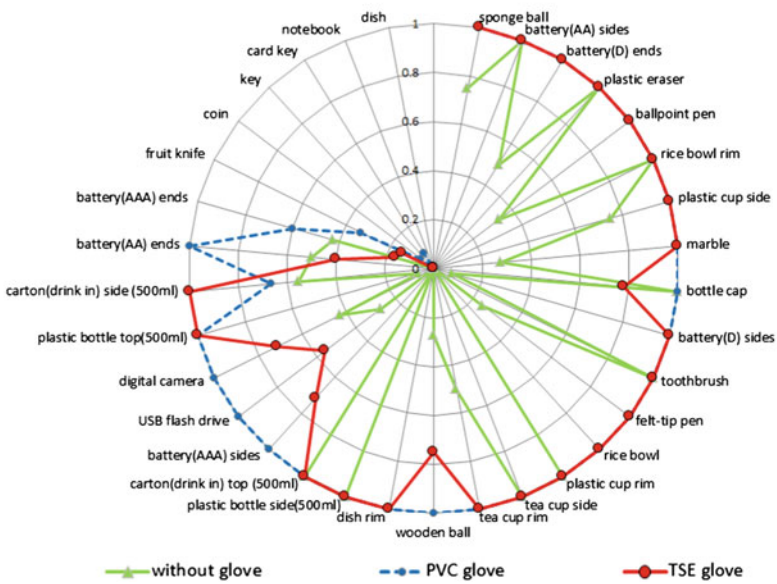
Object	Weight (g)	Shape of grasp part	Size (mm)
Marble	6	Sphere	Diameter 16.9
Wooden ball	24	Sphere	Diameter 40.7
Sponge ball	33	Sphere	Diameter 70
Bottle cap	2	Column	Diameter 30, Height 14
Battery (AAA)	11	Column	Diameter 10.2, Height 44.4
Battery (AA)	23	Column	Diameter 13.9, Height 50.2
Battery (D)	103	Column	Diameter 32.8, Height 61.1
Coin	5	Thin board	Diameter 25, Thickness 1.6
Key	10	Thin board	Depth 13.9, Width 25.8, Thickness 2.3
Card key	4	Thin board	Depth 54, Width 85.7, Thickness 0.6
USB memory	8	Rectangular solid	Depth 18.7, Width 59.8, Thickness 8.5
Plastic eraser	18	Rectangular solid	Depth 58.8, Width 24.4, Thickness 10.7
Digital camera	113	Rectangular solid	Depth 55.2, Width 93.6, Thickness 19
Notebook	138	Rectangular solid	Depth 257, Width 182, Thickness 3.8
Toothbrush	13	Rectangular solid	Depth 7.1, Width 12.8
Ballpoint pen	13	Cylindrical	Diameter 12.6
Felt-tip pen	24	Cylindrical	Diameter 17.6
Fruit knife	50	Rectangular solid	Depth 18.5, Width 10.5
Rice bowl	132	Dome-type	Diameter (upper)120, Diameter (lower)45, Height 54
Dish	263	Disc-type	Diameter 180, Thickness 18
Plastic cup side	49	Cylindrical	Diameter 71.5
Teacup	173	Cylindrical	Diameter 69.5, Height 92.4
Plastic bottle (500 ml)	534	Cylindrical	Diameter 60.2
Drink box (500 ml)	537	Rectangular solid	Depth 73, Width 73

The objects selected for the test are used in daily life (Table 2). The overall performance is shown in Fig. 10. Both gloves significantly improve upon the case of a prosthetic hand without a glove.

The radar chart which indicated the success rate (Fig. 10b) shows that the success rate of the PVC glove seems to be equal to a TSE glove. But the radar chart of the frequency of success (Fig. 10a), you find out in the result that there is a lot of frequency of success in a glove made of TSE. It seems that the PVC glove



(a) Frequency of success



(b) The success rate

Fig. 10 Results of pick-and-place experiments

is possible to be a steady grip, but it won't release when it holds something because the surface is hard and less flexibility. On the other hand, because the TSE glove's flexibility is high and doesn't have unnecessary load, it seems that the rate of either open motion or shut motion of the prosthetic hand covered with TSE glove didn't decelerate.

Thus, we were certain that the TSE glove is able to follow the robotic hand's move and not to disturb the movement.

5 Summary

This paper described the development and assessment of a new cosmetic glove for EMG prosthetic hands by considering five design requirements based on existing studies and practical use. The five design requirements were cosmetics, grasping and handling capability, durability, natural feeling, and flexibility.

In order to obtain a natural cosmetic glove, the TSE material was applied. The TSE glove achieved a human-like shape and texture with the help of existing research results and precise mold shaping for sizes ranging from children to adult.

For high performance in grasping and handling, passive stable control using two fingers was applied together with the injection molding method, and adequate flexibility was achieved by using an adequate thickness map with a thick fingertip and palm and thin joint part.

The result that TSE material is selected for durability and flexibility, the extension rate of glove is 1300 %. The characteristics of TSE are optimized through comparisons of its Young's modulus with that of human skin.

For evaluating the overall performance, a material engineering test and pick-and-place test with an EMG prosthetic hand were conducted. The material engineering test involved a property evaluation test, tear-strength test, and cosmetic evaluation. The pick-and-place test with an EMG prosthetic hand involved a fitting test, flexibility test, and handling performance test.

Acknowledgements This research is partially supported by "Brain Machine Interface Development" and "A-step" from Japan Agency for Medical Research and development, AMED, and also Takahashi Industrial and Economic Research Foundation.

The authors would like to thank the EMG prosthetic hand team members: Yuta Murai, Naoyuki Tani, Yutaro Hiyoshi, Yuta Suzuki, Feng Xiang, Hesong Ye, Shintaro Sakoda, Yusuke Yamanoi, Kenichi Takahana, Tatsuya Seki, Masahiro Kasuya, and Soichiro Morishita, for their great help in testing the prosthetic hand and the glove.

References

1. Seki, T., Nakamura, T., Kato, R., Morishita, S., Yokoi, H.: Development of five-finger multi-dof myoelectric hands with a power allocation mechanism. *J. Mech. Eng. Autom.* **4**, 97–105 (2014)
2. Nishioka, K.: WIME HAND. *Jpn. Soc. Prosthet. Orthot.* **9**(4), 347–351 (1993)
3. Kawamura, J., Fukui, N., Nakagawa, M., Fujishita, T., Aoyama, T., Furukawa, H.: The Upper-Limb Amputees: a Survey and Trends in Kinki Area of Japan, vol. 36(6), pp. 384–389. The Japanese Association of Rehabilitation Medicine (1999)
4. Tahara, K., Yamaguchi, M., Arimoto, S.: Sensory Feedback for Stable Grasping and Posture Control by using a Pair of Minimum-DOF Robot Fingers with Soft Tips, vol. 21, no. 7, pp. 763–769. The Robotics Society of Japan (2003)
5. Inoue, T., Hirai, S.: Secure Grasping Effect of Soft-fingered Manipulating motions with minimum D.O.F. Robotic Hand, T.SICE, vol. 42, no. 1 (2006)
6. Maeno, T., Kobayashi, K., Yamazaki, N.: Relationship between structure of finger tissue and location of tactile receptors. *Trans. JSME* **63**(607), 881–888 (1997)
7. Nakazawa, N., Ikeura, R., Inooka, H.: Characteristics of human fingertips in shearing direction. *Trans. JSME* **64**(624), 3076–3082 (1998)
8. Nakagawa, A., Chin, T., Nakura, H.: Experience of clinical use of emg hand. *Jpn Soc. Mech. Eng.* (3–28), 21–23 (2003)
9. Kawasaki, K., Okuno, R., Yoshida, M., Akasawa, K., Kawasaki, A., Miyamoto, Y.: Study of low deformation resistance rubber-glove for the biomimetic hand. *Inst. Electron. Inf. Commun. Eng.* (97–525), 109–115 (1998)

Part V
Human Robot Interaction

Cloud-Based Task Planning for Smart Robots

Elisa Tosello, Zhengjie Fan, Alejandro Gatto Castro and Enrico Pagello

Abstract This paper proposes an Open Semantic Framework for knowledge acquisition of cognitive robots performing manipulation tasks. It integrates a Cloud-based Engine, which extracts discriminative features from the objects and generates their manipulation actions, and an Ontology, where the Engine saves data for future accesses. The Engine offloads robots by transferring computation on the Cloud. The Ontology favors knowledge sharing among manipulator robots by defining a common manipulation vocabulary. It extends the work proposed by the IEEE RAS Ontology for Robotics and Automation Working Group by covering the manipulation task domain. During ontological data insertion, data duplication is avoided by providing a novel efficient interlinking algorithm. During their retrieval, visual data processing is optimized by using a cascade hashing algorithm that intelligently accesses data. No training is required for object recognition and manipulation because of the adoption of a human-robot cooperation. The framework is based on the open-source Robot Operating System.

Keywords Cognitive robotics · Task planning · Cloud-based object manipulation engine · Ontology · Core ontology · ROS

E. Tosello (✉) · A.G. Castro · E. Pagello (✉)
Intelligent Autonomous Systems Lab (IAS-Lab) Department of Information
Engineering (DEI), University of Padova, Via Gradenigo 6/B, 35131 Padova, Italy
e-mail: toselloe@dei.unipd.it

A.G. Castro
e-mail: alejandro.gattocastro@studenti.unipd.it

E. Pagello
e-mail: epv@dei.unipd.it; zepv@dei.unipd.it

Z. Fan
Big Data Team Department of Big Data and IT Technology,
China Mobile Research Institute (CMRI), Innovation Building, No. 32, Xuanwumen West
Avenue, 100053 Beijing, Xicheng District, People's Republic of China
e-mail: fanzhengjie@chinamobile.com

1 Introduction

Robots of the future should be Smart: versatile and efficient Artificial Intelligence systems able to perform behaviors of growing complexity, adapt to changes, collaborate with humans and other robots, learn from the past and from action performed by other agents, and build on their capabilities based on that knowledge. Two are the reasons that limit robots intelligence: the limited capacity of on board data elaboration and the absence of a common medium to communicate and share knowledge. Tapping into the Cloud is the solution. A Cloud server allows to offload robots from CPU-heavy tasks and to perform intensive computation while meeting the hard real-time constraints of operations. A Web Ontology lets the definition of a common vocabulary that ensures a common understanding during the interaction as well as an efficient data transfer and integration.

In automation, a large amount of objects should be real-time manipulated. Failures can lead to high costs. Moving data and computation to the Cloud favors data sharing [1], enables robots learning the stability of finger contacts from previous manipulations on the same object [2], and lets the application of strategies used on some objects on similar parts encountered later [3].

This paper proposes an Open Semantic Framework for knowledge acquisition of cognitive robots performing manipulation tasks. An Ontology and a Cloud-based Engine have been implemented. The former stores data about objects and actions necessary to manipulate them. The latter detects objects in the scene and retrieves their manipulation actions from the Ontology. If no ontological data exists, the Engine generates and stores it on the Ontology.

The rest of the paper is organized as follows. Section 2 details researches done in this context and compares existing works with the one proposed by this paper. Section 3 describes the Ontology design and its implementation details. Section 4 shows the Cloud-based Engine focusing on the intelligent ontological data insertion and retrieval. It also highlights how the elimination of the training phase for object recognition and manipulation and its substitution with a human-robot cooperation makes gradual the robots knowledge growth. Section 5 proves the good performances of the proposed system by discussing the experiments done. Section 6 presents conclusions and future work.

2 Related Work

Different Cloud Platforms exist. Their inadequacy for robotics scenario is mainly due to the difference in Web applications and robotics applications. Many Web applications are stateless, single processes that use a request-response model to talk to the client. Robotic applications are state-full, multi-processed, and require a bidirectional communication with the client. An example of efficient and widespread Cloud Platform which, however, is not suited for robotics applications, is the Google App

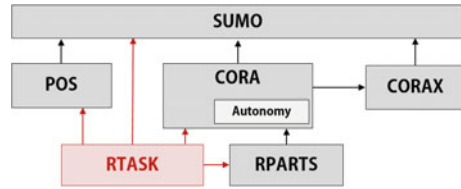
Engine.¹ It exposes only a limited subset of program APIs tailored specifically for Web applications, allows only a single process, and does not expose sockets, which are indispensable for robotic middlewares such as the open-source Robot Operating System (ROS) [4]. In order to overcome these limitations, some Cloud Robotics Platforms have been implemented. An example is Rapyuta, the RoboEarth Cloud Engine [5], a platform designed for robots to share data and action experiences with each other. With respect to Rapyuta, the proposed Engine focuses only on the robotics sharing of manipulation data and actions, but guarantees an efficient Cloud data access by adopting a cascade hashing algorithm [6]. Moreover, it avoids data duplication during the insertion of new data by using a novel powerful interlinking algorithm [7]. The algorithm finds the interlinking pattern of two data sets by applying two machine learning methods: the K-medoids [8] and the Version Space [9].

Focusing on the Knowledge Base to which the Engine accesses, many existing works are available online. Examples are the Columbia Grasp dataset [10], the KIT object dataset [11], and the Willow Garage Household Objects Database [12]. KnowRob [13], the knowledge base of RoboEarth [14], is the most widespread. They store information about objects in the environment and their grasp poses. The Household object database is a *simple* SQL database: the SQL format does not favor robotics knowledge scalability. The others are well-defined by an Ontology. RoboEarth models objects as 3D colored Point Clouds [15], the others store objects as triangular meshes. Stored items are of high quality but each object model consists of several recordings from different point of views; thus requiring either a lot of manual work or expensive scanning equipments.

With respect to the existing Knowledge Bases, the one proposed, named RTASK, is scalable because of the adoption of an Ontology that defines data. It guarantees an intelligent data storage and access because of the type of data saved. Every object is characterized by multiple visual features (2D Images, B-Splines, and Point Clouds). When detecting an object, the recognition process starts the comparison of the smallest features (e.g. the ones representing the 2D Images), and eventually expands to the others (B-Splines and Point Clouds, in increasing order). No onerous manual work is required to store objects from different view points: an object is stored even if there exists only a single registration of one its views. A human teacher helps robots in recognizing objects when viewed from other orientations. The teacher exploits the connection between the new view point and other object properties, e.g., name and function. These new features will be stored in the Ontology gradually incrementing robots knowledge about the object itself. Moreover, the proposed Ontology observes the IEEE standards proposed by the IEEE Robotics and Automation Society (RAS)'s Ontology for Robotics and Automation (ORA) Working Group (WG) by extending the Knowledge Base it proposed (see Fig. 1) [16–18]. Respecting the standard, Balakirsky et al. [19] proposed a kitting ontology. RTASK generalizes the manipulation concept by introducing the notions of manipulation *task* and *action*. This means that any manipulation task can be represented, e.g., grasps and pushes.

¹Google, Inc. “Google App Engine”. Online: <https://cloud.google.com/appengine/> (2014).

Fig. 1 The RTASK extension to the IEEE ontology for robotics and automation



3 The Ontology

RTASK formulates a common vocabulary for robotics manipulation. As proposed in [20], it separates the concepts of *tasks* and *tasks executions*. Tasks are *abstract* entities that describe goals to be reached; while tasks executions are *events* composed of *actions* that are performed by robots in order to reach goals.

3.1 Design

Figure 2 depicts the Ontology design: a **Task** is assigned to an **Agent**, e.g., a **Robot**. It should be executed within a certain **time** interval and requires the fulfillment of a certain **Motion** in order to be performed. **Manipulation** is a sub-class of **Task**. Several types of manipulations exist, e.g., grasps and pushes. They involve the handling of an **Object** located at a certain **Pose** (**Position** and **Orientation**) through the execution of a **Manipulating** action. If the **Task** is assigned to a **Robot**, then the **Motion** will be represented by a **Robot Action**. In detail, the **Robot Manipulation Action** involves the activation of the robot **End Effector**. Studies have demonstrated that: (i) placing the arm in front of the object before acting improves actions; (ii) humans typically simplify the manipulation tasks by selecting one of only a few different prehensile postures based on the object geometry [21]. On this view, the **End Effector** is first placed at a **Pose** p' at a distance d from the **Object**, with its actuatable **Joints** at a certain **Pre** manipulation **Posture**. Then, the **End Effector** is placed at a **Pose** p close to the **Object** and the effective manipulation **Posture** is assigned to its **Joints**.

In order to retrieve the manipulation data of an object in the scene, the object should be recognized as an instance previously stored in the Ontology. For this purpose, every **Object** is characterized by an *id*, *name*, *function*, and the visual features obtained by the **Sensors**. For every **Object**, RTASK stores multiple types of visual features: **2D Images**, **B-Splines**, and **Point Clouds**.

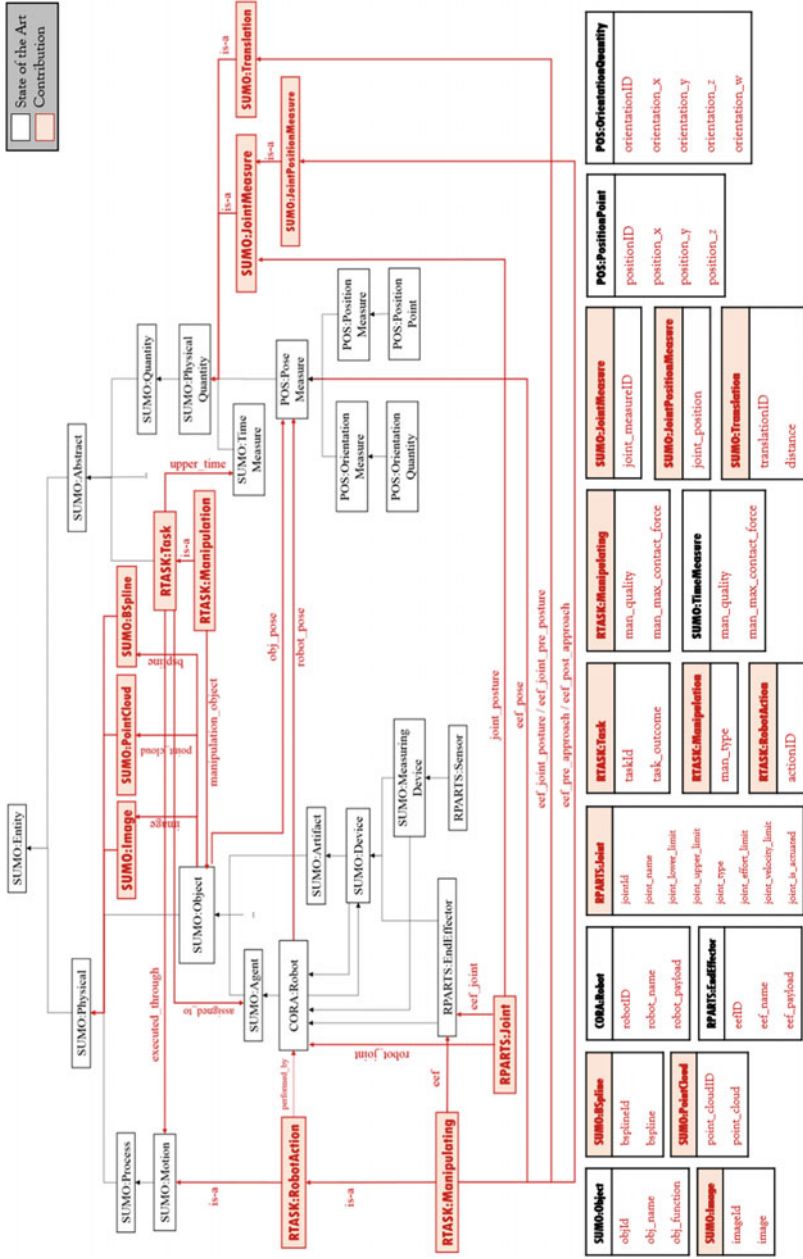


Fig. 2 RTASK: The ontology

3.2 Implementation Details

RTASK is represented through the union of the Resource Description Format (RDF)² and the Web Ontology Language (OWL),³ namely OWL Full. RDF is used to define the structure of the data, OWL adds semantic to the schema and allows the user to specify relationships among data. OWL Full allows an ontology to augment the meaning of the RDF vocabulary guaranteeing the maximum expressiveness of OWL and the syntactic freedom of RDF. Indeed, OWL is adopted by the World Wide Web Consortium (W3C)⁴ and it is the representation language used by the RAS ORA WG. Protégé is used as ontology editor.⁵

Queries allow robots to investigate the knowledge base and retrieve existing data. A robot able to query the database has the capability of efficiently and intelligently perform tasks. In our case, a C++ interface lets ROS users query RTASK using SPARQL.⁶ Apache Jena Fuseki is used as SPARQL server.⁷

4 The Cloud-Based Engine

The current implementation of the Cloud-based Engine is based on a Cloud-based Object Recognition Engine for robotics (CORE) [22]. The robot has an internal *ROS node* that receives the segmented objects (objects are segmented using the functions offered by the Point Cloud Library [23]) and sends them to the Cloud-based Engine. The Engine is composed on another *ROS node* capable of reading the content of received messages. The communication is based on the **ros_bridge** interface, which provides a web socket channel between nodes.

4.1 Data Retrieval

4.1.1 Objects Manipulation Data Request

The robot asks the Cloud Server for the retrieval of the manipulation data of an object. It sends a message containing the type of its gripper and the compressed Point Cloud of the manipulable object. The compressed Point Cloud representation saves space and connection time. The compressed Point Cloud is encoded in order to

²Resource Description Format (RDF). Online: <http://www.w3.org/RDF>.

³Web Ontology Language (OWL). Online: <http://www.w3g/TR/owl-features>.

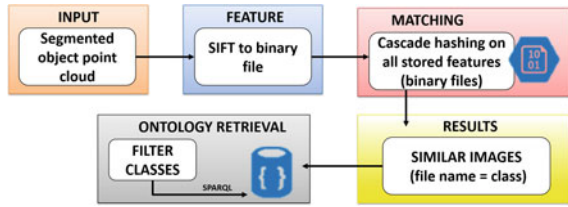
⁴World Wide Web Consortium (W3C). Online: <http://www.w3c.com>.

⁵Protégé. Online: <http://protege.stanford.edu/>.

⁶Simple Protocol and RDF Query Language (SPARQL). Online: <http://www.w3.org/TR/sparql11-query>.

⁷Apache Jena Fuseki. Online: <https://jena.apache.org/documentation/fuseki2/index.html>.

Fig. 3 Image processing pipeline of the matching phase



be transmitted over the Web Socket channel. After the encoding, the whole message is represented as a JavaScript Object Notation (JSON) object. The ROS message being encoded on the Web Socket request follows.

```
# Task name (pick , place for example)
string task
# Gripper name
string gripper_id
# Compressed Point Cloud (object representation)
string data
```

The Server receives the Client request and performs a super-fast search of the object inside RTASK. The search starts from the comparison of the object 2D Images features (SIFT, Scale Invariant Feature Transform) [24] and, in case of mismatch, ends with the comparison of its Point Clouds features (HOG, Histogram of Oriented Gradients) [25]. Steps of the super-fast search of SIFT features follow (see Fig. 3).

1. Decode the message and decompress the Point Cloud;
2. Convert the Point Cloud to color image using the Open-source Computer Vision (OpenCV) [26] (OpenCV) functions;
3. Extract the image SIFT features⁸ and store them on a binary file of a Server folder;
4. Match the features with the ones stored in the Server;
5. SPARQL query RTASK;
6. The Server returns a *moveit_msgs::Grasp* ROS message containing the relative manipulation data.

The search is fast because of the novel and super-fast Cascade hashing algorithm adopted during the matching phase (Step 4) [6] and because of the way in which features are stored. The algorithm allows constructing a dataset without a learning phase: there is not need to train the hashing function as in other Approximate Nearest Neighbor (ANN) methods. Given the input image, the function returns the names of its most similar images features (according to the SIFT parameters), the names are in the form of integer numbers. The same names define the object classes of RTASK. Moreover, features of stored objects are precalculated and stored on a Server folder: they are not calculated at every data set access. Thanks to the combination of these

⁸Chris Sweeney, Theia Multiview Geometry Library: Tutorial & Reference. Online: <http://theia-sfm.org>.

Algorithm 1 Interlinking Instances across Data Sets

Input: Two Data Sets
Output: Links across Data Sets

```

1: The data set  $D, D'$ ; /*two data sets to be interlinked*/
2: Similarity threshold  $T$ 
3: for Each property/relation in the data set  $D$  do
4:   for Each property/relation in the data set  $D'$  do
5:     Match properties/relations that are corresponding to each other and store as the alignment  $A$ 
6:   end for
7: end for
8: for Each instance in the data set  $D$  do
9:   for Each instance in the data set  $D'$  do
10:    Compare instances' property values according to the correspondences of the alignment  $A$ ;
11:    Aggregate all similarities between property values as a similarity value  $v$ 
12:    if  $v \geq T$  then
13:      The two compared instances are interlinked with owl:sameAs.
14:    end if
15:  end for
16: end for

```

two characteristics only one query at the end of the matching process is needed in order to retrieve the similar objects stored on the Ontology.

4.2 Data Insertion

An example of Client message aiming to create a new Object instance follows. It is encoded as JSON string

```

{ "op": "send_new_object",
  "service": "insert",
  "new_object":
  [
    { "new_point_cloud": "zc5p81H1cO8P+Ksmfdf...",
      "X": "0.5", "Y": "2.5", "Z": "1.5", "rad": "0.314" },
  ]
}

```

When the Server receives the message, it calculates the relative SIFT features and stores them on the features folder and on RTASK.

During the insertion of elements in RTASK, duplication avoidance is desirable. To this end, Algorithm 1 is applied to automate the interlinking process. It was proposed in [7] and finds out the interlink pattern of two data sets by applying two machine learning methods: the K-medoids and the Version Space. Although interlinking algorithms require interactions with users for the sake of the interlinking precision, computations of comparing instances are largely reduced than manually interlinking. As the work-flow of Algorithm 1 shows, when interlinking two instances across two data sets D and D' , the algorithm first computes property/relation correspondences across two data sets (line 5). Then, instances property values are compared by referring to the correspondences (line 10). A similarity value v is generated

upon all similarities of property values (line 11). If such a similarity is equal to or larger than a predefined threshold T , the two compared instances can be used to build a link with the relation *owl:sameAs* (line 12–14).

4.3 Human-Robot Interaction

Usually, an onerous a-priori human manual work is required to store objects visual features on a Robotics Knowledge Base: every object is represented by a large amount of registrations from different points of view. Objects representation is accurate but the a priori work is onerous. Our approach eliminates this prerequisite by introducing a human teacher that supports robots during their learning phase. First registrations will not be accurate, but knowledge will gradually increase until becoming absolute and giving robots autonomy.

Figure 4 highlights the reasoning at the bottom of this cooperation approach. Robots require human intervention to confirm the identity of a recognized object or assign one to a new object. Moreover, humans help robots in connecting visual features of objects already stored in the Knowledge Base but seen from other points of view. To perform the connection, other objects properties are exploited, e.g., name and function.

Many advantages are introduced by this approach. For example, it eliminates the a priori work currently done by humans by introducing a cognitive and social robot able to interact with other agents, e.g., human operators. A key outcome follows: A robot capable of learning is a flexible, adaptable, and scalable cyber-physical system.

The message used for human-robot interaction follows.

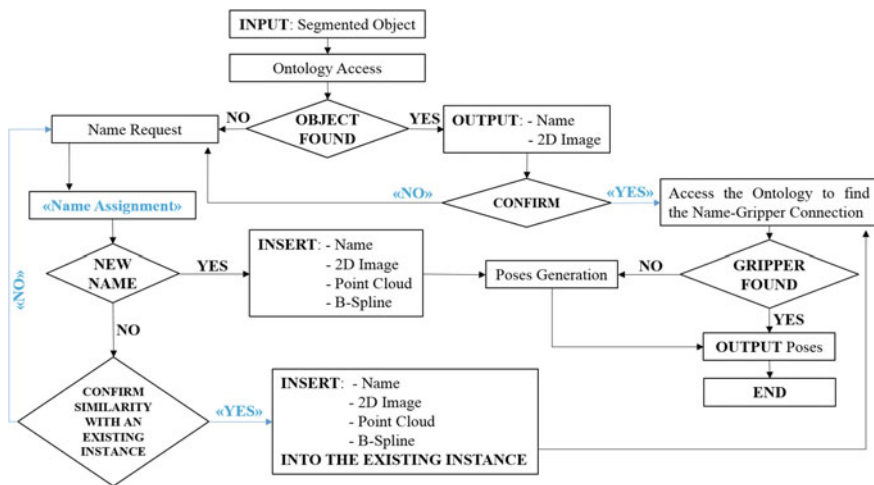


Fig. 4 The reasoning pipeline. The blue parts represent human interventions

```
# Operation type
string operation
# Object class name
string class_description
# Compressed Point Cloud (object representation)
string data
```

The human operator shows an object to the robot and gives it a description (e.g., coke, can, pen). The Server seeks for a similar object by filtering the Ontology through the object description. If matches exist, the Server returns a message containing the classes of the similar objects found:

```
{ "op": "search_object_response",
  "service": "query",
  "similar_objects":
  [
    { "class": "1", "X": "0.5", "Y": "2.5", "Z": "1.5", "rad": "0.314" },
    { "class": "2", "X": "0.1", "Y": "24.5", "Z": "1.3", "rad": "0.24" },
    { "class": "3", "X": "0.2", "Y": "3.5", "Z": "0.5", "rad": "2.14" }
  ]
}
```

a human feedback confirms the object class and visual features are added to it. Otherwise, a new instance is inserted in RTASK:

```
{ "op": "send_new_object",
  "service": "insert",
  "new_object":
  [
    { "new_point_cloud": "zc5p81H1cO8P+Ksmfdf...",
      "X": "0.5", "Y": "2.5", "Z": "1.5", "rad": "0.314" },
  ]
}
```

4.4 Manipulation Data Generation

Given new object, the Cloud-based manipulation planner generates a list of possible manipulations, each consisting of a gripper pose relative to the object itself. Manipulations consist of both grasps and pushes. The current version of the generator aligns the hand with the object principal axes, starting from either the top or the side of the object, and tries to manipulate it around its Center Of Mass through a trial-and-error Reinforcement Learning technique. As for GraspIt! [27] and the MoveIt! Simple Grasps tool developed by Dave T. Coleman,⁹ given the safety distance d at which the gripper must be positioned before making the manipulation, the generator returns:

⁹MoveIt! Simple Grasps tool. Online: https://github.com/davetcoleman/moveit_simple_grasps.

- a pre-manipulation configuration: the gripper pose and joints configuration at the safety distance;
- a manipulation configuration: the gripper pose and joints configuration to be maintained during the manipulation.

Experiments proposed by Dave T. Coleman demonstrate that the grasping tool he developed does not fail because, in the scene

- the block to be grasped is known a priori;
- the configuration that the gripper has to maintain during the grasp is manually predetermined according to the block's dimensions;
- collision checking is not performed to verify the feasibility of grasps.

If we reason about arbitrary shapes, collisions or contact losses can be induced by pre-selecting the manipulation configurations. To overcome the problem, the generator used by the Engine exploits the Reinforcement Learning benefits and generates grasps and pushes according to the input object.

5 Experiments

Experiments aim to provide truthful temporal results on Cloud access and ontological data retrieval. For this purpose, RTASK and the Engine have been integrated inside the Cloud environment of CORE, which is available on the Wisconsin Cloud-Lab cluster¹⁰ under the project “core-robotics”. It consists of one x86 node running Ubuntu 14.04 with ROS Indigo installed. Moreover, RTASK has been integrated with the Object Segmentation Database (OSD)¹¹: a data set of 726 MB currently containing 111 different objects, all characterized by a 2D color Image and a Point Cloud.

In simulation, an Husky mobile robot equipped with a Universal Robot UR5 manipulator and a Robotiq 2 finger gripper has to solve a tabletop object manipulation problem: it has to grasp the nearest object located on the table in front of it. A Microsoft Kinect acquires the scene. Gazebo [28] is used as simulator. A ROS Sense-Model-Act framework has been implemented in order to give the robot the ability to detect objects on the table, access the Cloud server, and retrieve manipulation data (see Fig. 5). Implementation details follow.

Sense The Kinect acquires the RGB-D images of the environment. From the collected data, a compressed segmented 3D Point Cloud of each individual manipulable object is computed using PCL.

Model On the Cloud, from the compressed Point Cloud, the relative 2D images is computed together with its B-Spline [29] representation. The relative visual features

¹⁰CloudLab. Online: <http://www.cloudlab.us>.

¹¹Object Segmentation Database. Online: <http://www.acin.tuwien.ac.at/forschung/v4r/software-tools/osd/>.

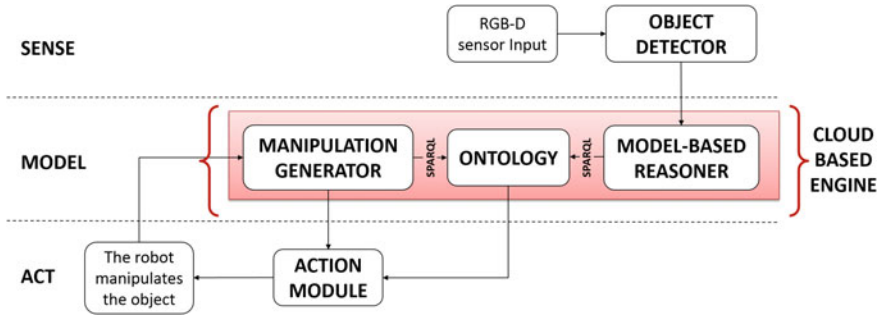


Fig. 5 The sense-model-act framework

are computed, e.g., SIFTs and HOGs. The cascade hashing algorithm starts searching for a match between the features of the segmented object and that saved in the data set. The comparison starts from SIFTs and eventually expands to B-Splines (by computing the squared distance among points) and HOGs. The reasoner accesses RTASK in order to retrieve the manipulation actions relative to the detected features. Again, if a match exists, together with the information relative to the assigned manipulation action (e.g., *push* or *grasp*) and to the relative gripper joints configuration, then the information will be outputted. Otherwise, the Manipulation Data Generator computes the necessary poses.

Act The module lets the robot move by activating its simulated engines. MoveIt!¹² generates the kinematic information required for the system to pass from the current to the goal configuration. During the motion, the information acquired by the sensors is used to compare the system final state with the expected one. In case of mismatch, a trial and error routine starts correcting the joints configuration. The configuration that allows the task achievement is saved in RTASK.

5.1 Results

Figure 6 shows the system in operation: the robot detects the surrounding environment, segments the objects in front of it, and grasps the nearest objects through the manipulation configuration retrieved from the Cloud.

Tables 1 and 2 reports the most significant time data. Table 1 depicts the time taken for the extraction of the descriptors of the 2D images stored in the data set. The table shows two types of extraction: the extraction of descriptors of all the 111 images contained in the data set and that of descriptors of a single image. For each type of extraction, tests were done on 1, 2, 3, and 4 threads respectively. Authors point out the 0.387422 s used to extract descriptors of a single image through the

¹²Ioan A. Sucas and Sachin Chitta, “MoveIt!”, Online: <http://moveit.ros.org>.

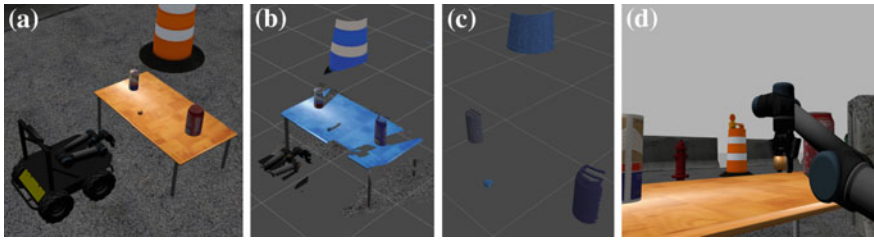


Fig. 6 a The robot in the scene; b The detected scene; c The segmented objects; d The grasp of the nearest object

Table 1 Elaboration time taken for features extraction

Features extraction				
	1 Thread (s)	2 Thread	3 Thread	4 Thread (s)
111 images	45.1554	21.8381 s	15.331 s	012.7545
1 image	0.379014	0.408295	0.396394	0.387422

Table 2 Elaboration time taken for features matching

Features matching		
	Time (s)	Matches on 6105 possible image pairs
15 inliers	4.82	1592
300 inliers	4.48500	18

employment of 4 threads. Table 2 focuses on times taken for features matching. By using 300 inliers, the match is accurate and the computational time does not affect the real-time constraints of a robotics manipulation.

The reported computational times consider SIFT features. Times gradually increments if B-Splines or HOGs are considered. Moreover, the computational effort depends on the richness of the features: the more complex the Point Cloud is, the greater the extraction time will be. The segmentation helps maintaining computational times low.

During the experiments, MoveIt! took on average 3.765 s to find a feasible inverse kinematics solution (best case: 0.162757 s; worst case: 7.367439 s; number of trials: 100) on a Dell Intel Core i7-4470 CPU @ 3.40 GHz x 8, 15.6 GiB Memory, 970 GB Disk. In the best case, it completed the planning after 0.8486 s. Reported data proves that the intelligent and efficient structure of the proposed Open Semantic Framework does not adversely affect the time required to complete a manipulation: executing the features extraction and matching on 4 threads off-loads robots and increases system performances.

6 Conclusion and Future Work

This paper presented an Open Semantic Framework able to increase robots knowledge and capabilities on objects manipulation. The Framework is composed of an OWL Ontology and a Cloud-based Engine. From the study of human actions when handling objects, the Ontology formulates a common vocabulary that encodes the robotics manipulation domain. The Engine, instead, was developed in order to transfer the computation on the Cloud: it off-loads robot CPUs and speeds up the robots learning phase. Given an object in the scene, the Engine retrieves its visual features and accesses the Ontology in order to extract the corresponding manipulation action. If no information is stored, a Reinforcement Learning technique is used to generate the gripper manipulation poses that will be stored on the Ontology. The Ontology respects the IEEE Standard by extending the existing CORA. The Engine minimizes visual data processing through an intelligent ontological data access and retrieval. During ontological data retrieval, a cascade hashing algorithm is adopted in order to optimize the comparison between saved and new visual features. Instead, in order to avoid data duplication during the insertion of new instances in the Ontology, a novel efficient interlinking algorithm has been adopted. Furthermore, the training for objects recognition and manipulation is replaced by a human-robot interaction.

We proved the efficiency and effectiveness of the proposed approach by building a ROS Sense-Model-Act framework able to associate manipulation actions to the features of the objects in the scene. Tests were performed in simulation.

As future work, we aim to extend the proposed Ontology by defining other tasks and actions, e.g., we would like to explore the Navigation domain. We aim to assign robots the new tasks and execute the relative new actions in order to increase the capabilities of the Cloud-based Engine. Moreover, we are developing a new Reinforcement Learning technique for the generation of the manipulation configurations.

References

1. Goldfeder, C., Allen, P.K.: Data-driven grasping. *Auton. Robot.* **31**(1), 1–20 (2011)
2. Dang, H., Allen, P.K.: Learning grasp stability. In: *Proceedings of International Conference on Robotics and Automation (ICRA)*, pp. 2392–2397. Saint Paul, MN (2012)
3. Glover, J., Rus, D., Roy, N.: Probabilistic models of object geometry for grasp planning. In: *Proceedings of Robotics Science and Systems (RSS)*, Zurich, Switzerland (2008)
4. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source robot operating system. In: *ICRA Workshop on Open Source Software* (2009)
5. Mohanarajah, G., Hunziker, D., D’Andrea, R., Waibel, M.: Rapyuta: a cloud robotics platform. *IEEE Trans. Autom. Sci. Eng.* **12**(2), 481–493 (2015)
6. Cheng, J., Leng, C., Wu, J., Cui, H., Lu, H.: Fast and accurate image matching with cascade hashing for 3d reconstruction. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014)
7. Fan, Z.: Concise Pattern Learning for RDF Data Sets Interlinking. PhD thesis, University of Grenoble (2014)

8. Kaufman, L., Rousseeuw, P.J.: *Statistical Data Analysis Based on the L1-Norm and Related Methods*, chapter Clustering by means of Medoids. North-Holland, pp. 405–416 (1987)
9. Dubois, V., Quafafou, M.: Concept learning with approximation: rough version spaces. In: *Rough Sets and Current Trends in Computing: Proceedings of the Third International Conference (RSCTC)*, Malvern, PA, USA, pp. 239–246 (2002)
10. Goldfeder, C., Ciocarlie, M., Allen, P.: The Columbia grasp dataset. In: *Proceedings of International Conference on Robotics Automation (ICRA)*, Kobe, Japan, pp. 1710–1716 (2009)
11. Kasper, A., Xue, Z., Dillman, R.: The KIT object models database: an object model database for object recognition, localization and manipulation in service robotics. *Int. J. Robot. Res. (IJRR)* **31**(8), 927–934 (2012)
12. Ciocarlie, M., Hsiao, K., Jones, E., Chitta, S., Rusu, R., Sucas, I.: Towards reliable grasping and manipulation in household environments. In: *Proceedings of International Symposium on Experimental Robotics*, Delhi, India, pp. 1–12 (2010)
13. Tenorth, M., Beetz, M.: KnowRob: a knowledge processing infrastructure for cognition-enabled robots. *Int. J. Robot. Res. (IJRR)* **23**(5), 566–590 (2013)
14. Waibel, M., Beetz, M., Civera, J., D’Andrea, R., Elfring, J., Galvez-Lopez, D., Haussermann, K., Janssen, R., Montiel, J.M.M., Perzylo, A., Schiessle, B., Tenorth, M., Zweigle, O., van de Molengraft, R.: RoboEarth—a world wide web for robots. *IEEE Robot. Autom. Mag.* **18**(2), 69–82 (2011)
15. Di Marco, D., Koch, A., Zweigle, O., Haussermann, K., Schiessle, B., Levi, P., Galvez-Lopez, D., Riazuelo, L., Civera, J., Montiel, J.M.M., Tenorth, M., Perzylo, A., Waibel, M., Van de Molengraft, R.: Creating and using RoboEarth object models. *IEEE Int. Conf. Robot. Autom. (ICRA)* 3549–3550 (2012)
16. Prestes, E., Fiorini, S.R., Carbonera, J.: Core Ontology for Robotics and Automation. *Standardized Knowledge Representation and Ontologies for Robotics and Automation. Workshop on the 18th*, pp. 7–9. Illinois, USA, Chicago (2014)
17. Schlenoff, C., Prestes, E., Madhavan, R., Goncalves, P., Li, H., Balakirsky, S., Kramer, T., Miguelanez, E.: An IEEE standard ontology for robotics and automation. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1337–1342. Vilamoura, Algarve, Portugal (2012)
18. Schlenoff, C.I.: An overview of the IEEE ontology for robotics and automation (ORA) standardization effort. In: *Workshop on the 18th, Standardized Knowledge Representation and Ontologies for Robotics and Automation*, pp. 1–2. Illinois, USA, Chicago (2014)
19. Balakirsky, S., Kootbally, Z., T. RKramer, A. Pietromartire, C. Schlenoff, Gupta, S.: Knowledge driven robotics for kitting applications. *Robot. Auton. Syst.* **61**(11), 1205–1214 (2013)
20. Fiorini, S.R., Carbonera, J.L., Gonçalves, P., Jorge, V.A., Rey, V.F., Haidegger, T., Abel, M., Redfield, S.A., Balakirsky, S., Ragavan, V., Li, H., Schlenoff, C., Prestes, E.: Extensions to the core ontology for robotics and automation. *Robot. Comput.-Integr. Manuf.* **33**(C), 3–11 (2015)
21. Cutkosky, M.R.: On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Trans. Robot. Automa.* **5**(3), 269–279 (1989)
22. Beksi, W.J., Spruth, J., Papanikolopoulos, N.: Core: a cloud-based object recognition engine for robotics. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4512–4517 (2015)
23. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9–13 (2011)
24. Lowe, D.G.: Object recognition from local scale-invariant features. In: *Proceedings of the International Conference on Computer Vision-Volume 2–Volume 2, ICCV ’99*, pp. 1150–1157 (1999)
25. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: *11th European Conference on Computer Vision (ECCV)*, Hersonissos, Greece, September 5–11 (2010)
26. Bradski, G.: *Dr. Dobb’s Journal of Software Tools*
27. Miller, A.T., Allen, P.K.: Graspit! a versatile simulator for robotic grasping. *IEEE Robot. Autom. Mag.* **11**(4), 110–122 (2004). Dec

28. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS 2004), pp. 2149–2154 (2004)
29. Mörwald, T., Vincze, M.: Object Modelling for Cognitive Robotics. PhD Thesis, Vienna University of Technology (2013)

Tracking Control of Human-Following Robot with Sonar Sensors

Wei Peng, Jingchuan Wang and Weidong Chen

Abstract Human-following has become one of the most important functions as to human-friendly robots that are able to coexist with humans and serve humans. In this paper, we propose a tracking control strategy for a human-following robot which applies a sonar ring as its rangefinder to detect human and obstacles. In order to detect human, we equip the human with a guiding device. The tracking control strategy consists of firing strategy and human-following and obstacle-avoiding strategy. Firing strategy can reduce crosstalk effectively. Human-following and obstacle-avoiding strategy ensure that the robot can follow human in a constant following distance with an appropriate orientation and avoiding obstacles under unknown environments.

Keywords Human-following · Tracking control · Crosstalk · Guiding device · Obstacle-avoiding

1 Introduction

During the past decades, robotics have become an increasingly important part in ordinary people's lives, and have evolved from factory manufacturing to human's daily life. Their functions are becoming more diverse, including entertainment, assistance and keeping people from difficult or tedious work. This kind of robot is called service robot. In general, the term service robot is used to refer to robots that

W. Peng · J. Wang · W. Chen (✉)
Key Laboratory of System Control and Information Processing,
Department of Automation, Ministry of Education of China,
Shanghai Jiao Tong University, Shanghai 200240, China
e-mail: wdchen@sjtu.edu.cn

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_22

do services to humankind and manufacturing. Some service robots have already entered ordinary people's life. For example, the use of autonomous vacuum cleaners has grown enormously since the arrival of the Roomba [1]. Now, increasing researches have been developing this kind of robots to help people with their lives. One of the important field is human-following, which is also the basic function to realize other higher level assistance for service robot. Therefore, as to service robot, the control of human-following is essential.

In this paper, we develop a tracking control strategy for human-following robot. By the use of this tracking control strategy, the human-following robot is capable of following a person with a constant following distance and avoiding obstacles when necessary.

Concerning tracking and following robot, a lot of relevant previous works have been done. In [2], the authors conclude that laser and vision offer good performance in well illuminated circumstance, but here comes the problem, laser is limited in price and vision is easily affected by lighting; Therefore, sonar ToF (time of flight) offers the best robust tracking. However, a robot system is always apply multi-sonars, which would lead to crosstalk [3]. Borenstein J and Koren Y propose error eliminating rapid ultrasonic firing (EERUF) [4] to reduce crosstalk among sonars and achieve fine performance. Ming A, Kajihara K develop rapid obstacle sensing using sonar ring for mobile robot (ROUSM) to reject crosstalk. In [5–10], the authors only developed some methods to implement human detecting and tracking, but they didn't consider the characteristics of robot sensing ability. That means in real situation, the system may be poorly robust in terms of working efficiently. To develop a more robust tracking and following robot, some researches have been done. In [11], the authors designed a motion control model to optimize robot's velocity. The model constructs a relationship between the robot sensing ability and human's motion. But it must be provided with a map of the environment and is not able to take dynamic obstacles into account, which is not practical outdoor. In [12], a robot based on vision-detection and ultrasound obstacle avoidance is developed. But it can't restrain the undesired interface such as crosstalk which has adverse effect on the robot's performance. A method which has taken relative velocity between human and robot into consideration to calculate robot's velocity is developed in [13]. In this paper, a novel tracking control strategy is proposed to generate collision-free movement as well as follow human equipped with a guiding device under unknown environments.

In the following part, we present the hardware and software structure of the robot (Sect. 2). Then, the tracking control strategy is described in Sect. 3. The experiment and analysis are shown in Sect. 4. Finally, we conclude with some remarks on future research directions (Sect. 5).

2 Robot System Structure

2.1 Hardware Structure

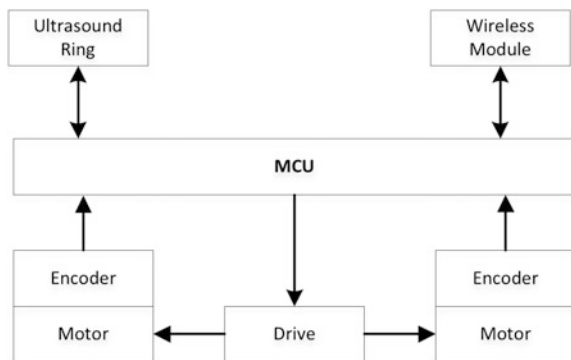
There are two drive motors on the base of the robot. Each drive motor has an encoder with it. The MCU used in this robot is STM32. A sonar ring, including nine ultrasound sensors which are evenly distributed within the scope of 180°, is used to human-detecting and obstacles-detecting. As shown in Fig. 3, the sonar ring can equip the robot with large scan range. Meanwhile, a wireless module is used to synchronize the emitting and receiving of ultrasound signals.

In order to detect human, a guiding device consisting of an ultrasound transmitter and a wireless module is developed to equip the human. During working, the guiding device takes use of its wireless module to emit synchronizing signal and use the ultrasound transmitter to emit ultrasound signals. If the robot’s wireless module receives the synchronizing signal, the robot will fires ultrasound receivers to receive ultrasound signals. If not, the robot will not begin to receive ultrasound signals. The hardware structure of robot is shown Fig. 1.

2.2 Software Structure

As shown in Fig. 2, the software structure of control system consists of three sectors: firing strategy, human-following and obstacle-avoiding strategy, and closed loop control of velocity. Firing strategy make the ultrasonic sensors fire by a unique time sequence, which can reduce the crosstalk effectively. Human-following and obstacle-avoiding strategy figures out collision-free motion command to follow human in constant following distance. The closed loop control of speed guarantees the robot run with the desired velocity. All details will be described in the next section (Fig. 3).

Fig. 1 Hardware structure



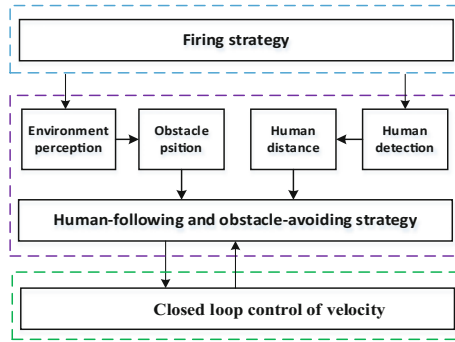


Fig. 2 Software structure

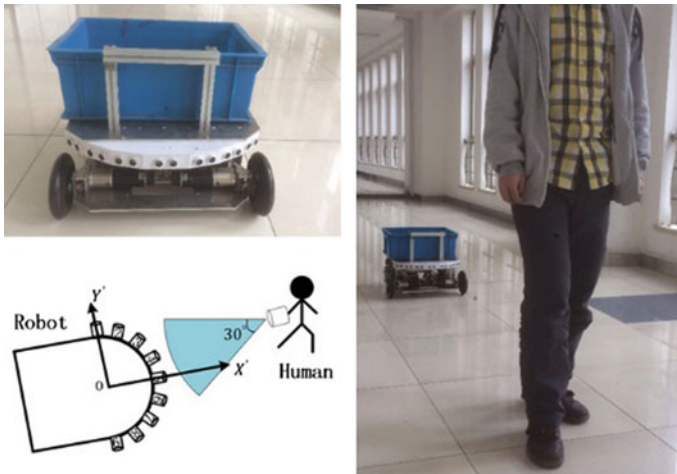


Fig. 3 Robot system

3 Tracking Control Strategy

3.1 Firing Strategy

Firstly, we define three types of ultrasonic sensors according to their functions. They are respectively $SG_a(a = 1 \sim M_1, M_1 \geq 1)$, the guiding ultrasonic sensor; $SR_b(b = 1 \sim M_2, M_2 \geq 2)$, the human-detecting sensor used to receive the signals emitted by SG_h ; $SE_c(c = 1 \sim M_3, M_3 \geq 1)$, the obstacle-detecting sensor which can detect obstacles. M_1, M_2 and M_3 are respectively the number of corresponding sensor. According to these definitions, it can be inferred that SR_b and SE_c belong to

sonar ring which is mounted on the robot, and SG_a is belong to guiding device worn on a human.

In order to detect human, we should select human-detecting sensors from sonar ring. We define P_i as the reference value of one ultrasonic sensor selected as human-detecting sensor. The larger P_i is, the more possible is that its corresponding sensor can be chosen as the human-detecting sensor. In the local coordinate, we define μ_i as the direction angle of sensor i with respect to the local X' axis, which is shown in Fig. 4. Since the sensors in the sonar ring are uniform distribution, we can take use of Eq. 1 to calculate μ_i .

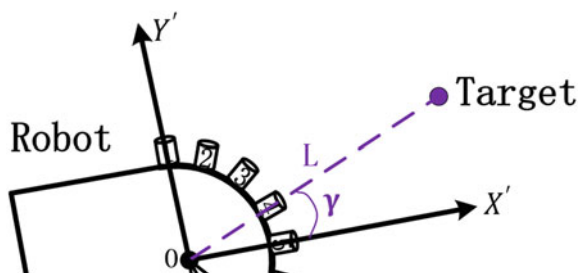
$$\mu_i = 90 - i * \Delta\mu \quad (i \leq N), \quad (1)$$

where $\Delta\mu$ is the angle between two ultrasonic sensors and N is the number of sensors.

The position of human is described as $q_T = (x_T, y_T)$ in generalized coordinate system, which can be transformed to (L, γ) in the polar coordinate. As shown in Fig. 4, L is the distance between human and robot and γ is the direction angle of human with respect to the local X' axis. It is reasonable to conclude that, for a certain ultrasonic sensor, when the deviation between μ_i and γ is smaller, the more possible is that this ultrasonic sensor can receive the ultrasound signals of guiding device. In addition, we define $\Delta P_{i,k}$ as the increase of reference value for sensor i in the k th control cycle, as formulated in Eq. 2:

$$\Delta P_{i,k} = \begin{cases} 3 & |\mu_i - \gamma| \leq \frac{\Delta\mu}{2} \\ 2 & \frac{\Delta\mu}{2} \leq |\mu_i - \gamma| \leq \Delta\mu \\ 1 & \Delta\mu \leq |\mu_i - \gamma| \leq \frac{3\Delta\mu}{2} \\ 0 & \frac{3\Delta\mu}{2} \leq |\mu_i - \gamma| \leq 2\Delta\mu \quad i \leq N, \\ -2 & 2\Delta\mu \leq |\mu_i - \gamma| \leq \frac{5\Delta\mu}{2} \\ -4 & \frac{5\Delta\mu}{2} \leq |\mu_i - \gamma| \leq 3\Delta\mu \\ -6 & |\mu_i - \gamma| \geq 3\Delta\mu \end{cases} \quad (2)$$

Fig. 4 Direction angle



Then the possibility-updating equation is:

$$P_{i,k} = \begin{cases} P_{up} & P_{i,k} > P_{up} \\ \epsilon * P_{i,k-1} + \Delta P_{i,k} & P_{i,k} \leq P_{up} \& \& P_{i,k} \geq P_{down} \\ P_{down} & P_{i,k} < P_{down} \end{cases} \quad (3)$$

where i is the index of sensor, and k indicates the k th control cycle, and ϵ is the forgetting factor which is always 0.9 and can prevent data saturation. According to $P_{i,k}$, we can choose the human-detecting sensor. This P_{up} and P_{down} are respectively the maximum and minimum reference value. The number of human-detecting sensor are usually three. Then the steps of determining human-detecting sensors is described as followings:

- Step (1) At the first control cycle, we let $P_{i,k}|_{k=0} = 0$ and assume $V_T(0) = 0$ at the next interval, and let $V_R(0) = 0$; then fire all the ultrasonic sensors to detect human and calculate the position $(L(k), \gamma(k))|_{k=0}$ through triangulation location. Apply $(L(k), \gamma(k))|_{k=0}$ to figure out $\Delta P_{i,k}|_{k=0}$ and update $P_{i,k}|_{k=1}$. Then, choose the sensors whose $P_{i,k}|_{k=1}$ are top M_2 as SR_b ($b = 1 \sim M_2$). Then, go to Step 2.
- Step (2) Make use of SR_b selected in step 1 to calculate the human position $(L(k), \gamma(k))|_{k=1}$, SE_c ($i = 1 \sim M_3, M_3 = N - M_2$) to detect obstacles, and then figure out the robot speed $V_R(t)$ through Motion Planning Strategy. The control frequency used in our robot is more than 10 Hz. Then, according to $(L(k), \gamma(k))|_{k=1}$, we can obtain the $\Delta P_{i,k}|_{k=1}$ and update the $P_{i,k}|_{k=2}$, and then select the human-detecting sensors.
- Step (3) Repeat Step 2 along with the k increasing.

Applying sonar ring would lead to occurrence of crosstalk, and crosstalk can be classified into three types below:

1. SE_c receive the signals emitted by SG_a , Fig. 5b;
2. SR_b receive the signal emitted by SE_c , Fig. 5c;
3. SE_c receive the signal emitted by SE_d , $c \neq d$, Fig. 5d.

The red sonars emit ultrasonic signals and the radial is the path of signals propagation. 1 and 3 would lead to the consequence that the robot cannot obtain accurate information of the obstacles, or even get false obstacles' information. 2 would lead to inaccurate calculation of the position of human.

In order to minimize the crosstalk among ultrasound sensors, we take use of EERUF method to control ultrasound sensors. We regard SG_a and SR_b as a virtual ultrasonic sensor- S_{track} . SG_a is the emitter part of S_{track} , and SR_b are the receiver part of S_{track} . SG_a sends the guidance ultrasound. SR_b receives ultrasonic signals, and never emits ultrasound. SR_b receives the ultrasound emitted by SG_a , just like SE_c receives his own echoes. Then, the types of ultrasonic sensors in the Tracking & Following Robot System become SE_c and S_{track} . Then, we can use EERUF to reject crosstalk in our system.

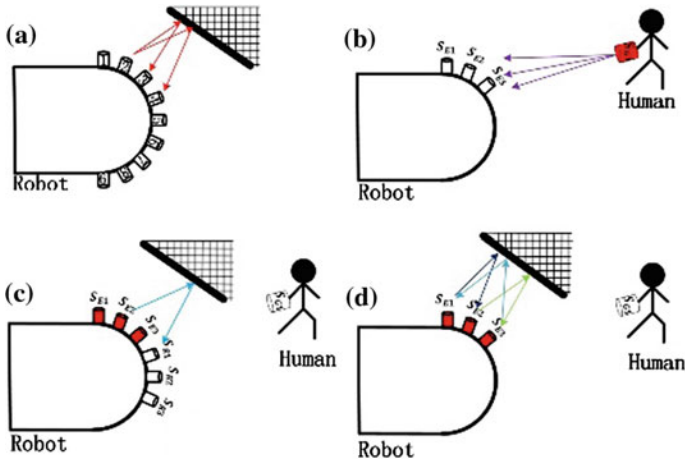
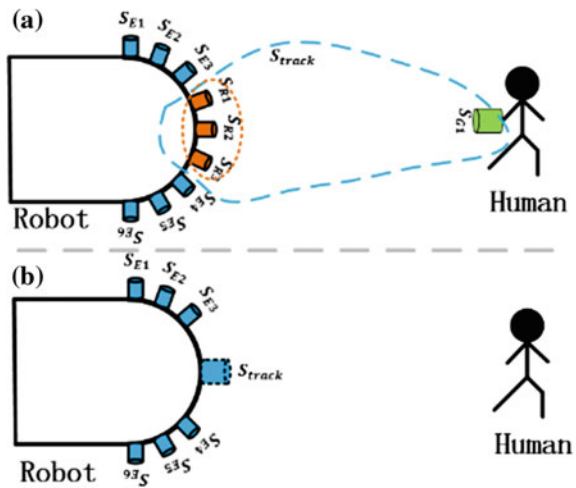


Fig. 5 Crosstalk in the robot

Fig. 6 The S_{track} of robot system



As shown in Fig. 6, we select $M_1 = 1$, $M_2 = 3$ and $N = 9$. We choose human-detecting sensor $SE_c(c = 1 \sim 3)$. Then, we can apply EERUF for SE_c and S_{track} . Under this circumstance, SR_b would receive ultrasound from SG_a and reject noise from SE_c ; meanwhile SE_c would obtain their own echoes and eliminate noise. Then we use EERUF to fire SE_c and S_{track} at a special time sequence. Referring to [4], you can get the detail steps of EERUF.

Compared with [4], the uniqueness of our work is as below: first, making use of sonar ring, we realized both human-detecting and obstacle-detecting functions, while [4] can only detect obstacles. Second, [4] just eliminate the crosstalk among SE_c , and we can eliminate the crosstalk among SG_a , SR_b and SE_c , which is far more complex and efficient than [4].

3.2 Human-Following and Obstacle-Avoiding Strategy

In this part, a novel human-following and obstacle-avoiding strategy is discussed. In each iteration, the robot applies triangulation location to acquire the position of human; then, the desired circle (DC) is determined by this position of the person and the following distance; the selectable path set (SP_{OB}) is determined by the positions of this human, robot and obstacles; the most feasible path is selected according to a weighted function. Finally, the velocity command of the robot at the next interval is figured out by a control law. The block diagram of human-following and obstacle-avoiding strategy is shown in Fig. 8.

Taking use of ultrasonic sensor, we can achieve the positions of obstacles and human in the local coordinate system. Once the obstacle's relative position $p_{OBi} = (x_{OBi}, y_{OBi})$ ($i \leq M_4, M_4$ is the number of sensor which detect obstacle) is obtained, we adopt the inflation model to enlarge the point p_{OBi} to the same size of following robot.

Human-locating

The human-detecting sensors SR_b perceive the signals emitted by the guiding ultrasonic sensor; and the robot can figure out the distances between SR_b and the guiding ultrasonic sensor by ToF. Then, the position of human in the local coordinate can be acquire by triangulation location.

Desired Circle

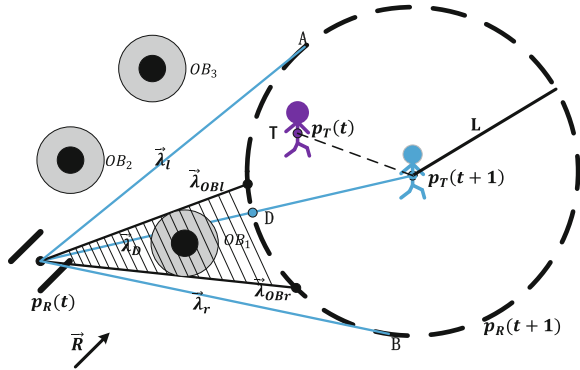
We innovate the concept of Desired Circle (DC) for target human. The position of human at the time of t is shown by $p_T(t) = (x_T(t), y_T(t))$. The velocity of human is $\vec{v}_T(t)$. Then, we can estimate the position of human at time $(t + 1)$, which can be denoted by $p_T(t + 1) = (x_T(t + 1), y_T(t + 1))$. The DC for human at time $(t + 1)$ is determined by drawing a circle around the $p_T(t + 1)$ with a radius of the following distance L . In order to follow the human at a constant distance L , the robot should arrive at one position on the DC with an appropriate orientation at time $(t + 1)$. Therefore, DC can be treated as the set of desired position of robot at time $(t + 1)$:

$$DC = \left\{ p_R(t + 1) \mid (x_R(t + 1) - x_T(t + 1))^2 + (y_R(t + 1) - y_T(t + 1))^2 = L^2 \right\},$$

where $p_R(t + 1) = (x_R(t + 1), y_R(t + 1))$ is the position of robot at time $(t + 1)$.

Because the current position of robot $p_R(t) = (x_R(t), y_R(t))$ is known, we can draw many vectors from $p_R(t)$ to DC. We define $\vec{\lambda}$ to represent this kind of vector. The length of $\vec{\lambda}$ is the distance from $p_R(t)$ to DC and the direction of $\vec{\lambda}$ is the angle with respect to global x axis. These vectors selectable paths of the robot during the next time intervals Δt . If there is an obstacle located on one certain vector, this path must be forbidden. As shown in Fig. 7, OB_1 is the inflated obstacle and $\vec{\lambda}_{OB_l}$ and $\vec{\lambda}_{OB_r}$ are tangent to OB_1 , so if one of the vectors between $\vec{\lambda}_{OB_l}$ and $\vec{\lambda}_{OB_r}$ is selected as robot's path, the robot would collide with OB_1 . $\vec{\lambda}_l$ and $\vec{\lambda}_r$ represent the two path tangents to the DC at point A and B respectively. The line between $p_R(t)$ and $p_T(t + 1)$ intersects DC at point D and $\vec{\lambda}_D$ is the path from $p_R(t)$ to point D.

Fig. 7 Desired circle and selectable path



Selectable Path Set

Under the condition of taking no account of obstacles, all line segments between $\vec{\lambda}_l$ and $\vec{\lambda}_r$ are selectable paths. We define the selectable path sets SP as following:

$$SP = \{ \vec{\lambda}_s | \langle \vec{\lambda}_s, \vec{\lambda}_l \rangle \leq \langle \vec{\lambda}_l, \vec{\lambda}_r \rangle \ \&\& \ \langle \vec{\lambda}_s, \vec{\lambda}_r \rangle \leq \langle \vec{\lambda}_l, \vec{\lambda}_r \rangle \},$$

where $\langle \vec{\lambda}_s, \vec{\lambda}_l \rangle$ is the angle between $\vec{\lambda}_s$ and $\vec{\lambda}_l$, the same to $\langle \vec{\lambda}_s, \vec{\lambda}_r \rangle$ and $\langle \vec{\lambda}_l, \vec{\lambda}_r \rangle$. From the Fig. 7, we can conclude that $\vec{\lambda}_D$ has the shortest length. Furthermore, if the robot moves along the $\vec{\lambda}_D$, the human will locate at the center of robot’s field of vision at time (t + 1). That is to say, the robot has the maximum visibility for the human. Therefor, $\vec{\lambda}_D$ is optimal path for robot under the conditions of regardless of obstacles.

When considering the obstacles, we define forbidden path set FP_{OB} for robot.

$$FP_{OB} = \{ \vec{\lambda}_f | \vec{\lambda}_f \in SP, \vec{\lambda}_f \cap OB_i \neq \emptyset \} \quad i \leq M_4,$$

in which OB_i represents the obstacle. Then the selectable path set becomes SP_{OB} :

$$SP_{OB} = \{ \vec{\lambda}_s | \vec{\lambda}_s \in SP, \vec{\lambda}_s \notin FP_{OB} \}.$$

Most Feasible Path

We define the $\vec{\lambda}_{t+1}$ as the most feasible path during the next interval time, and then there are two possible cases:

- (1) $\vec{\lambda}_D$ falls within the SP_{OB} ; that is, $\vec{\lambda}_D \in SP_{OB}$. Then the robot can select $\vec{\lambda}_D$ as the best feasible path during the Δt and avoid any obstacles.
- (2) $\vec{\lambda}_D$ falls within the FP_{OB} . Then the robot must abandon the $\vec{\lambda}_D$ and seek for the best feasible path. For this purpose, the robot should select the $\vec{\lambda}_{t+1}$ based on five criteria.

1. The $\vec{\lambda}_{t+1}$ must be admissible; that is, $\vec{\lambda}_{t+1} \in SP_{OB}$
2. The $\vec{\lambda}_{t+1}$ must have minimum angle of deviation from $\vec{\lambda}_D$.
3. The $\vec{\lambda}_{t+1}$ must have minimum angle of deviation from current direction of robot.
4. The direction of $\vec{\lambda}_{t+1}$ must have maximum angle of deviation from OB_i
5. The length of $\vec{\lambda}_{t+1}$ must be as short as possible.

Therefore, a weighted function for these five criteria is formulated in Eq. 4.

$$\vec{\lambda}_{t+1} = \operatorname{argmin}(w_1 \cdot \langle \vec{\lambda}_s, \vec{\lambda}_D \rangle + w_2 \cdot \langle \vec{\lambda}_s, \vec{R} \rangle - w_3 \cdot \langle \vec{\lambda}_s, \vec{\lambda}_{OB} \rangle + w_4 \cdot |\vec{\lambda}_s|),$$

$$\forall \vec{\lambda}_s \in SP_{OB}; w_1, w_2, w_3, w_4 > 0 \tag{4}$$

where $\vec{\lambda}_{OB}$ is the path tangent to OB_i and has the minimum deviation from $\vec{\lambda}_s$. \vec{R} is the unit vector in the front direction of robot's center line. $\langle \vec{\lambda}_s, \vec{\lambda}_D \rangle$ [13] is the angle between $\vec{\lambda}_s$ and $\vec{\lambda}_D$, $\langle \vec{\lambda}_s, \vec{\lambda} \rangle$ [13] is the angle between $\vec{\lambda}_s$ and \vec{R} .

The larger w_1 is, the more possible $\vec{\lambda}_{t+1}$ approaches $\vec{\lambda}_D$. w_2 affects the smoothness of the movement. w_3 ensures the safety of movement. w_4 drives the robot to select the shortest path to DC. These four parameters have a trade-off relationship.

Control Law

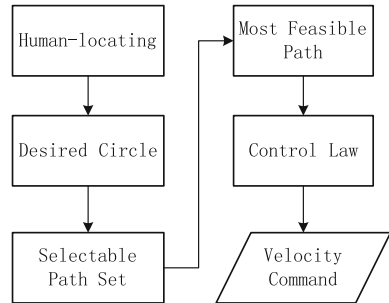
Once we calculate the most feasible path $\vec{\lambda}_{t+1}$, we can figure out the linear velocity and angular velocity of the robot, which are calculated as:

$$v_{t+1} = \min\left\{\frac{|\vec{\lambda}_{t+1}|}{\Delta t}, v_{max}\right\},$$

$$\omega_{t+1} = \min\left\{\frac{\langle \vec{\lambda}_{t+1}, \vec{R} \rangle}{\Delta t}, \omega_{max}\right\} \tag{5}$$

in which the v_{t+1} and ω_{t+1} is the linear velocity and angular velocity of robot respectively. v_{max} and ω_{max} are the maximum linear velocity and angular velocity of robot (Fig. 8).

Fig. 8 Human-following and obstacle-avoiding strategy



4 Experiment and Analysis

We test the human-following Robot System in two different environments, and set the obstacle course comprising of some boxes whose width are ranging from 200 mm to 1000 mm. The system includes nine ultrasound sensors ($N = 9$) and the control period of this system is 100 ms. The desired following distance $L = 1900$ mm, the max linear speed of robot $v_{max} = 1000$ mm/s, the max angular speed of robot $\omega_{max} = 1.5$ rad/s. Meanwhile, $M_1 = 1$, $M_2 = 3$, $M_3 = 6$. The positions of robot and human are acquired by VICON motion capture system. Experimental results are shown in Figs. 9 and 10. Figure 9 describes the trajectories of human and robot. Figure 10 shows the actual following distance between human and robot, desired following distance L , following error and the average of following error. The following error is the deviation between desired following distance and actual following distance.

The red curve is the trajectory of human in the global coordinate system, whose initial position is point A and terminal point is point C. The blue curve is the motion trajectory of robot, and its initial point and terminal point are point B and D respectively. The gray rectangles are some obstacles in the test environment.

Figure 9 shows the trajectories of the robot and human and the robot's trajectory from initial point to terminal point is continuous and smooth. In the beginning, OB_2 intersects with the line segment between point A and point B. That is to say, the robot would collide with OB_2 if the robot moves straightly from A to B. However, the consequence shows that the robot bypasses OB_2 , which is owed to the tracking control strategy. When the robot passes through the region between OB_3 and OB_4 , the robot selects the path which would make it avoid both OB_3 and OB_4 , and follow the human stably. Corresponding analyses are shown in Fig. 10. The following error is changing from -300 to 300 mm, which is an acceptable range.

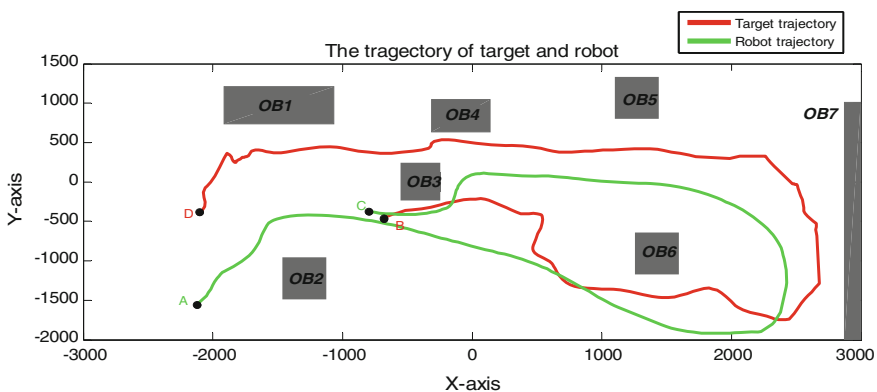


Fig. 9 Experiment of human-following and obstacle-avoiding

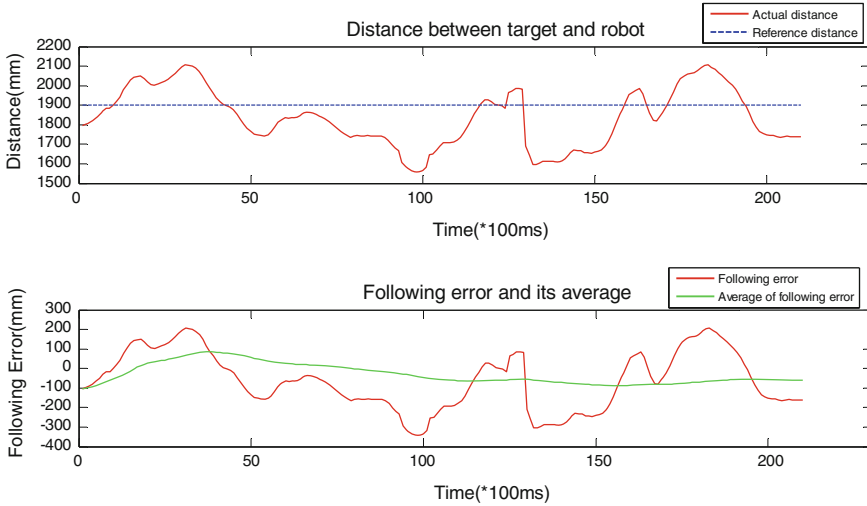


Fig. 10 Data analyses of experiment

The results and relevant analysis demonstrate that, by the use of the proposed tracking control strategy, the robot can follow the human and avoid obstacles robustly and smoothly through selecting appropriate path under different unknown environments.

5 Conclusion

This paper proposes a tracking control strategy for human-following robot. To make it practical in real life, ultrasound sensors are used to detect human and obstacles in the environment. The tracking control strategy includes firing strategy as well as human-following and obstacle-avoiding strategy. Firing strategy can effectively improve the ability of finding person reject crosstalk among ultrasound sensors. The human-following and obstacles-avoiding strategy is used to figure out human-following and collision-free motion command. The experiment verify that this tracking control strategy is robust and feasible. For future work, we plan to take use of this human-following robot to carry individual's luggage and packages for old person in supermarket or other places.

Acknowledgements This work is partially supported by the China Domestic Research Project for the International Thermonuclear Experimental Reactor (ITER) under Grant 2012GB102001, the Natural Science Foundation of China under Grants 61175088, 61221003 and 61573243.

References

1. Jones, J.L.: Robots at the tipping point: the road to iRobot Roomba. *Robot. Autom. Mag.* **13** (1), 76–78 (2006)
2. Sales Gil, J., Marín Prades, R., Cervera Mateu, E.: Multi-sensor person following in low-visibility scenarios. In: *Sensors* (2010)
3. Meng, Q., Yao, F., Wu, Y.: Review of crosstalk elimination methods for ultrasonic range systems in mobile robots. In: *IEEE/RJS International Conference on Intelligent Robots and Systems*, pp. 1164–1169 (2006)
4. Borenstein, J., Koren, Y.: Error eliminating rapid ultrasonic firing for mobile robot obstacle avoidance. *IEEE Trans. Robot. Autom.* **11**(1), 132–138 (1995)
5. Kwon, H., Yoon, Y., Park J.B., et al.: Person tracking with a mobile robot using two uncalibrated independently moving cameras. In: *IEEE International Conference on Robotics and Automation*, pp. 2877–2883 (2005)
6. Tsuda, N., Harimoto, S., Saitoh, T., et al.: Mobile robot with following function and autonomous return function. In: *International Conference on Control, Automation and Systems*, pp. 635–640 (2009)
7. Okusako, S., Sakane, S.: Human tracking with a mobile robot using a laser range-finder. *J. Robot. Soc. Jpn.* **24**(5), 43 (2006)
8. Zender, H., Jensfelt, P., Kruijff G.J.M.: Human-and situation-aware people following. In: *16th IEEE International Conference on Robot and Human interactive Communication*, pp. 1131–1136 (2007)
9. Gockley, R., Forlizzi, J., Simmons, R.: Natural person-following behavior for social robots. In: *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, pp. 17–24 (2007)
10. Nagumo, Y., Ohya, A.: Human following behavior of an autonomous mobile robot using light-emitting device. In: *10th IEEE International Workshop on Robot and Human Interactive Communication*, pp. 225–230 (2001)
11. Murrieta-Cid, R., Muñoz-Gomez, L., Alencastre-Miranda, M., et al.: Maintaining visibility of a moving holonomic target at a fixed distance with a non-holonomic robot. In: *IEEE/RJS International Conference on Intelligent Robots and Systems*, pp. 2687–2693 (2005)
12. Yoshimi, T., Nishiyama, M., Sonoura, T., et al.: Development of a person following robot with vision based target detection. In: *IEEE/RJS International Conference on Intelligent Robots and Systems*, pp. 5286–5291 (2006)
13. Masehian, E., Katebi, Y.: Sensor-based motion planning of wheeled mobile robots in unknown dynamic environments. *J. Intell. Robot. Syst.* **74**(3–4), 893–914 (2014)

Active Sensing for Human Activity Recognition by a Home Bio-monitoring Robot in a Home Living Environment

Keigo Nakahata, Enrique Dorrnzoro, Nevrez Imamoglu,
Masashi Sekine, Kahori Kita and Wenwei Yu

Abstract It has been shown that mobile robots could be a potential solution to home bio-monitoring for the elderly. Through our previous studies, a mobile robot system that is able to recognize daily living activities of a target person has been developed. However, in a home environment, there are several factors of uncertainty, such as confusion with surrounding objects, occlusion by furniture, etc. Thus, the features extracted could not guarantee the correct recognition. To solve the problem, we applied active sensing strategy to the robot, especially to the body contour based behavior recognition part, by implementing 3 algorithms in a row, which enabled (1) judging irregularity of feature extraction; (2) adjusting robot viewpoints accordingly; (3) avoiding excessive viewpoint adjustment based on a short-term memory mechanism, respectively. As a result of experiment in a home living scenario, higher activity recognition accuracy was achieved by the proposed active sensing algorithms.

Keywords Active sensing · Home bio-monitoring robot · Activities recognition

K. Nakahata (✉)

Department of Medical System Engineering, Chiba University, Chiba, Japan
e-mail: aeya0173@chiba-u.jp

E. Dorrnzoro (✉) · N. Imamoglu (✉) · M. Sekine (✉) · K. Kita (✉) · W. Yu (✉)

Center for Frontier Medical Engineering, Chiba University, Chiba, Japan
e-mail: enriquedorzu@gmail.com

N. Imamoglu

e-mail: nevrez@ieee.org

M. Sekine

e-mail: sekine@office.chiba-u.jp

K. Kita

e-mail: kkita@chiba-u.jp

W. Yu

e-mail: yuwill@faculty.chiba-u.jp

N. Imamoglu

National Institute of Advanced Industrial Science and Technology, Ibaraki, Japan

1 Introduction

Population of the world is aging. In countries like Japan, Germany or Italy more than 23 % of the population is over 65 years old. Population age rises, just as it does the demand of caregivers. As caregivers are getting overload the quality of their service is decreasing.

Therefore, new tools and techniques are necessary to cover the needs of the elderly, in order to increase their quality of life and reduce the needs of human caregivers.

This research work addressed home monitoring, which serves to not only ensure the safety, but also to know the rhythm and quality of home daily living. In the literature, three main approaches have been reported on elderly monitoring: Wearable systems, smart houses and robots [1]. Wearable systems may cause user discomfort which leads to discontinuous monitoring [2]. Smart houses usually require a large number of sensors which increases the cost and the complexity. Even more, blind spots and dead-angle due to the furniture layout may cause safety problem. With the use of a mobile robot the number of sensors can be reduced which will lead to a reduction of cost and deployment complexity, and most importantly, well controlled, the temporally and spatially seamless monitoring for safe home daily living.

The idea of using robot as a tool to provide support to the elderly is not new [3–5]. Recent research studies have focused on the use of mobile robots for these purposes. But in these solutions robots are companion or service robots, and only few had used them as a mobile bio-monitoring platform [4].

In our previous studies we developed a home bio-monitoring system to monitor the elderly and function impaired persons (MIPs) [5]. This robot is capable to track the subject and perform tasks such as observation, analysis, behavior recognition, etc. by using the location information of the target person, and the extracted body contour features [6, 7]. The system was evaluated with an excellent performance, more than 99.00 % detection and tracking accuracy in testing datasets [6] and an overall correct rate of our human activity recognition of 98.6–99.4 % [8]. However, the experiments were performed in a controlled scenario. The activities were carried out one by one at one place and repeatedly during a certain period of time.

In a home environment, there are various factors of uncertainty, such as confusion with surrounding objects, occlusion by furniture, difficulties of localization and movement control due to uneven terrain, and frequent partial furniture layout alternations, etc. Thus, the features extracted could not guarantee the correct recognition. Especially, frequently errors due to the interference of surrounding objects, resulted from the proximity of objects that share the same depth with the subject. Because of the nature of the body contour extraction algorithm, these objects are included as part of the subject contour which leads to a wrong activity recognition result.

In this paper we present a novel solution called active sensing, especially to the human contour based behavior recognition part, which aims at detecting the

occurrence of the problem, and dealing with the problem by changing the point of view of the robot.

The performance of the active sensing has been evaluated with a set of experiments.

This paper is organized as follows. Section 2 describes the active sensing method and the experiments performed to evaluate the performance of this improvement. In Sect. 3, we describe the experimental results and discussions, and finally, concluding remarks are stated in Sect. 4.

2 Methods

In this section we describe the active sensing method and the experiments performed to evaluate the performance of this improvement.

2.1 Active Sensing

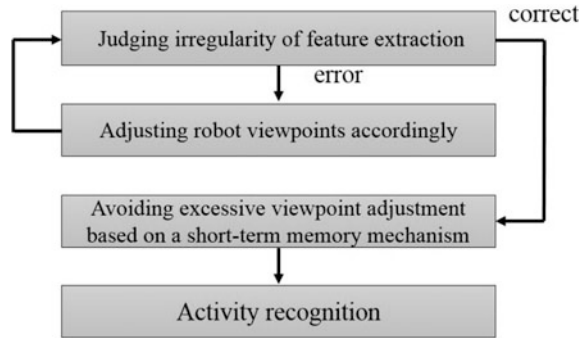
The active sensing is the ability to infer information under an uncertain environment by spontaneous sensing activity. It includes two categories. The first category includes the use of self-generated signals to probe the environment, such as echolocation chirps in bats [9]. Another one uses a self-motion, such as a move to find out an object hidden in the shadows. The active sensing mechanism implemented in the mobile robot is inspired by the latter.

In this study, the home mobile bio-monitoring robot performs tracking and activity recognition using a Kinect sensor which is mounted on a rotating platform making possible to be rotated to face designed direction. In other words, Kinect takes the role of the eye, the rotation table takes the role of the movement of the eye, and the mobile robot takes the role of the head and the foot of a human body. We used this analogy to develop an active sensing mechanism in the robot.

The current robot behavior does not consider those situations where an object that is located close to the subject may affect the activity recognition performance of the system. Therefore, moving the robot to a better position where the activity can be recognized with a higher success rate, will be an important step to solve this problem.

One control framework for active sensing by the human sight is C-DAC (Context-Dependent Active Controller). C-DAC assumes that the observer aims to optimize a context-sensitive objective function that takes into account behavioral costs such as temporal delay, response error, and the cost of switching from one sensing location to another [10]. This framework allows us to derive behaviorally optimal procedures for making decisions about where to acquire sensory inputs, when to move from one observation location to another, and how to negotiate the exploration-exploitation tradeoff between collecting additional data versus terminating the observation process [10].

Fig. 1 Flow of the active sensing



In addition, the human brain maintains information by short-term memory and pays attention to appropriate information about the aim and restrains trivial information and an excessive action.

When applying this to the home bio-monitoring robot, the behavioral cost is the time of changing the viewpoint, worsening accuracy of activity recognition. Three algorithms were performed in a row, (1) judging irregularity of feature extraction; (2) adjusting robot viewpoints accordingly; (3) avoiding excessive viewpoint adjustment based on a short-term memory mechanism so that the behavioral cost was minimized. The flow of the active sensing, with the 3 algorithms is shown in Fig. 1.

(1) Judging irregularity of feature extraction

In this paper, we assumed the case that a subject sits down close to an object at his same depth (Fig. 2). In this situation the human body contour extraction fails and part of the object is included the subject contour (Fig. 3).

To solve this problem, the first step is to identify when this situation is occurring. The algorithm is illustrated in detail in Fig. 4. Basically, the robot starts turning in a

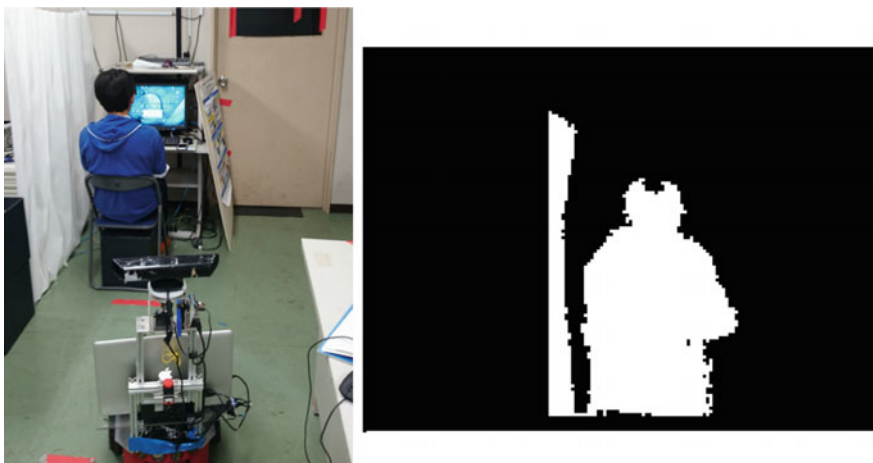


Fig. 2 Confusion with surrounding objects



Fig. 3 Suitable human body contour region when subject is sitting

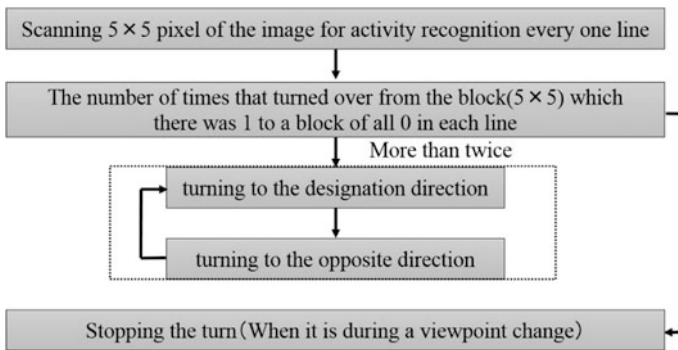


Fig. 4 Judging irregularity of feature extraction and adjusting robot viewpoints accordingly

direction. The image taken by the robot is scanned in blocks of 5×5 pixels with the aim of detecting more than one block of blocks per line where the pixels take the 0 value. When the image has one or none of this blocks per line, it is considered a suitable image for the body contour extraction and the robot stops turning.

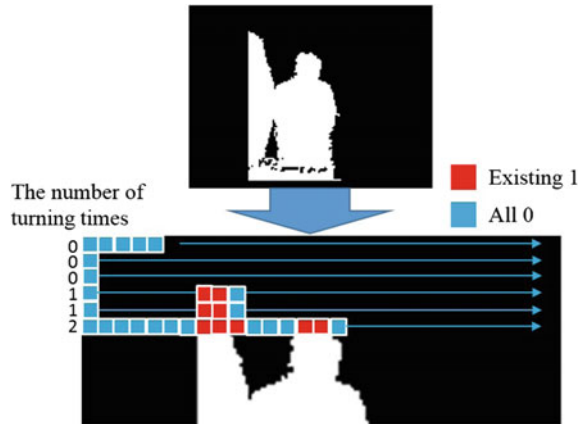
The human body contour region, used for activity recognition, is expressed in a binary image. In the situation illustrated in the Fig. 2, the binary image (right) the human body contour includes part of the wall located at the left side of the subject. In this scenario, a correct feature extraction is not possible and the robot fails in activity recognition.

Figure 3 presents a suitable contour where the wall interference does not appear (Fig. 5).

(2) Adjusting robot viewpoints accordingly

Once an irregular situation has been identified, the next step is to move the robot to a new position, where it will have a different viewpoint of the scene.

Fig. 5 How to judge irregularity of feature extraction



In order to optimize the active sensing operation time, instead of moving the robot to a different position we decided to apply a rotation movement. This rotation is done in both directions until it is able to identify that there is no irregularity in the body extraction process, using the method described in step 1.

(3) Avoiding excessive viewpoint adjustment based on a short-term memory

The robot may be confused by objects due to an error of Kinect and target person's minute movement, even if the robot stopped at a point of view that the robot can get a correct human body contour region. In that case, the robot would change its viewpoint frequently. In our algorithm (Fig. 6), the robot memorizes the position of the obstacle that causes incorrect activity recognition to avoid excessive viewpoint adjustment. By memorizing and setting a higher threshold for showing binary image in that area, the excessive viewpoint changing motion could be avoided.

For the evaluation of the performance of the active sensing improvement on the mobile robot we have performed two experiments.

2.2 Experiments

This section describes the methodology used for the two different experiments performed during this study.

2.2.1 Experiment 1, Controlled Scenario

The aim of this experiment is to evaluate the performance of the active sensing when an object is interfering in the body contour extraction process.

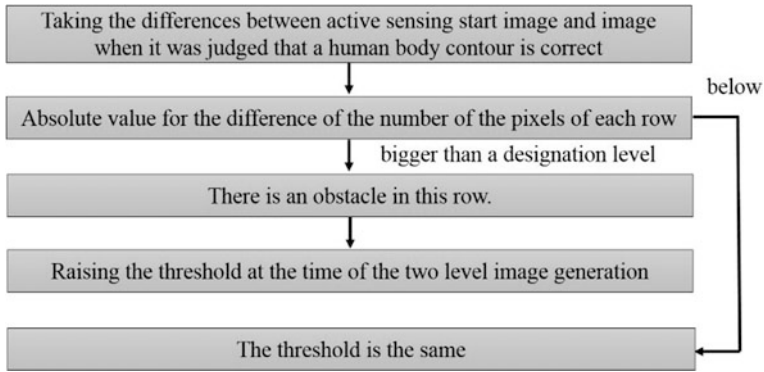
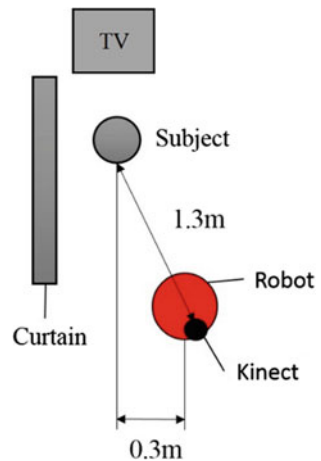


Fig. 6 Avoiding excessive viewpoint adjustment based on a short-term memory

Fig. 7 Each position relation in experiment 1



At this scenario the subject is sitting beside a partition curtain, as illustrated in Fig. 2. The robot was located (manually) in a position where the curtain interferes with the human body contour extraction (Fig. 7).

At this scenario we evaluated three different configurations, conventional system, active sensing with short-term memory and active sensing without short-term memory. In addition, we evaluated not only the accuracy but also the number of rotations that the robot required for a most suitable recognition. This measurement is needed to evaluate the effectiveness of the short-term memory function.

2.2.2 Experiment 2, Daily Life Activity Scenario

The aim of this experiment is to evaluate the active sensing performance in a flow of home living activities. In contrast with the previous experiment, the interference of a surrounding object is not ensured for all the activities.

The experiment had a total duration of 1 h and was performed by two subjects for the conventional system and by three subjects in the proposed one (active sensing with short-memory).

In this experiment, the robot tracks a subject moving in a home living environment. Therefore, the place where the robot stops is not always the same.

We set up two rooms as illustrated in Fig. 8. The subject moves in the scenario performing the following activities: Initially the subject arrives at home ①. The robot is waiting at the entrance and it starts tracking the subject. Then, the subject moves towards the kitchen and he washes his hands ②. He walks to the TV, takes a seat and watches TV ③. After watching TV for a while, he stands up and picks a drink from the fridge ④. When he finishes his drinking, he goes to the table and reads a newspaper ⑤. After reading the paper he moves to his desk and reads a book ⑥. Some minutes later the subject goes to ⑦, and begins to walk in an open area, as an exercise. When the exercise is finished he goes to the bed for sleep ⑧.

During the experiment the robot tracks the subject and identify the performed activity (standing, walking, sitting, bending, lying down) in real time and it also logs the information for further analysis. The full experiment is recorded with a video camera. Once the experiment is over we analyzed the log file and the video recording data to calculate the accuracy of the activity recognition for each activity.

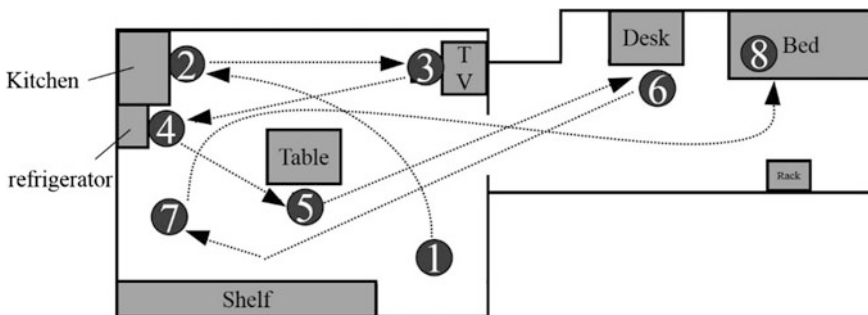


Fig. 8 Planned situations and sites

3 Results and Discussion

3.1 Experiment 1, Controlled Scenario

The results of Experiment 1 are summarized in Table 1. Accuracy of activity recognition by the conventional system was 12.4 %. In contrast, the system including the active sensing has an accuracy over 90 %. Furthermore, in the conventional system even there is an error due to the interference of the curtain the robot as the does not move when observing the problem persists while the subject does not change the performing activity. On the other hand, we can observe that, when active sensing is implemented, the robot moves to change the viewpoint as it detects the interference. Therefore, when activity recognition is uncertain, through the active sensing algorithm the robot (1) judges the irregularity of feature extraction and (2) Adjust the viewpoints accordingly to improve the accuracy.

On the other hand, without short-term memory the robot changed 30 times the viewpoint by rotating before been able to remove the interference, but it did just move once when short-term memory was enabled. As a result, it is proven that the algorithm to (3) avoid excessive viewpoint adjustment based on a short-term is more effective as it reduces the behavioral costs.

3.2 Experiment 2, Daily Life Activity Scenario

The comparison of recognition accuracy grouped by activity between proposed system and conventional system is shown in Fig. 9. All recognition accuracy except for walking seemed to improve by having introduced active sensing in Fig. 9. The accuracy of walking and standing activities is affected by the distance between the robot and the subject, so it is dependent of the subject trajectory and speed. This is why different accuracy results are obtained as active sensing has no impact in these activities. Other activities may also be affected by the position of the robot while observing the subject. However, in this paper we aim for the accuracy in the sitting

Table 1 The activity recognition result and the number of times of changing viewpoint by each systems

	Conventional system	Active sensing introduced system without short-term memory	Active sensing introduced system with short-term memory
Frame	500	500	500
Matched frame	62	464	492
Accuracy (%)	12.4	92.8	98.4
The number of times of changing viewpoint	0	30	1

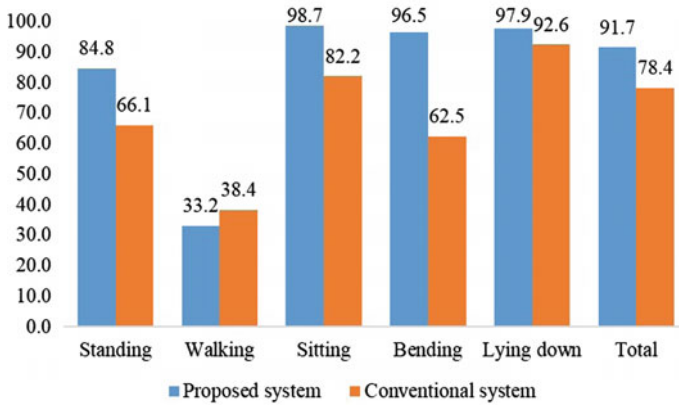


Fig. 9 Accuracy of recognition performed by proposed system and conventional system grouped by activity (%)

activity by using active sensing. Actually, during the experiment the active sensing mechanism was operating only in the situation, "③Watching TV". But, we understood that active sensing was not carried out at the time of the recognition of other activity accidentally and adverse effects were not given.

The activity recognition in ③ Watching TV was 70.6 % by the conventional system, however the activity recognition accuracy is improved up to 97.8 % by the proposed system.

This increment in the accuracy could be a result of the position of the robot during the experiments as it happened with other activities. Therefore, we analyzed the pose of the robot during this activity to find if active sensing was active.

Figure 10 shows the changes of the robot of the pose on the global map after the robot has stopped to observe the activity ③. In the conventional system, the robot did not move until the subject began to move again once the robot has stopped to

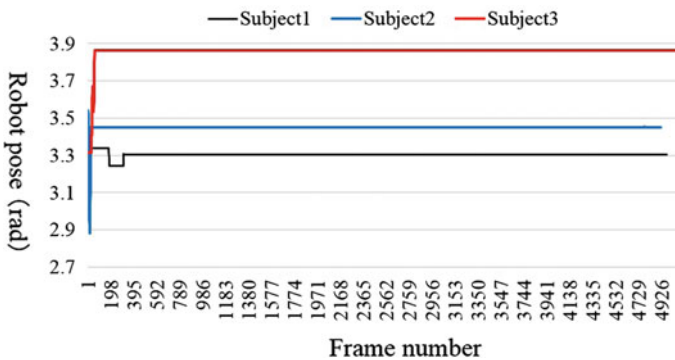


Fig. 10 Change of the robot pose on global map

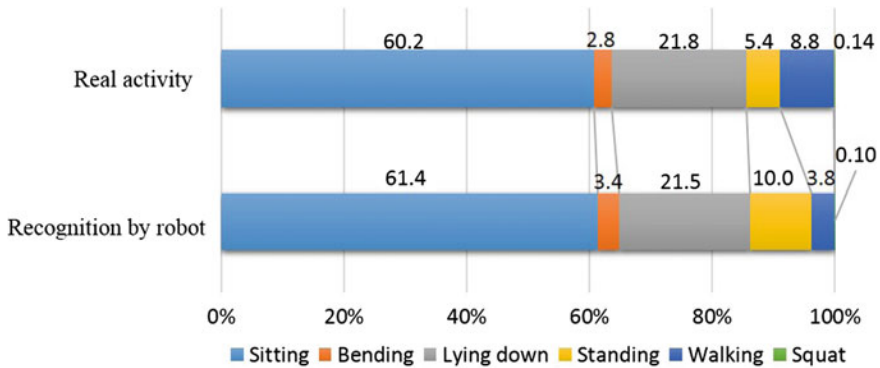


Fig. 11 Global ratio of the activities performed by the subject and recognized by proposed system (%)

observe the activity. Therefore, it means that active sensing is working for this activity during the experiments.

Thus, we understood that active sensing acted in a flow in home living effectively.

The upper part of Fig. 11 shows the ratio of action that the subject carried out by an experiment. The lower part shows the recognition result with the proposed system. Sitting, Bending, Lying down were about the same ratios for real activity and the one recognized by the robot. However, Walking and Standing showed a greatly different ratio.

The accuracy of Walking and Standing activities is low. However, we can consider Walking and Standing to be the same standing position. Therefore, the identification of the posture is possible for the actual system.

4 Conclusion

Our previous work presents the use of a robot for elderly and MIPs. The accuracy of the system in a controlled environment is over 98 %. However, in a home environment, there are various factors of uncertainty. One of them is the proximity of an object, located at a similar depth than the subject, which interfere with the activity recognition procedure, which reduces the accuracy.

In this paper we presented a novel algorithm based in active sensing. This algorithm is able to detect the interference of surrounding objects and change the robot viewpoint to eliminate this interference. During the experiments presented in this paper the algorithm has proven to improve the accuracy of the system when this interference is present.

We have also avoided the excessive viewpoint adjustment using short-term memory. During the execution of experiment 1, the viewpoint adjustment has been drastically reduced from 30 to 1.

In our future works would be to resolve other uncertain situations, such as occlusion by furniture, difficulties of localization and movement control due to uneven terrain and frequent partial furniture layout alternations, etc.

References

1. Sabelli, A.M., Kanda, T., Hagita, N.: Human-Robot Interaction (HRI). In: 2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 37–44 (2011)
2. Fasola, J., Mataric, M.J.: Using socially assistive human-robot interaction to motivate physical exercise for older adults. *Proc. IEEE* **100**(8), 2512–2526 (2012)
3. Huete, A.J., Victores, J.G., Martinez, S., Gimenez, A., Balaguer, C.: Personal autonomy rehabilitation in home environments by a portable assistive robot. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **42**(4), 561–570 (2012)
4. Bedaf, S., Gelderblom, G.J., De Witte, L.: Overview and categorization of robots supporting independent living of elderly people: what activities do they support and how far have they developed. *Assist. Technol.* **27**(2), 88–100 (2015)
5. Myagmarbayar, N., Yuki, Y., Imamoglu, N., Gonzalez, J., Otake, M., Yu, W.: Human body contour data based activity recognition. *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.* **2013**, 5634–7 (2013)
6. Imamoglu, N., Dorrnzoro, E., Sekine, M., Kita, K., Yu, W.: Top-down spatial attention for visual search: novelty detection-tracking using spatial memory with a mobile robot. *Adv. Image Video Process.* **2**(5) (2014)
7. Imamoglu, N., Dorrnzoro, E., Wei, Z., Shi, H., Sekine, M., González, J., Gu, D., Chen, W., Yu, W.: Development of robust behaviour recognition for an at-home biomonitoring robot with assistance of subject localization and enhanced visual tracking. *Sci. World J.* **2014**, 280207 (2014)
8. Nergui, M., Yoshida, Y., Imamoglu, N., Gonzalez, J., Otake, M., Yu, W.: Human activity recognition using body contour parameters extracted from depth images. *J. Med. Imaging Health Inf.* **3**(3), 455–461 (2013)
9. Moss, C.F., Surlykke, A.: Auditory scene analysis by echolocation in bats. *J. Acoust. Soc. Am.* **110**, 2207–2226 (2001)
10. Ahmad, S., Huang, H., Yu, A.J.: Cost-sensitive Bayesian control policy in human active sensing. *Front. Hum. Neurosci.* (2014)

Part VI
Legged Robot

An Underactuated Biped Robot Guided via Elastic Elements: EKF-Based Estimation of Ankle Mechanical Parameters

Roberto Bortoletto, Thomas Reilly, Enrico Pagello and Davide Piovesan

Abstract When studying humanoid robots, many of the processes involving the synthesis of robots motion have much in common with problems found in biomechanics and human motor-control research. Based on the use of simulation tools, additive manufacturing and system identification techniques, this work presents the design and implementation of an underactuated biped robot guided via elastic elements. In particular, the focus was on the analysis of stable posture maintenance during standing obtained on a completely passive humanoid robotic system. An estimation of the mechanical parameters of the elastic mechanical network was performed using an Extended Kalman Filter. The results provided here are part of a larger project that aims to implement a hybrid robotic system where the balance of bipeds are obtained using passive elastic elements, thus simplifying the control of gait. This work contributes to the research and development of increasingly efficient human-like robots.

Keywords Humanoid robot · Elastic elements · Extended Kalman filter · Mechanical parameters

R. Bortoletto (✉) · E. Pagello
Intelligent Autonomous Systems Laboratory (IAS-Lab),
Department of Information Engineering, University of Padova,
Via G. Gradenigo, 6/b, 35131 Padova, Italy
e-mail: bortolet@dei.unipd.it

E. Pagello
e-mail: epv@dei.unipd.it

T. Reilly · D. Piovesan
Biomedical Program, Mechanical Engineering Department, Gannon University,
109 University Square, PMB #3251, Erie, PA 16541, USA
e-mail: thomas.reilly@gannon.edu

D. Piovesan
e-mail: piovesan001@gannon.edu

1 Introduction

The synthesis of human motion is a complex procedure that includes accurate reconstruction of movement sequences; modeling of musculoskeletal kinematics, dynamics and actuation as well as the description of reliable performance criteria. Many of these processes are directly applicable to robotics research, with the advent of complex humanoid systems [1]. It is clear that these human-like robots are very different than today's industrial robots and their capability to generate coordinated natural motion has proven to be a challenging task. The characterization and control of humanoid systems can provide the support to understand biological functions of the human body (i.e., biomechanics), as well as tools to design machines and spaces where humans operate (i.e., ergonomics), and simulate environments to study the effects of musculoskeletal impairment in order to improve the design of rehabilitation systems.

When considering bipedal locomotion, three main types of actuation approaches can be distinguished: (a) *Active Walking* (i.e., fully actuated robots) which allows for the execution of tasks with reasonable speed and position accuracy at the cost of high control efforts, low efficiencies and often unnatural gaits [2–4]; (b) *Passive Dynamic Walking* (i.e., completely passive robots) which demonstrates extremely natural walking motions without the need of any control action [5, 6]; (c) the coupling of the above approaches in order to maximize the efficiency of locomotion [7, 8]. Through these major control approaches, researchers have increasingly focused on actuators with adaptable compliance [9, 10] that can change joint stiffness in order to adjust the overall leg properties with respect to robustness, energy efficiency and speed of motion [11–14]. In particular, the use of springs in legged locomotion has been accepted and promoted mostly with the introduction of the *Series Elastic Actuators* (SEA) [15].

In this scenario, understanding the principles of human movement is becoming increasingly important in order to design and develop more sophisticated bio-inspired robots and motion controllers [16, 17]. The use of multi-body dynamics simulation environments allows investigating the cause-effect relationships between human neuromuscular excitation patterns, muscle forces and motions of the body, as well as testing multiple design solutions during the development and implementation of a bio-inspired robot.

This paper focuses on the assessment of the robot ability to maintain a stable standing posture, using a completely passive actuation structure (i.e., without any motors). In this regard, a hybrid non-linear Extended Kalman Filter (EKF) based technique is proposed. The EKF is widely used in many engineering fields to quantify the state of linear systems encompassing quickness in the estimation and quasi-optimal results. Here, an EKF with augmented states was utilized. The perturbation to the humanoid was delivered using a base-excitation. Among the advantages of EKF-based methods is that the excitation does not need to be a canonical input but can be delivered using an arbitrary input [18]. This method allows for fast estima-

tions on a single trial, offering the potential to estimate joint stiffness during walking when proper robot actuation is implemented.

The estimates obtained from numerical simulations show a high correspondence with the experimental measures to guarantee the effectiveness of the adopted methods. This work lays the basis for the combination of a hybrid control systems where the actuation of the gait can be mostly concerned with the movement of the legs outsourcing the control of the standing posture to a simpler passive mechanical network.

2 Design

The proposed assembly is a bio-inspired four-segmented underactuated biped robot, originally inspired to the *Jena Walker II* [19], and *BioBiped R1* [20]. The total mass of the implement is about 2 kg and its height is about 40 cm. Previous studies have implemented the apparatus in simulation [21] developing a prototype for simulating human walking [22] and testing an impedance-based control approach for driving the mechanism to imitate human gait [23]. In this work, the design and prototype was improved in order to make a robot capable to maintain a quiet standing posture using only a completely passive actuation structure. Furthermore, the reconstruction of the continuous-time state space parameters proper of the assembly's control of quiet standing was also investigated. The reconstruction is based on a hybrid non-linear Extended Kalman Filter (EKF) to combine a base-excited inverted pendulum kinematic model of the robot with the discrete-time position measurements.

The position of the robot Center of Mass (COM) was measured using the Tracker Video Analysis and Modeling Tool¹ from a web-camera recording [24]. The modeling and simulation phases were conducted using OpenSim² and SimWise 4D.³ The former is a freely available, user extensible software system that lets users develop models of human musculoskeletal structures and create dynamic simulations of movement [25]. The latter is a proprietary solution for design and engineering professionals developing products involving assemblies of 3D parts. The two multi-body dynamics simulation environments were used to define an effective human-like skeletal structure, and to determine the spring stiffness positioning and natural lengths adopting a top-down heuristic approach [26].

¹<https://www.cabrillo.edu/~dbrown/tracker/>.

²<https://simtk.org/home/opensim>.

³<https://www.design-simulation.com/SimWise4d/index.php>.

2.1 Skeletal Structure

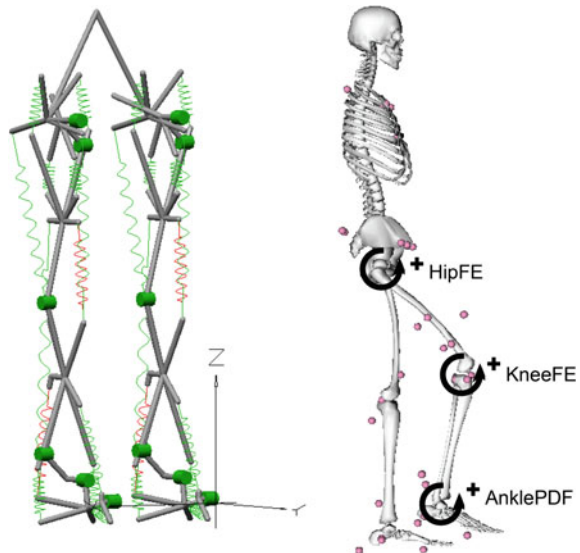
The simulated model and the corresponding prototype comprise of four segments for each leg: thigh, shank, foot and toes, plus one body segment called waist, which represents with its COM and dimensions the upper part of the skeleton (Table 1). Each leg has a total of 6 Degrees Of Freedom (DOFs) (Fig. 1):

- *Hip Flexion-Extension* (HipFE) and *Hip Adduction-Abduction* (HipAA) implemented by a universal joint;
- *Knee Flexion-Extension* (KneeFE) implemented via a single DOF revolute joint allowing the rotation of the shank with respect to the thigh;

Table 1 Volume, mass and dimensions of the principal robot bodies. Hip and Ankle represent the blocks that interconnect the waist to the thighs and the shanks to the feet, respectively

Body	Vol. [mm ³]	Weight [kg]	Dim. [mm]		
			x	y	z
Waist	2.22 e+5	0.231	150	102	99
Hip	5.89 e+4	0.061	42	67	81
Thigh	7.66 e+4	0.079	54	150	42
Shank	7.08 e+4	0.074	54	142	54
Ankle	3.28 e+4	0.034	42	50	41
Foot	2.81 e+4	0.029	48	104	46
Toe	1.61 e+4	0.019	50	48	27

Fig. 1 Kinematic structure of the robot (i.e., available DOFs, green cylinders). On the right hand side, the corresponding tree major DOFs along the sagittal plane are shown in the adopted human model



- *Ankle Plantar-Dorsi Flexion* (AnklePDF) and *Ankle Inversion-Eversion* (AnkleIE) implemented coupling two single DOF revolute joints;
- *Toe Flexion-Extension* (ToeFE) implemented via a single rotational DOF. Needed to accommodate the movement of the foot with respect to the ground.

2.2 Elastic Actuation Structure

The assembly is completely passive (i.e., without any motors). Eleven linear springs (Table 2) were inserted according to the position of the major muscle groups as depicted in Fig. 2a.

A detailed description of the top-down heuristic approach adopted for properly choose and positioning each spring is available in [26]. Here, we highlight the rela-

Table 2 ID, path (i.e., attachment and insertion points), and role of each spring. The initial guess made (PRE) on lengths [mm] and stiffness coefficients [N/mm], and the corresponding adjusted values after the prototyping phase (POST)

ID	Path	Role	PRE		POST	
			Length [mm]	K [N/mm]	Length [mm]	K [N/mm]
1	Ankle Bracket—Shank	Pulls shank towards toe-end of foot	69.8	0.222	63.5	0.305
2	Toe—Ankle Bracket	Pulls toe downward into ground	25.4	0.228	25.4	0.228
3	Shank—Ankle Bracket	Balances shank to sides of foot	38.1	0.525	38.1	0.525
4	Ankle Bracket—Shank	Pulls shank towards heel-end of foot	88.9	0.306	69.9	0.223
5	Thigh—Ankle Bracket	Pulls thigh towards heel-end of foot	222	0.715	127.0	0.715
6	Shank—Hip Bracket	Pulls shank to front-end of hips	152	0.441	114.3	0.441
7	Shank—Thigh	Pulls thigh forward over shank	76.2	0.543	76.2	0.543
8	Shank—Hip Bracket	Pulls shank to rear-end of hips	167	0.072	152.4	0.072
9	Thigh—Hip Swing Arm	Pulls thigh to rear-end of hips	25.4	0.531	25.4	0.531
10	Thigh—Hip Swing Arm	Pulls thigh to front-end of hips	25.4	0.531	25.4	0.531
11	Hip Bracket—Hip Swing Arm	Stabilizes side motion of hips and thighs	38.1	0.525	38.1	0.525

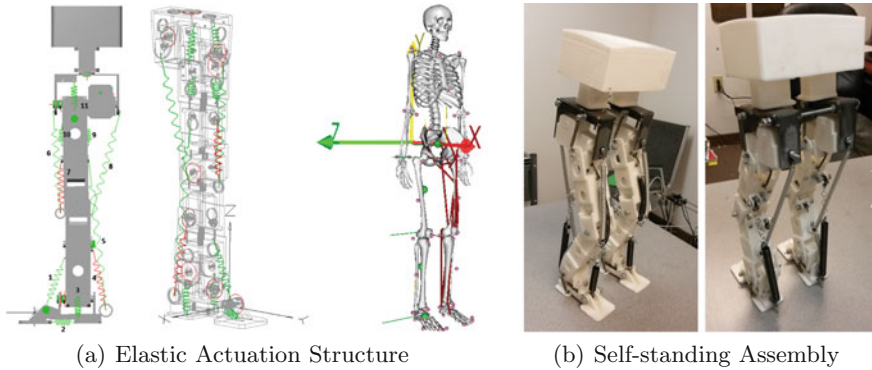


Fig. 2 **a** The elastic actuation structure, spanning the hip, knee and ankle joints, and the corresponding major human muscle groups, including from the *top* to the *bottom*: gluteus, psoas, bicep femoris and rectus femoris, vasti, gastrocnemius, soleus and tibialis anterior. **b** *Front* and *rear* views of the self-standing assembly

tionship between the spring sets and the corresponding human muscle groups. Calling the spring as the corresponding muscle groups, the assembly comprises of eight mono-articular springs (i.e., muscle groups represented by springs crossing only one joint: psoas, gluteus, vasti, soleus, and tibialis anterior), and three bi-articular springs (i.e., muscle groups represented by springs crossing two joints: rectus femoris, bicep femoris, and gastrocnemius).

3 Prototyping

Each body part was fabricated using a plastic-extrusion type rapid prototyping machine. This method was chosen for its relatively quick turn-around, and because most of the links were hollow (Fig. 2b).

The joints of the robot were implemented using steel screws. The attachment for the spring were screwed directly into the link, which are made of PLA plastic. The spring attachments were custom made for easily adjusting the pretensioning of the springs in order to obtain the proper equilibrium point of the structure [26]. Since the strength of PLA is generally low and the bracket connecting the hip with the waist is required to remain thin, this piece was fabricated using a 12-gage metal sheet. Changing the material to increase strength was necessary due to the high force generated by one of the strongest springs attached on the bracket simulating the vastus medialis. The material was chosen for its low cost and easiness to be fabricated by press-bracketing. Additional strengthening at the foot was provided with an aluminum insert so to attach the gastrocnemius. Alluminum was chosen for its favorable mechanical properties and the easiness to be machined within our facility compared to steel.

4 Viscoelastic Properties Estimation

The impedance of a mechanical system is often modelled as a second order system [27]. Using such a model requires computing the first and second derivatives (i.e., $\dot{\theta}$, and $\ddot{\theta}$, respectively) of the joint angles (i.e., θ). Here, the identification of the stiffness and damping parameters of the assembly was investigated using a time series analysis of the measurement by means of an EKF [28].

As a first approximation, the assembly system can be modeled as a base-excited inverted pendulum with one DOF (i.e., the rotation around the ankle joint), which can be represented as follows:

$$I\ddot{\theta} + b\dot{\theta} + k\theta = mgh \cdot \sin(\theta) + u + w_s \tag{1}$$

where for small displacement $\sin(\theta) \cong \theta$, yielding:

$$I\ddot{\theta} + b\dot{\theta} + (k - mgh)\theta = u + w_s \tag{2}$$

where I is the inertia of the whole system with respect to the ankle, b is the damping coefficient, k is the stiffness coefficient, m is the mass of the assembly, h is the distance between waist and ankle COMs, and g is the acceleration of gravity. $u = b\dot{\phi} + k\phi$, is the torque at the base due to a horizontal displacement d , where $\tan(\phi) = d/h \cong \phi$. w_s is the uncertainties of the system modeled as a Gaussian noise.

The corresponding augmented state space model can be defined by adding a number of states equal to the number of parameters that must be estimated (i.e., the damping coefficient, b , and the stiffness coefficient, k), so that the states are defined as follows:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ b + w_B \\ K_S + w_{K_S} \end{bmatrix} \tag{3}$$

where $K_S = k - mgh$. The terms w_B and w_{K_S} are artificial noise terms that must be added to the system for each of the desired parameters in order to allow the Kalman filter to modify their estimates [29]. Hence, the EKF can be applied as follows:

$$EKF = \begin{cases} \dot{x} = f(x, w, u, t) \\ y = h(x, v) \\ w \sim (0, Q) \\ v \sim (0, R) \end{cases} \tag{4}$$

where w and v are random noise of the augmented system x , and the measurements vector y respectively, Q and R are the corresponding covariance matrices. The subscript indicates the j -th discrete experimental measure. Given Eq. 3, the augmented

state space system is defined by the following function:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{1}{I} (w_B - x_3 x_2 - x_4 x_1) \\ w_B \\ w_{K_S} \end{bmatrix} \quad (5)$$

The filter was initialized as follows:

$$EKF_{init} = \begin{cases} \hat{x}_0 = E[x_0] \\ P_0 = E[(x_x - \hat{x}_0)(x_x - \hat{x}_0)^T] \end{cases} \quad (6)$$

where x_0 and P_0 represent the initial state and the covariance, respectively. The prediction phase was implemented as:

$$EKF_{pred} = \begin{cases} \hat{x} = f(x(t_j), u_j) \\ P_j = F_{j-1} P_{j-1} F_{j-1}^T + Q_{j-1} \end{cases} \quad (7)$$

and the update step as follows:

$$EKF_{update} = \begin{cases} G_j = P_j H_j^T (H_j P_j H_j^T + R)^{-1} \\ \hat{x}_j = \hat{x}_j + G_j (y_j - h(\hat{x}_j)) \\ P_j = (I - G_j H_j) P_j \end{cases} \quad (8)$$

where F and H are the Jacobian matrices of functions f and h , respectively. G represents the Kalman Gain, indirectly computed from the noise. Position and velocity information were corrupted imposing a Gaussian noise $N(0, 0.01)$.

Finally, the EKF was tuned using the following parameters, based on the simulation design process:

$$\begin{aligned} I_h &= mh^2 = 0.2672 \text{ kg m}^2/\text{rad} \\ m &= 1.67 \text{ kg} \\ h &= 0.4 \text{ m} \\ g &= 9.81 \text{ m/s}^2 \\ k &= 10 \text{ Nm/rad} \\ K_S &= 3.45 \\ b &= \sqrt{I_h k_s} = 0.96 \text{ Ns/m} \end{aligned} \quad (9)$$

with initial state equal to $x_0 = [0.2 \ 0 \ b \ K_S]^T$, P_0 identity matrix, Q diagonal matrix with elements equal to 0.01 and $R = 0.001$.

5 Evaluation and Results

The virtual model of the robot was tested within SimWise 4D by sliding the ground of 25 mm, in order to observe the waist angular displacement with respect to the ankle joint.

Figure 3a, shows a representation of the experiment and the obtained results. The waist displacement ranges from -0.19 to -0.08 rad at its maximum, and reaches the minimum displacement of 0.0044 rad in 10.2 s. The input perturbation and the corresponding waist displacement were then used within the previously described EKF in order to estimate the viscoelastic properties of the ankle joint. In particular, the last plot in Fig. 3a shows the obtained damping [Ns/m] and stiffness [Nm/rad] coefficients, which appear to be underestimated relative to the reference profiles obtained

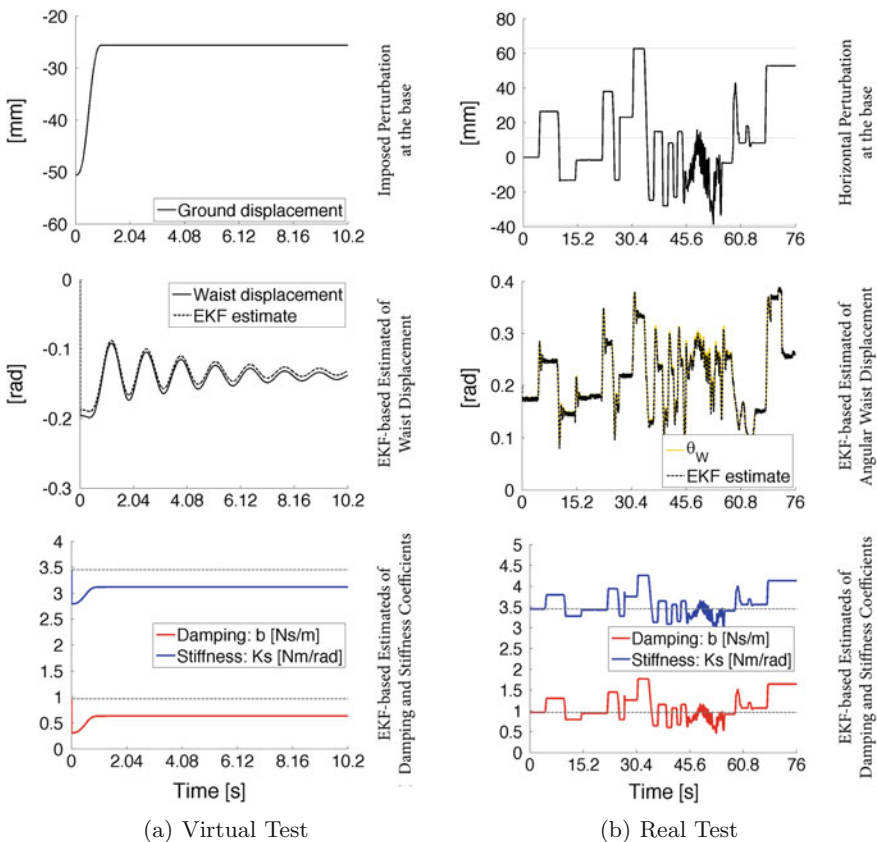


Fig. 3 From the *top* to the *bottom*: the imposed displacement at the base [mm], the recorded waist movement [rad], and the corresponding EKF-based estimates of damping and stiffness coefficients. The *horizontal dotted gray lines* represents the damping and stiffness reference values

from Eq. 9 that are 0.96 Ns/m and 3.45 Nm/rad, respectively. The waist displacement was tracked with a Root-Mean-Square Error (RMSE) of 0.0074 rad.

A second evaluation was conducted on the real robot, by placing it on a mobile base and moving it as in the same fashion of the simulation. Figure 3b shows the horizontal perturbation of the base [mm], the corresponding EKF-based estimates of the waist angular displacement [rad] computed with respect to the ankle, and finally the obtained damping [Ns/m] and stiffness [Nm/rad] coefficients. The RMSE between tracked angle and estimates was of 0.0052 rad (i.e., about 0.3°). In this case, both coefficients move around the reference values. The horizontal sections, visible in both profiles, correspond to a joint angle $\theta = 0$. It should be noted that while the stiffness of the springs around the ankle are constant, the moment arm of the spring changes as function of θ and therefore a change in the stiffness is to be expected. Furthermore, there exist a slight play of the ankle mechanism where the assembly does not return to $\theta = 0$ after every perturbation, therefore producing a bias in the estimation of stiffness and damping.

6 Conclusions

This paper investigates the mechanical parameters estimation of a self standing bipedal robot that was fabricated in our laboratory. It is worth noting that the system is not a simple inverted pendulum but has many more DOFs. The system was carefully designed so that the combination of springs spanning multiple joints acts as if the system was a global elastic structure. Stability is not guaranteed in such a type of system because the knees could buckle and the change in the moment arm at the spring that cross the knee could directly impact the stiffness at the ankle.

To date very few robotic devices implement an elastic mechanical-network. Most of these devices are actuated by SEA [30]. The control of SEA can be difficult because the elastic elements positioned in series introduce a delay in the actuation that the control needs to compensate for. This problem can be solved by a predictive control strategy (e.g., Smith predictor [31]) given an accurate knowledge of the plant. This work is part of a larger project that aims to propose a control architecture using *Parallel Elastic Actuators* [32]. Thus, the maintenance of posture and the control of gait can be achieved via two separate control architectures.

In this installment, we estimated the mechanical properties of the passive elastic network described in [26] allowing the bipeds to stand and compensate for external perturbation. The robot was excited at its base and the stiffness and damping at the ankle was estimated using an EKF. The methodology utilized here is similar to the one used for the estimation of the same physiologically relevant parameters in humans [28]. The EKF is ideally suited for this operation because it is capable of handling both noisy measurements of input and output, as well as modeling errors of the mechanical structure for fast parameter estimation on a single trial bases. It is important to notice that, even though passive, the control system is able to maintain the assembly in equilibrium after perturbations up to 25 % of its height. Furthermore,

even though the springs were not damped, the elastic network produces a damped response. This behavior was previously reported for pre-tensioned structures in [33], where muscle co-contraction was found to modulate damping in a multi-link biomechanic limb. Future studies will take into account also the joint friction as a possible source of damping.

The present study has at least three limitations. First, in order to reduce costs due to fabrication and material, we designed a small humanoid robot. In the future it will be necessary to increase its size in order to include motors, control modules and sensors. Second, this prototype does not have a real torso and arms. It is well known that these components are essential for maintaining stability during posture and gait in humans. Third, the EKF was manually tuned, as shown in Eq. 9. Choosing parameters of springs in elastic network is the fundamental problem [34]. If on one hand, the SISO modeling eliminates the advantage of having an elastic network, on the other hand, tuning the parameters and modeling the assembly as a MIMO system is the real challenge [35].

In the future, we aim to carry on the work by investigating the viscoelastic properties of the whole assembly, by solving the aforementioned limitations, and by introducing motors, control modules and sensors so that this new elastic humanoid robot can move its first step forward.

7 Supplementary Material

The experimental processes described in Sect. 5 are shown in the video provided as supplementary material at the following link:

<https://youtu.be/VYIZTNSfL4E>.

Acknowledgements We would like to thank Ph.D. Fuben He and Prof. Yande Liang, which are with the School of Mechanical Engineering of the Dalian University of Technology, China. They contributed on the mechanical project and fabrication of the first prototype, as documented in references.

References

1. Khatib, O., Demircan, E., De Sapio, V., Sentis, L., Besier, T., Delp, S.: Robotics-based synthesis of human motion. *J. Physiol. Paris* **103**(3–5), 211–219 (2009)
2. Ohta, H., Yamakita, M., Furuta, K.: From passive to active dynamic walking. *Int. J. Robust Nonlinear Control* **2001**(11), 287–303 (2001)
3. Hanazawa, Y., Suda, H., Iemura, Y., Yamakita, M.: Active walking robot mimicking flat-footed passive dynamic walking. In: 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1281–1286 (2012)
4. Zhou, X., Guan, Y., Zhu, H., Wu, W., Chen, X., Zhang, H., Fu, Y.: Bibot-U6: a novel 6-DoF biped active walking robot—modeling, planning and control. *Int. J. Human. Robot.* **11**, 1450014 (2014)

5. McGeer, T.: Passive dynamic walking. *Int. J. Robot. Res.* **9**(2), 62–82 (1990)
6. Collins, S.H., Ruina, A.L., Tedrake, R., Wisse, M.: Efficient bipedal robots based on passive-dynamic walkers. *Science* **307**, 1082 (2005)
7. Hitomi, K., Shibata, T., Nakamura, Y., Ishii, S.: Reinforcement learning for quasi-passive dynamic walking of an unstable biped robot. *Robot. Autom. Syst.* **54**(12), 982–988 (2006)
8. Omer, A.M.M., Ghorbani, R., Hun-ok, L., Takanishi, A.: Semi-passive dynamic walking for biped walking robot using controllable joint stiffness based on dynamic simulation. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2009*, pp. 1600–1605 (2009)
9. Visser, L.C., Carloni, R., Stramigioli, S.: Variable stiffness actuators: a port-based analysis and a comparison of energy efficiency. In: *2010 IEEE International Conference on Robotics and Automation (ICRA) (2010)*
10. Cotton, S., Oлару, I.M.C., Bellman, M., van der Ven, T., Godowski, J., Pratt, J.: Fast runner: a fast, efficient and robust bipedal robot. Concept and planar simulation. In: *Proceedings of 2012 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2358–2364 (2012)
11. Van Ham, R., Vanderborght, B., Van Damme, M., Verrelst, B., Lefeber, D.: MACCEPA: the actuator with adaptable compliance for dynamic walking bipeds. In: *Proceedings of the 8th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR 2005)*, pp. 759–766 (2006)
12. Vanderborght, B., Van Ham, R., Lefeber, D., Sugar, T.G., Hollander, K.W.: Comparison of mechanical design and energy consumption of adaptable, passive-compliant actuators. *Int. J. Robot. Res.* **28**(1), 90–103 (2009)
13. Radkhah, K., Lens, T., Seyfarth, A., von Stryk, O.: On the influence of elastic actuation and monoarticular structures in biologically inspired bipedal robots. In: *Proceedings of the 2010 IEEE International Conference on Biomedical Robotics and Biomechanics*, pp. 389–394 (2010)
14. Junius, K., Cherelle, P., Brackx, B., Geeroms, J., Schepers, T., Vanderborght, B., Lefeber, D.: On the use of adaptable compliant actuators in prosthetics, rehabilitation and assistive robotics. In: *Proceedings of the 9th International Workshop on Robot Motion and Control, Wasowo Palace, Wasowo, Poland, 3–5 July 2013*
15. Pratt, G.A., Williamson, M.M.: Series elastic actuators. *Intelligent robots and systems 95. Human robot interaction and cooperative robots*. In: *IEEE/RSJ International Conference on Proceedings 1995*, vol. 1, pp. 399–406 (1995)
16. Winter, D.A., Patla, A.E., Rietdyk, S., Ischac, M.G. 2001. Ankle muscle stiffness in the control of balance during quiet standing. *J. Neurophys.* 2001 Jun; **85**(6):2630-3
17. Casadio, M., Morasso, P.G., Sanguineti, V.: Direct measurement of ankle stiffness during quiet standing: implications for control modelling and clinical application. *Gait Posture* **21**(4), 410–424 (2005)
18. Westwick, D.T., Perreault, E.J.: Estimates of acausal joint impedance models. *IEEE Trans. Biomed. Eng.* **59**(10), 2913–2921 (2012)
19. Radkhah, K., Scholz, D., von Stryk, O., Maus, M., Seyfarth, A.: Towards human-like bipedal locomotion with three-segmented elastic legs. In: *Proceedings of the 41st International Symposium on Robotics (ISR 2010) and 6th German Conference on Robotics (ROBOTIK 2010)*, pp. 696–703 (2010)
20. Radkhah, K., Lens, T., von Stryk, O.: Detailed dynamics modeling of BioBiped’s monoarticular and biarticular tendon-driven actuation system. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4243–4250 (2012)
21. Bortoletto, R., Sartori, M., He, F., Pagello, E.: Simulating an elastic bipedal robot based on musculoskeletal modeling. In: *Proceedings of Biomimetic and Biohybrid Systems (Living Machines). LNCS*, vol. 7375, pp. 26–37 (2012)
22. Bortoletto, R., Sartori, M., He, F., Pagello, E.: Modeling and simulating compliant movements in a musculoskeletal bipedal robot. In: *Proceedings of Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN)*. LNCS, vol. 7628, pp. 237–250 (2012)

23. He, F., Liang, Y., Zhang, H., Pagello, E.: Modeling, dynamics and control of an extended elastic actuator in musculoskeletal robot system. In: Proceedings of the 12th International Conference IAS-12. Advances in Intelligent Systems and Computing, vol. 194, pp. 671–681 (2013)
24. Papich J.R., Kennett C.J., Piovesan D.: Open-source software in biomedical education: from tracking to modeling movements. In: 121st American Society for Education in Engineering Annual Conference ASEE (2014)
25. Delp, S.L., Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T., Guendelman, E., Thelen, D.G.: OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE Trans. Biomed. Eng.* **54**(11), 1940–50 (2007)
26. Reilly, T., O'Rourke, J.K., Steudler, D., Piovesan D., Bortoletto, R.: Locomotive Underactuated Implement Guided via Elastic Elements (L.U.I.G.E.E.): a preliminary design. In Proceedings of the ASME 2015 International Mechanical Engineering Congress & Exposition (2015)
27. Piovesan, D., Pierobon, A., DiZio, P., Lackner, J.R.: Measuring multi-joint stiffness during single movements: numerical validation of a novel time-frequency approach. In: *PloS one*, vol. 7, p. e 33086 (2012)
28. Romero, C.R., Cardenas, R. Piovesan, D.: Viscoelastic properties of the ankle during quiet standing via raster images and EKF. In: 2014 IEEE Signal Processing in Medicine and Biology Symposium, pp. 1–5 (2014)
29. Simon, D.: Optimal State Estimation, Kalman, H_{∞} , and Nonlinear Approaches. John Wiley & Sons, Inc., New Jersey (2006)
30. Knabe, C.S., Orekhov, V., Hopkins, M.A., Lattimer, B.Y., Hong, D.W.: Two configurations of series elastic actuators for linearly actuated humanoid robots with large range of motion. In: 2014 14th IEEE-RAS International Conference on Humanoid Robots (2014)
31. Garcia, C.E., Prett, D.M., Morari, M.: Model predictive control: theory and practice—a survey. *Automatica* **25**(3), 335–348 (1989)
32. Grimmer, M., Eslamy, M., Glied, S., Seyfarth, A.: A comparison of parallel- and series elastic elements in an actuator for mimicking human ankle joint in walking and running. In: International Conference on Robotics and Automation, pp. 2463–2470 (2012)
33. Heitmann, S., Breakspear, M., Ferns, N.: Muscle co-contraction modulates damping and joint stability in a three-link biomechanical limb. *Front. Neurobotics* **11**(5), 5 (2012)
34. Saha, M., Goswami, B., Ghosh, R.: Two novel costs for determining the tuning parameters of the Kalman filter. In: Proceedings of Advances in Control and Optimization of Dynamic Systems (ACODS-2012) (2011). eprint [arXiv:1110.3895v2](https://arxiv.org/abs/1110.3895v2)
35. Esfandiari, R.S., Lu, B.: Modeling and Analysis of Dynamic Systems. CRC Press, 566 pp., 24 Apr 2014. ISBN: 1466574933, 9781466574939

Higher Jumping of a Biped Musculoskeletal Robot with Foot Windlass Mechanism

Xiangxiao Liu, Yu Duan, Andre Rosendo, Shuhei Ikemoto
and Koh Hosoda

Abstract The complex of human foot plays an important role in the locomotion. Properly replicating the human foot characteristics on the humanoid robot foot design is supposed to improve the robot locomotion performance. In this research, we proposed three kinds of foot design, stiff foot, the windlass mechanism foot (with stiff plantar fascia) and the windlass mechanism foot (with elastic plantar fascia). Using a musculoskeletal biped robot and via a large set of dropping jump experiments, we confirmed that (1) the robot could achieve toe-off motion in the lifting off phase of jumping and (2) the windlass mechanism could increase the jumping height. This investigation on robot foot is expected to both improve the humanoid robot jumping performance and help us understanding how human achieve high performance locomotion.

Keywords Musculoskeleton · Pneumatic actuator · Humanoid robot · Bio-inspired robot · Jumping · Windlass mechanism

1 Introduction

In the human locomotion such as jumping and running, the foot complex plays an important role in the performance and energy recycling. Replicating the human foot characteristics on robot foot design is supposed to both improve the locomotion performance and the understanding of human foot. In the past decades, many studies on robot foot design were implemented.

Equipped with novel feet, some robots achieved human-like behavior in walking. Following the shape of human foot, Kwon et al. carried out flexible robot foot design

X. Liu (✉) · Y. Duan · S. Ikemoto · K. Hosoda
Osaka University, 1-3, Machikaneyama, Toyonaka, Osaka, Japan
e-mail: liuwx416@gmail.com
URL: <http://www-arl.sys.es.osaka-u.ac.jp/>

A. Rosendo
University of Cambridge, Trumpington Street, Cambridge, UK

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_25

and showed that the ankle trajectory could follow the rotational motion of a human foot while being lifted from and landing on the ground [1]. Ogura et al. proposed a compliant foot design with a passive toe constrained by springs [2]. Being equipped with the passive toes, the experiment showed that the double peaks of the vertical forces could be obtained in walking, which is the same as human.

Comparing to the conventional stiff and flat robot foot, Some researchers showed that with the novel foot, the robot locomotion performance could be improved. Minakata et al. carried out a pair of flexible robot shoes [3]. With the springs on the bottom, the shoes could optimize the energy consumption in walking. In [4], the human-like foot with a passive toe joint enabled the humanoid robot WABIAN-2R to turn by using slipping motion. Comparing to the stepping turn, the slipping turn is quicker and more energy efficient. In [5], with walking simulation, Sellaouti et al. showed that with the passive toe joints, the step length could be increased and the robot could walk faster and smoother.

Those works mentioned above centralized in improving the walking behavior by novel robot foot design. However, as the best of the author's knowledge, There is few research has investigated the contribution of novel robot foot design on jumping except in [6]. In [6], biped robot HRP-2LR with toe springs was developed and could achieve dropping jump. While, in this research, the contribution of the toe-off motion was not clear.

Improving robot jumping performance always attracts researchers. The investigations on robot jumping could both improve the robot performance and the understanding of human locomotion. Leading by the one leg hopping robot developed by pioneer M. Raibert [7], in the past decades, many investigations were implemented to replicate jumping on robots and uncover its theory. While, the researches of robot foot centralized in the contribution in walking and there is few investigation on the contribution of the novel robot foot with human foot characteristics in jumping. It is necessary to investigate how to improve the jumping performance by new robot foot design.

Plantar fascia is a thick connective tissue runs from the calcareous to the heads of the metatarsal bones. Together with the bones (phalanges, metatarsals and calcareous) in the foot, the musculoskeletal structure forms the foot windlass mechanism and plays an important role in the kinetics qualities and kinematics of human foot and ankle [8]. In medicine and biology, Cadaver feet and animal experiment are wildly applied to uncover the activity of plantar fascia. In [9], by the observation on human, A. GefenBy found that plantar fascia elongates faster in mid stance in comparing with heel off to toe off. G.D. Stainsby also showed a similar observation that plantar fascia is stretched in toe-off motion [10]. Testing the cadaver feet in a dynamic gait stimulator, Hamel et al. proved that plantar fascia enables more efficient force transmission through the high gear axis during locomotion [11]. Those researches mentioned above indicate that plantar fascia is significantly stretched and activated in toe-off motion.

A few investigation on the locomotion contribution of plantar fascia were implemented. For example, a animal experiment taken by B.C. Abbott et al., with the combination of the soft tissues and muscles, the calf muscle-tendon unit can minimize the

positive muscle work and operate more efficiently than muscles alone [12]. While, in this experiment, the method of testing the energy efficiency was based on measurement of oxygen consumption during animal's locomotion. As not only muscles consume the oxygen and energy, also it is need to consider that there are many other aspects are influencing the oxygen consumption. Additionally, because of the limitation of the measurement and ethics in animal experiments. It is necessary to find new approaches and take further investigation on the contribution of the windlass mechanism structure.

Considering that the plantar fascia is significantly stretched in toe-off motion and the soft tissues play an important role in the locomotion, we hypothesizes that, with the windlass mechanism, the robot could achieve toe-off motion in jumping. Also, the robot could jump higher with the windlass mechanism feet.

In our research, first, a biped musculoskeletal robot, actuated by pneumatic artificial muscles, was developed. Following the electromyography data of human dropping jump in biology, we modified a control sequence and made the robot achieving jumping. In order to investigate the contribution of the foot windlass mechanism, we designed three kinds of robot foot, stiff foot (without the windlass mechanism), windlass mechanism foot with stiff plantar fascia (WMFSP foot) and windlass mechanism foot with elastic plantar fascia (WMFEP foot). A large set of dropping jump experiments were implemented. We compared the average jumping height of three kinds of designed foot. As the result, we confirmed that (1) with the foot windlass mechanism, the robot shows the toe-off motion in the end stance phase in jumping and (2) the foot windlass mechanism could improve the maximum jumping height.

This research, on one hand, improved robot performance by proposing a passive musculoskeletal foot, which could enrich the theory of robot design; on the other hand, this research improved the understanding of human foot musculoskeletal mechanism.

2 Robot Design

A biped pneumatic musculoskeletal robot was used in our investigation (shown in Fig. 1a). With the height of 1280 mm, depth of 150 mm and width of 2400 mm (distance between both feet), the robot could imitate a small size human. The center of gravity (COG) locates about 57 % of the robot height from the ground, approaching the characteristics of the location of human's COG. Designed for biped jumping, the robot is built with three hinge joints in each leg (hip, knee and ankle). Following the anatomy, each leg is equipped with and actuated by nine artificial muscles.

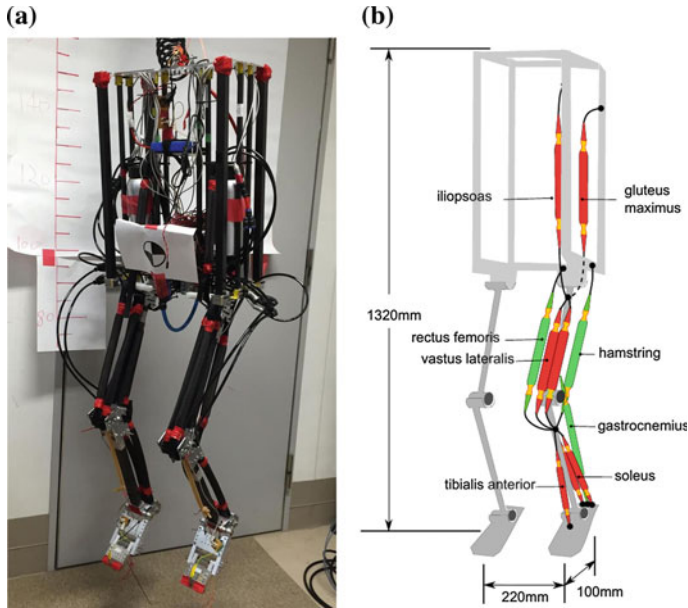


Fig. 1 **a** The developed robot. **b** Musculoskeletal structure of the robot, monoarticular muscles in red and biarticular muscles in green

2.1 Artificial Air Muscle

Successfully used as artificial muscles in many robotic experiments [13, 14], pneumatic artificial muscles are chosen as the robot's actuators. Pneumatic artificial muscles are considered as a good replacement for the human muscles for its properties, such as high power-mass ratio, soft and elastic. Supplying high pressure air, a air muscle can contract and provide tension as a biological muscle. Exhaust the air and the muscles can replicate the relaxing effect of biological muscles. The muscles are shown in Fig. 2. The force provided by a pneumatic artificial muscle has a proportional relationship with internal air pressure and muscular deformation (shown in the following equation) [15].

$$F \propto \frac{P_{air} \Delta L}{L_0}$$

where F is the tension, P_{air} is the supplying air pressure, ΔL is the deformation with air supply and L_0 is the muscle's length without air supply. Thus, the output (force) of a pneumatic artificial muscle is just related to the muscle's deformation and the pressure of the air. At the same pressure level, muscles with high deformation could generate higher output than with low deformation.

Fig. 2 Artificial muscles in the robot. With supply (*upper*) shows the length of 18 cm, and without supply (*below*) shows the length of 20 cm



2.2 Musculoskeletal Structure

Nine artificial muscles are equipped in each leg of the robot to actuate the robot. Three of them are biarticular muscles and six of them are monoarticular muscles (shown in Fig. 1b). The length of the each muscle is determined by following the data from the observations of human anatomy and modified in robot experiment [16–18].

2.3 Valve Operation

In order to achieve jumping, the robot should be light. Consequently, instead of using heavy proportion valves which could enable accurate control of airflow. We chose the light pilot-operated on-off valves. Animals possess two kinds of muscles, monoarticular muscles and biarticular muscles. Monoarticular muscles mainly contribute to power supplement during locomotion and biarticular muscles mainly contribute to coordination [14, 19, 20]. Therefore, we only activated the monoarticular muscles during jumping.

Jumping is one of the most difficult locomotion for both human and robot. Following the observation of human muscles activation during jumping in biomechanics and adjusting in robot jumping experiments [21, 22], we generated a three state valve operation control adapting for landing, complying and supplying. The control program runs among three states in loop (shown in Fig. 3).

In the landing phase of human jumping, before touching the ground, human muscles are activated in a certain extent [23, 24]. In the initialization of the robot dropping jump, a predetermined air is supplied to each muscle and constrains the robot to the posture preparing for landing. The landing posture is based on the investigation of the dynamics in human jumping [23, 25].

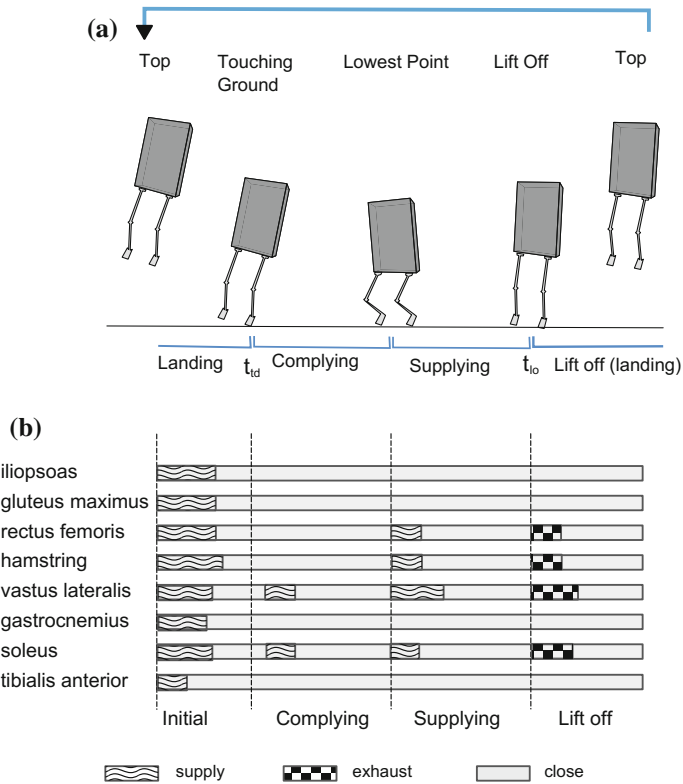


Fig. 3 **a** Three states control for jumping (landing, complying and supplying) and **b** Valves operation for jumping

2.3.1 Landing State

In the landing state, first, the robot adjusts each muscle to a predetermined initialization pressure. After adjustment, the robot waits for touching the ground. FSRs (force sensing resistors) are used to detect the touching between the feet and ground.

2.3.2 Complying State

It is found that in landing phase of human jumping, a strong muscle activation (stretch reflex) occurs on soleus and vastus lateralis after landing immediately (just with very short delay) [26]. Consequently, after the robot feet touch the ground, we applied stretch reflex to both soleus and vastus lateralis muscles.

After the first foot touching the ground, the robot reaches the complying state. In this state, a extra air supply is applied to imitate stretch reflex on soleus and vastus lateralis muscles. After the robot touches the ground, a 60 ms air supply to soleus

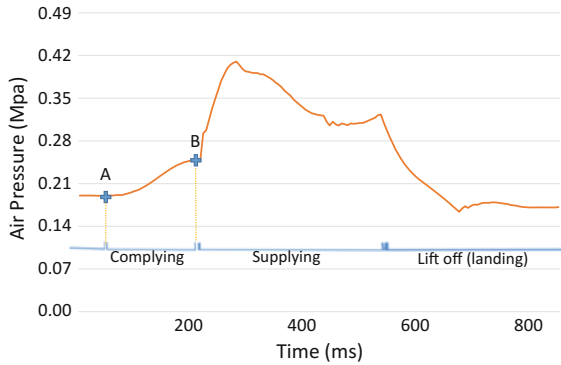


Fig. 4 Detection of the bottom point in jumping. The blue line indicate the air pressure of the soleus in the robot. In point A, the foot touches the ground and the robot starts its complying state. The complying state ends when the pressure of vastus lateralis increases to the top and starts to decrease (point B). At this point, the vastus lateralis is longest stretched and the robot reaches the bottom. The typical time between touching the ground and reaching the bottom is 180 ms

muscle is set to act as stretch reflex. We chose the valve of 30 ms based on investigation of muscle activation in human jumping by measuring electromyography kinematics [21, 26].

Air pressure in muscles are measured by pressure sensors (produced by SMC Co., Ltd.). When the pressure inside the a soleus comes to a peak (shown in Fig. 4), the artificial muscle is stretched to the maximum deformation and the robot reaches the lowest point [14]. At this moment, the next state (supplying state) is activated. as shown in the following equation:

$$\frac{\partial P_{air}}{\partial t} = 0$$

where P_{air} is the air pressure of the soleus and t is the time.

2.3.3 Supplying State

In this state, air is supplied to three monoarticular muscles (soleus, vastus lateralis and gluteus maximus) to generate upward power. This state ends when the robot loses contact with the ground.

2.4 Foot Design

In biology, it was found that the plantar fascia continuously elongated during the contact phase of gait. The plantar fascia goes through rapid elongation before and

immediately after mid-stance, reaching a maximum of 9–12 % elongation between mid-stance and toe-off. Also, the research in [10] shows a similar finding on the elongation of plantar fascia in toe-off motion. With the stretch of plantar fascia, a force could be generated along the lower-leg limb and push the body moving fore-upward with the foot windlass mechanism (shown in Fig. 5a1, a2). Besides, Hamel et al. proved that plantar fascia enables more efficient force transmission through the high gear axis during locomotion [11].

We believe that comparing to the conventional robot foot design, the foot windlass mechanism could improve the robot jumping height. Considering the skeletal struc-

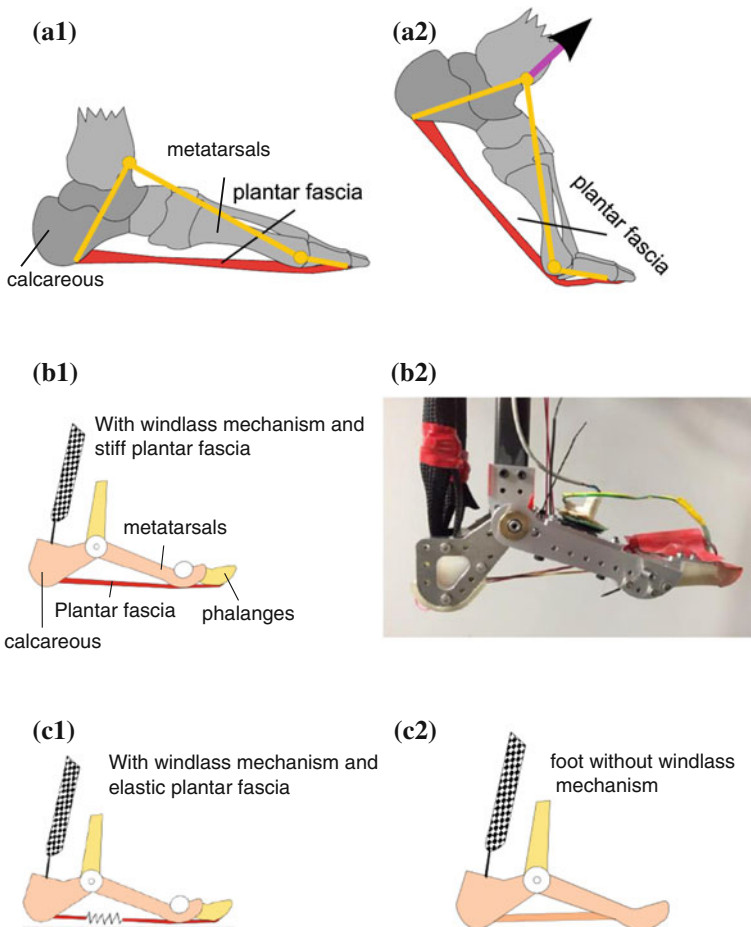


Fig. 5 Foot design. **a1** The windlass mechanism in human foot. **a2** The plantar fascia is stretched in toe-off motion and generate a fore-upward force. **b1** The WMFSP foot design, including the phalanges, metatarsals, calcareous and one soft tissue. **b2** The designed foot (WMFSP foot). **c1** The WMFEP foot. **c2** The stiff foot

ture of human foot, we proposed a foot design with three segments (WMFSP foot), which replicate the phalanges, metatarsals and calcareous, the three main segments in human foot (shown in Fig. 5b1). Two hinge joints connect the three segments and allow the rotation in sagittal plane. Replicating the plantar fascia in human foot, a artificial tissue made by nylon connects between the phalanges to calcareous and plays the role of the plantar fascia tissue. Combining with the bones (phalanges, metatarsals and calcareous), this musculoskeletal structure could mimic the windlass mechanism in human foot.

In order to make comparison, we also proposed other two kind of foot design, stiff foot and WMFEP foot (shown in Fig. 5c1, c2). As the stiff foot, based on the WMFSP foot design, the connection between calcareous and metatarsals bones was fixed. Besides, the toe joint was un-installed. This design is to mimic the conventional stiff foot robot foot. To replicate the elasticity of plantar fascia, we carried out the WMFEP foot. Based on the WMFSP foot, the stiff plantar fascia was replaced by a elastic plantar fascia made by spring.

For all the foot design, under the foot, soft plastic shell is attached to simulate human connective tissue and epithelial tissue on foot bottom. On one hand, this design could improve the adaption on the uneven ground; on the other hand, this design could help the robot absorbing the impact in landing. We kept the weight of each designed foot on 450 ± 10 grams.

2.5 Other Devices

Other devices include a air compressor to supply the high pressure air, 9-axis motion sensors to measure the body posture and a battery to supply the power. Those devices are properly arranged and placed in the robot so that the robot could locate the COG on about 57 % length of the body height from the ground.

3 Experiment

3.1 Jumping Experiment

Dropping jump is widely used in both medicine and robotics to evaluate the power and skill of locomotion. We implemented drop jumping experiment to verify the proposed foot design. First, equipping the stiff robot feet, we adjusted the jumping control sequence and made it to be optimal to the robot with the stiff feet. We also implemented the dropping jump experiment to the robot equipped with the other two kinds of designed feet (WMFSP foot and WMFEP foot) and made sure the robot could achieve jumping. Then, we investigated the performance contribution of the foot windlass mechanism. In this experiment, dropping jump experiments were

implemented and three kinds of designed foot were tested and compared (stiff foot, WMFSP foot and WMFEP foot). To cover the disturbances from the environment in jumping experiment, we focused on the average maximum jumping height and for each kind of foot design, over 30 times jumping trails have been implement. The jumping experiment is described in follow:

1. The experimenter hangs the robot in a height of 15 cm (the distance between the robot toes and the ground). Then, the experimenter releases the robot statically.
2. The robot jumps up following the control program.
3. After the robot reaching the highest point and starting landing, the experimenter catches the robot in midair and prepares for the next trial.

For all the jumping trails, considering the energy of jumping was supplied by the high pressure air, we applied the same air supplement for all trails. Consequently, each trail consumed the same energy. In order to measure the jumping height, a high speed camera (produced by DITECT corporation) is used to record the jumping. The sampling frequency of the high speed camera is 120 fps. Also, in the center of the robot torso (where the COG locates on), a mark was attached to help the detection of the position of robot. A ruler is set vertically near the robot to help measuring the jumping height. The robot reached the maximum height when the vertical speed decreases to zero, as described by the following equation.

$$\frac{\partial h}{\partial t} = 0$$

where h is the jumping height of the robot and t is the time.

Three kinds of foot design were compared, stiff foot, WMFSP foot and WMFEP foot. In the setting of the all designed feet, we ensured the height of arch was always 45 mm (± 1 mm). In order to evaluate the average behavior of the robot jumping, for each kind of designed foot, we implemented over 30 times jumping trails. The video taken by the high speed camera was analyzed after the experiment. The maximum jumping height was measured by comparing the position of the COG mark and the ruler in the video.

For the convenience and easy understanding, we divided the data into nine intervals based on the maximum jumping height, [5–10], [10–15], [15–20], [20–25], [25–30], [30–35], [35–40], [45–50] (unit: cm). The count of the maximum jumping height is described in Fig. 7a–c.

We compared the average jumping height of the experiment of three kinds of foot design. For evaluation, a one-factor ANOVA test with Tukey's HSD (Tukey's honest significant difference) was used to compare the results and detect the significant difference. In Fig. 7d. the means of maximum jumping height of the robot are presented and compared. It could be found that comparing to the jumping height of the robot with the stiff foot, the robot could jump higher with equipping the windlass mechanism feet (WMFSP foot and WMFEP foot). Comparing the average jumping height between the WMFSP foot and WMFEP foot, the Tukey's HSD test did not show the significance.

4 Conclusion and Discussion

In this research, a biped robot was developed by replicating the human musculoskeleton and three kinds of robot foot design were proposed. We implemented dropping jump experiment and investigated the contribution of the foot windlass mechanism in robot jumping.

In Fig. 6, it could be found that releasing the robot from a height of 15 cm (the height of COG is around 100 cm), the maximum jumping height could reach 45 cm. This result shows that the high performance of pneumatic musculoskeletal robot. From the shot of the video, we could confirm that at the end of stance phase in jumping, the robot achieved toe-off motion.

Figure 7d contains the mean of jumping height of the three kinds of foot design. It could be found that comparing with equipping the stiff foot, the robot could jump significantly higher with both WMFSP foot and WMFEP foot according to the Tukey's HSD test. This result indicates that the foot windlass mechanism could increase the height of robot jumping.

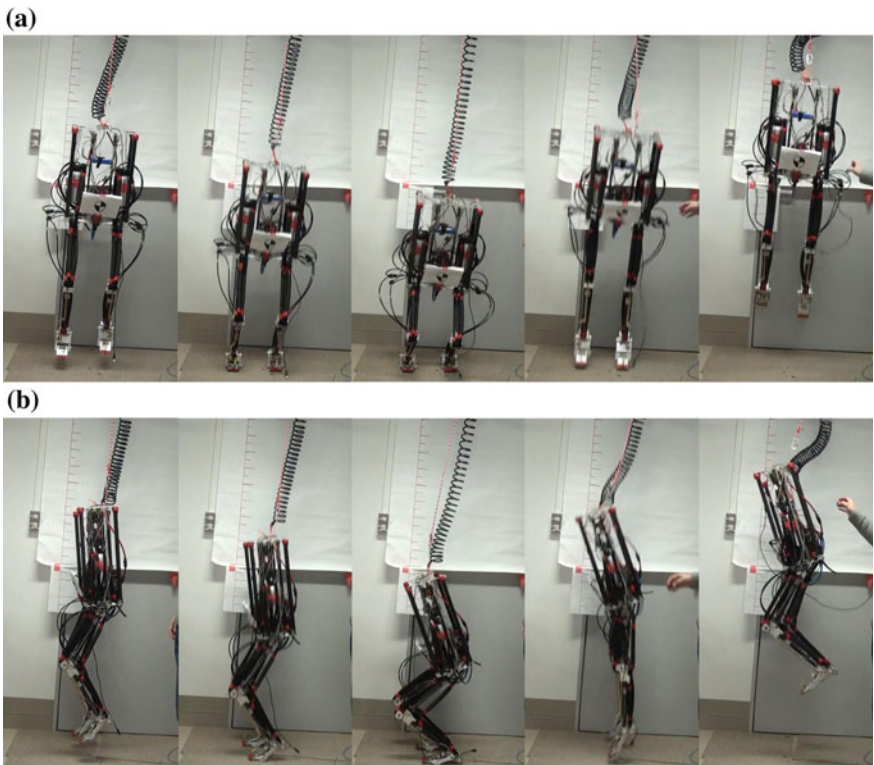
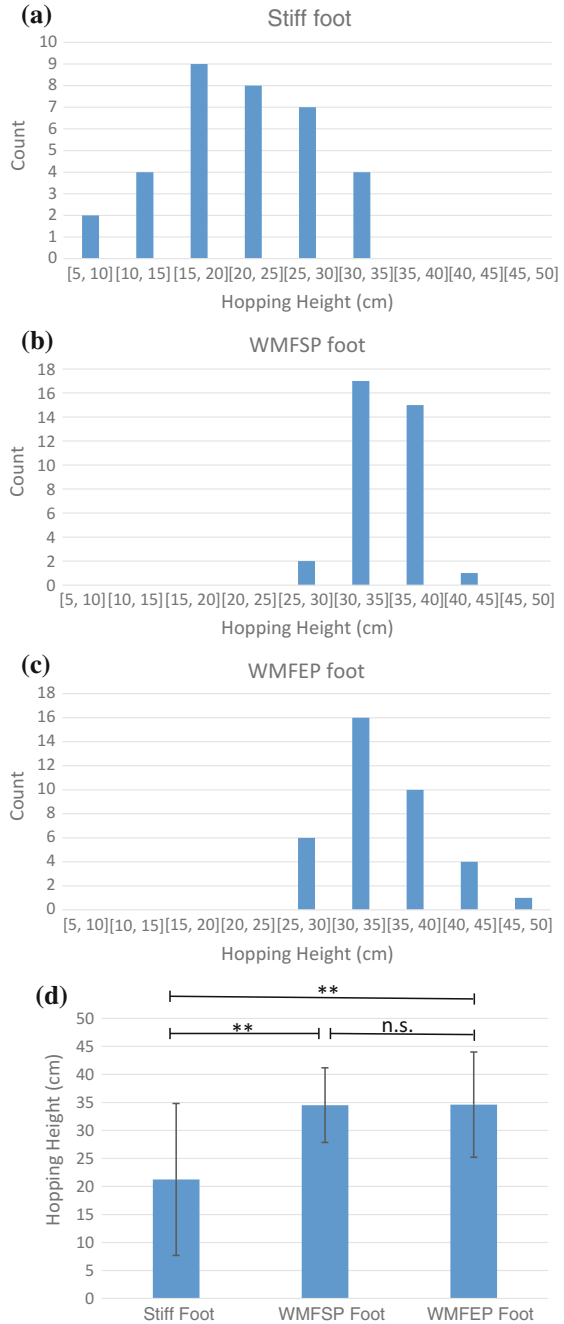


Fig. 6 Dropping jump experiment. Releasing from 15 cm height, the robot could jump 40 cm height

Fig. 7 Experiment results. For the figures (a), (b), (c), jumping height is in horizontal and count in vertical. **a** Stiff foot (without the windlass mechanism). **b** WMFSP foot and **c** WMFEP foot. In **d**, the average maximum jumping height of three kinds of foot design is compared. Error bars are affixed to show the standard deviation. Tukey' HSD test shows the significant difference. Non-significant (n.s.) and $p < 0.01(**)$ are marked



Also in Fig. 7d, comparing the average jumping height between the WMFSP foot and WMFEP foot, according to the Tukey's HSD test, no significance was detected. This result suggests that jumping height of both robot foot design with the foot windlass mechanism are similar. While, as human foot, the plantar fascia is elastic and is supposed to store the energy in locomotion. The robot should jump higher with WMFEP foot than with WMFSP foot. One possibility is that we chose a over-stiff spring (126 N/cm) to make the artificial plantar fascia for the robot. The investigation of using softer artificial plantar fascia and measuring the deformation of the robot arch remains the future issue.

References

1. Kwon, S., Park, J.: Kinesiology-based robot foot design for human-like walking. *Int. J. Adv. Robot. Syst.* **9** (2012)
2. Ogura, Y., Shimomura, K., Kondo, H., Morishima, A., Okubo, T., Momoki, S., Lim, H.-O., Takanishi, A.: Human-like walking with knee stretched, heel-contact and toe-off motion by a humanoid robot. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3976–3981. IEEE (2006)
3. Minakata, H., Seki, H., Tadakuma, S.: A study of energy-saving shoes for robot considering lateral plane motion. *IEEE Trans. Ind. Electron.* **55**(3), 1271–1276 (2008)
4. Hashimoto, K., Yoshimura, Y., Kondo, H., Lim, H.-O., Takanishi, A.: Realization of quick turn of biped humanoid robot by using slipping motion with both feet. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 2041–2046. IEEE (2011)
5. Sellaouti, R., Stasse, O., Kajita, S., Yokoi, K., Kheddar, A.: Faster and smoother walking of humanoid hrp-2 with passive toe joints. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4909–4914. IEEE (2006)
6. Kajita, S., Kaneko, K., Morisawa, M., Nakaoka, S., Hirukawa, H.: Zmp-based biped running enhanced by toe springs. In: 2007 IEEE International Conference on Robotics and Automation, pp. 3963–3969. IEEE (2007)
7. Raibert, M.H.: *Legged Robots That Balance*. MIT press (1986)
8. Gefen, A.: Stress analysis of the standing foot following surgical plantar fascia release. *J. Biomech.* **35**(5), 629–637 (2002)
9. Gefen, A.: The in vivo elastic properties of the plantar fascia during the contact phase of walking. *Foot Ankle Int.* **24**(3), 238–244 (2003)
10. Stainsby, G.: Pathological anatomy and dynamic effect of the displaced plantar plate and the importance of the integrity of the plantar plate-deep transverse metatarsal ligament tie-bar. *Ann. R. Coll. Surg. Engl.* **79**(1), 58 (1997)
11. Hamel, A.J., Donahue, S.W., Sharkey, N.A.: Contributions of active and passive toe flexion to forefoot loading. *Clin. Orthop. Relat. Res.* **393**, 326–334 (2001)
12. Abbott, B., Bigland, B., Ritchie, J.: The physiological cost of negative work. *J. Physiol.* **117**(3), 380–390 (1952)
13. Niiyama, R., Kuniyoshi, Y.: Design principle based on maximum output force profile for a musculoskeletal robot. *Ind. Robot Int. J.* **37**(3), 250–255 (2010)
14. Hosoda, K., Sakaguchi, Y., Takayama, H., Takuma, T.: Pneumatic-driven jumping robot with anthropomorphic muscular skeleton structure. *Auton. Robots* **28**(3), 307–316 (2010)
15. Fujii, F., Murata, H., Hirose, Y., Okada, T.: A simple phenomenological modeling of mckibben pneumatic actuator and its application to control. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 4724–4730. IEEE (2013)
16. Wickiewicz, T.L., Roy, R.R., Powell, P.L., Edgerton, V.R.: Muscle architecture of the human lower limb. *Clinical Orthop. Relat. Res.* **179**, 275–283 (1983)

17. Friederich, J.A., Brand, R.A.: Muscle fiber architecture in the human lower limb. *J. Biomech.* **23**(1), 91–95 (1990)
18. Hoy, M.G., Zajac, F.E., Gordon, M.E.: A musculoskeletal model of the human lower extremity: the effect of muscle, tendon, and moment arm on the moment-angle relationship of musculo-tendon actuators at the hip, knee, and ankle. *J. Biomech.* **23**(2), 157–169 (1990)
19. Paul, C., Bellotti, M., Jezernik, S., Curt, A.: Development of a human neuro-musculo-skeletal model for investigation of spinal cord injury. *Biol. Cybern.* **93**(3), 153–170 (2005)
20. van Ingen Schenau, G.J., Pratt, C.A., Macpherson, J.M.: Differential use and control of mono- and biarticular muscles. *Hum. Mov. Sci.* **13**(3), 495–517 (1994)
21. Bobbert, M.F., van Ingen Schenau, G.J.: Coordination in vertical jumping. *J. Biomech.* **21**(3), 249–262 (1988)
22. Pandy, M.G., Zajac, F.E.: Optimal muscular coordination strategies for jumping. *J. Biomech.* **24**(1), 1–10 (1991)
23. Farley, C.T., Morgenroth, D.C.: Leg stiffness primarily depends on ankle stiffness during human hopping. *J. Biomech.* **32**(3), 267–273 (1999)
24. Funase, K., Higashi, T., Sakakibara, A., Imanaka, K., Nishihira, Y., Miles, T.: Patterns of muscle activation in human hopping. *Eur. J. Appl. Physiol.* **84**(6), 503–509 (2001)
25. Auyang, A.G., Yen, J.T., Chang, Y.-H.: Neuromechanical stabilization of leg length and orientation through interjoint compensation during human hopping. *Exp. Brain Res.* **192**(2), 253–264 (2009)
26. Voigt, M., Dyhre-Poulsen, P., Simonsen, E.B.: Modulation of short latency stretch reflexes during human hopping. *Acta Physiol. Scand.* **163**(2), 181–194 (1998)

A Motion Planning Architecture for Conveyance Tasks with a Quadruped Robot

Giulio Cerruti, Wei-Zhong Guo and Fulvio Mastrogiovanni

Abstract This paper describes a motion planning architecture for a quadruped robot used for conveyance tasks. The architecture is designed to guarantee robot's locomotion stability using only proprioceptive information. The robot is equipped with linear displacement sensors and pressure sensors located on the hydraulic cylinders used as actuators, a gyroscope on the main body and contact sensors on the leg tips. The robot knowledge is limited to proprioceptive sensory data and no a priori information is given about the environment. The resulting walking behaviour is validated through simulations on both flat terrain (with unknown objects along the path) and slopes. The gait is performed using a generic leg sequence and a simplified foothold planner. Initial experiments on the real platform have been carried out as well.

Keywords Motion planning • Quadruped robot • Conveyance task • Gait

1 Introduction

During the past few years, a number of robots have been developed, which demonstrate the advanced capabilities of quadruped robots as far as mobility, autonomy and speed are concerned, e.g., *Big Dog* [1], *HyQ* [2] and *Little Dog* [3]. This paper

G. Cerruti

Institut de Recherche en Communications et Cybernetique de Nantes and Aldebaran,
Nantes Cedex, France
e-mail: gcerruti@aldebaran.com

W.-Z. Guo

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China
e-mail: wzguo@sjtu.edu.cn

F. Mastrogiovanni (✉)

Department of Informatics, Bioengineering, Robotics and Systems Engineering,
University of Genoa, Genoa, Italy
e-mail: fulvio.mastrogiovanni@unige.it

© Springer International Publishing AG 2017

W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_26

introduces an approach to obtain a static walking gait based on proprioceptive sensory feedback only, with the aim of performing conveyance tasks on uneven terrain. Conveyance tasks are characterised by two strict requirements:

1. the robot's body must be kept horizontal during locomotion;
2. a constant height with respect to the ground must be guaranteed.

Research on walking stability starts with the definition of the *Static Stability Margin* (SSM) by McGhee and Frank [4]. The idea is inspired by the motion of insects, which arrange their step sequence to balance body weight. Further stability criteria are introduced by Zhang and Song [5], who propose the *Longitudinal* and *Crab Longitudinal Stability Margins* (respectively, LSM and CLSM), to obtain an estimate of SSM with a lighter computation. Messuri and Klein propose a stability measure based on the height of the robot Centre of Gravity (CoG) with respect to the ground, namely the *Energy Stability Margin* (ESM) [6]. ESM provides an efficient static stability measurement related to the minimum potential energy required to make the robot tumble around an edge of the supporting polygon, which is entailed by the leg tips location. Hirose et al. define the Normalised ESM (NESM), which estimates the robot stability on uneven terrain independently of the robot weight [7]. Static stability criteria hold if dynamic effects are negligible during the walking behaviour. In this paper, only statically stable criteria are considered (dynamic stability criteria are discussed in [8]).

The planning of static stable walking gaits for legged robots has been widely addressed in the literature. The first SSM-based six-legged machine, able to navigate on horizontal grounds, is introduced in [9]. Hirose et al. developed the so-called TITAN robot, which is able to walk on stairs using statically stable gaits [10]. Statically stable gaits are used also in robots for volcanic exploration [11]. In order to perform conveyance tasks, intermittent crawling gaits are proposed in [12], whereas omnidirectional, NESM-based motions with transitions between crawl and rotation gaits are investigated in [13]. Pongas et al. propose a flexible walking pattern generator for quadruped locomotion over rough terrain [14]. A robust CoG trajectory is obtained, which is based on a sinusoidal function that is automatically and smoothly adjusted according to the foot placement pattern.

In this paper, robot balance and constant height are guaranteed by controlling both the body and the swing leg trajectories. Since trajectory generation is based on splines, the obtained motion is smooth and twice differentiable, which is a desirable property for conveyance tasks. The body motion approach is based on the ESM as defined by Messuri and Klein [6]. However, our approach extends the original formulation taking into account the current leg tip positions, the body orientation and the presence of obstacles. The target robot platform is *Baby Elephant*, a quadruped robot developed at Shanghai Jiao Tong University. The paper is organised as follows. *Baby Elephant* is briefly described in Sect. 2. Section 3 introduces and describes the developed on-line static walking gait for uneven terrain. The main constraints are investigated, which characterise the motion and the body trajectory. Section 4 discusses relevant results obtained by simulating the walking gait in different conditions. Conclusions follow.

2 The Target Robot Platform

Baby Elephant (Fig. 1a) is made-up of an aluminium alloy body structure, to which four legs are attached. Each leg is actuated using three hydraulic cylinders, which are located on the robot's body. The robot is equipped with linear displacement sensors and pressure sensors on the cylinders, a 3-axis gyroscope on the robot body and Boolean contact sensors on leg tips. The robot body frame (BF) is placed at the geometric centroid of the robot body (Fig. 1c). Its position and orientation are defined with respect to an absolute world frame (WF). BF is an orthonormal right-handed frame with the z axis defined orthogonally with respect to the body plane and the x axis along the body sagittal plane (SP) and orthogonal to the body coronal plane (CP). In order to describe leg postures independently from the body, additional hip frames (HFs) are attached to the robot structure. Robot legs are numbered from front to rear: 1 (front-left, FL), 2 (front-right, FR), 3 (rear-left, RL) and 4 (rear-right, RR).

Each leg is a serial-parallel mechanism with three DoFs (Fig. 1b). The actuation is performed using three hydraulic actuators per leg connected to the on-board pump. The four identical legs are position controlled. Leg motions can be described along the coronal plane and the sagittal plane. In the CP, the leg is modelled as a parallelogram structure constituted by three revolute and one prismatic joints (i.e., the hydraulic cylinder), which drive the whole mechanism. In the SP, the leg is composed of six links and eight revolute joints. The links are arranged as a double parallelogram structure and are connected to each other by revolute joints. A ninth joint is applied to the middle of the crossing bars, therefore reducing the DoFs of the mechanism. In this way, only two actuators are needed to control the leg in the SP. Three actuators are needed to move the leg in space. The parallel structure allows us to increase the overall robot payload. The foot is a flat rubber tip. It is attached to the leg with a spring and a spherical joint, thereby being able to adapt to irregular terrain. This allows for modelling the contact with the ground as a point contact [15]. The contact sensor is Boolean. The study of the leg's workspace can be simplified

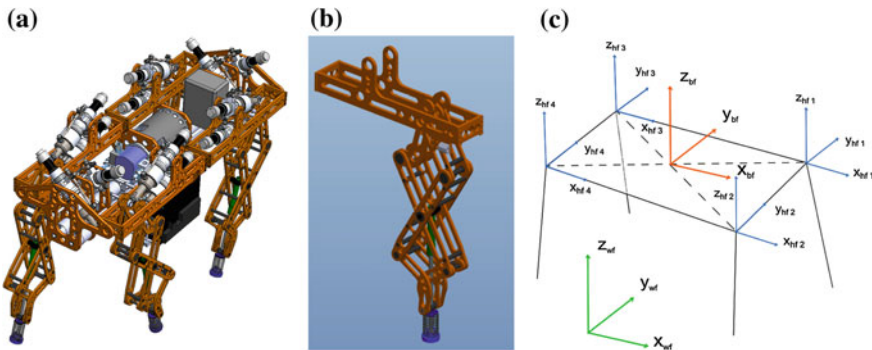


Fig. 1 **a** The Baby Elephant CAD model; **b** CAD model of the leg structure; **c** world frame (green), body frame (red) and hip frames (blue)

considering only its section in the SP. In particular, along the x axis (orthogonal to the CP) the maximal tip displacement is around 800 mm, whereas along the z axis (orthogonal to the horizontal plane) it is around 350 mm.

3 Obtaining a Static Stable Walking Gait

As anticipated, the target walking gait is conceived for conveyance tasks. For this reason, the robot's body must be kept horizontal and at a fixed height during walking. To this aim, priority is given to properly control the body position to guarantee static stability. Since static stability criteria hold only in absence of dynamic effects, the effect of inertial forces and momentum acting on the robot is minimised imposing slow motions. The adopted static stability criterion is ESM, which proves to be very accurate in measuring robot stability on uneven terrain.

3.1 ESM-Based Body Trajectory Planning

This Section describes how ESM is extended to stabilise the robot and how reference body poses are computed. Specifically, three steps are executed, namely the computation of the body pose sequence in accordance with the ESM criterion, its sampling to smooth the final displacement, and the interpolation of the final set of references with cubic splines to guarantee continuity during accelerations.

In order to maximise stability, the robot's body is horizontally shifted using three or four legs in support. For each configuration, a number of target poses characterised by the same ESM value are identified, which depend on the position of the CoG projected on the support polygon. This entails a three-dimensional energy surface which monotonically increases to the highest ESM value [6]. Hence, a gradient-based local search can be used to find the optimal (stable) pose. The trajectory Θ is iteratively computed searching for the best pose $p^*(x, y)$ among the neighbours $p^n(x, y)$ of the current body pose $p(x, y)$, such that $\Theta = \{p_1^*(x, y), \dots, p_T^*(x, y)\}$. It is noteworthy that since the motion is a pure shift, the z coordinate is assumed constant. When computing the generic point $p_i(x, y) \in \Theta$, if three legs are in support (i.e., three leg tips in contact are detected), the best neighbour pose maximises ESM:

$$p_i^*(x, y) = \arg \max_{p^n(x, y)} ESM(\text{CoG}(x, y)), \quad (1)$$

such that:

$$\begin{aligned} SSM(\text{CoG}(x, y), q) &> 0, \\ q &\in [q_{\min}, q_{\max}], \end{aligned} \quad (2)$$

where q is the vector of the joint configurations. If the robot has four legs in support (i.e., four leg tips in contact are detected), the best neighbour is subject to the same constraints, plus an additional constraint related to robot rotations:

$$p_t^*(x, y) = \arg \min_{p^n(x, y)} \Delta h_{max}(CoG(x, y)), \quad (3)$$

where:

$$\begin{aligned} \Delta h_{max} &= \max(\Delta h_{lr}, \Delta h_{fb}), \\ \Delta h_{lr} &= |h_l - h_r|, \\ \Delta h_{fb} &= |h_b - h_f|. \end{aligned} \quad (4)$$

In (4), parameters h_l , h_r , h_b and h_f are the CoG height variations considering robot rotations about the left, right, front and back support boundary edges, respectively. For each possible pose $p_t \in \Theta$, the algorithm checks its feasibility in terms of joint and stability constraints (see Sect. 3.3). If no neighbour pose is *feasible*, then the search is repeated and the step size of the gradient search is reduced by a constant value.

Once the trajectory Θ is obtained, it is sampled and interpolated with a cubic spline, in order to guarantee continuity in position, velocity and acceleration and to obtain a curve with a low average curvature [16]. The cubic spline function is defined by $T - 1$ third order, acceleration-linear polynomials, which connect the computed T poses in Θ . Assuming each point $p_t \in \Theta$ to be executed at a given time instant t , the spline, in each coordinate, is defined as follows:

$$\begin{aligned} \phi(t) &= c_0 + c_1 t + c_2 t^2 + c_3 t^3, \\ \dot{\phi}(t) &= c_1 + 2c_2 t + 3c_3 t^2, \\ \ddot{\phi}(t) &= 2c_2 + 6c_3 t. \end{aligned} \quad (5)$$

To completely describe the trajectory, $4(T - 1)$ boundary conditions are needed. The first four are related to the initial and final poses and accelerations, i.e., $\phi_1 = \phi_{init}$, $\ddot{\phi}_1 = \ddot{\phi}_{init}$, $\phi_T = \phi_{end}$, $\ddot{\phi}_T = \ddot{\phi}_{end}$, whereas the other $4(t - 2)$ constraints represent continuity velocity and acceleration conditions for poses $t = 2, \dots, T - 1$, namely $\dot{\phi}_{t-1} = \dot{\phi}_t$, $\ddot{\phi}_{t-1} = \ddot{\phi}_t$. It is noteworthy that also velocities at the initial and final positions are specified, by imposing two additional cubic splines at the beginning and end of the trajectory.

The trajectory Θ may undergo a resampling process to smooth the discrete step-by-step sequence obtained through the iterative optimisation of (1) or (3). However, considering a subset of Θ may lead to a resultant robot's trajectory slightly different from the original one. Examples with a sampling of order 2 and 8 are shown in Fig. 2.

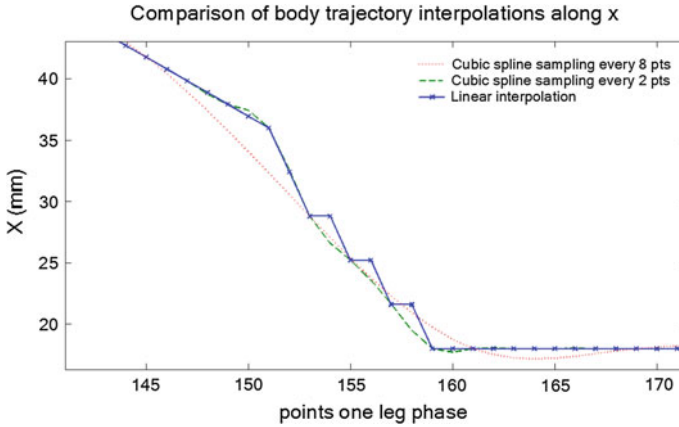


Fig. 2 In blue the optimization result; in green an order 2 sampling; in red an order 8 sampling

A stability and feasibility test on the resultant curve is run after the interpolation. If the final trajectory is not feasible or unstable, the sampling interval is reduced and the spline is recomputed.

3.2 Swing Leg Trajectory Generation

One locomotion cycle is made up of a fixed leg displacement sequence. In order to move legs in the Workspace, each leg is displaced for a given length and direction. We refer to CT as the time to complete one locomotion cycle, and to β as the normalised time period in which legs are supporting the body shift. In order to ensure smoothness in velocity and acceleration, fifth (for x and y axes in WF) and sixth (for the z axis) order polynomials are used to model each step. To describe the tip x and y coordinates in WF, we use:

$$\begin{aligned}
 \rho(t) &= c_0 + c_1t + c_2t^2 + c_3t^3 + c_4t^4 + c_5t^5, \\
 \dot{\rho}(t) &= c_1 + 2c_2t + 3c_3t^2 + 4c_4t^3 + 5c_5t^4, \\
 \ddot{\rho}(t) &= 2c_2 + 6c_3t + 12c_4t^2 + 20c_5t^3,
 \end{aligned} \tag{6}$$

where coefficients c_i , with $i = 0, \dots, 5$, are computed depending on positions, velocities and accelerations defined at the tip *lift off* t_{lo} and *touch down* t_{td} time instants (these events are detected using contact sensors on leg tips). If we rewrite in matrix form:

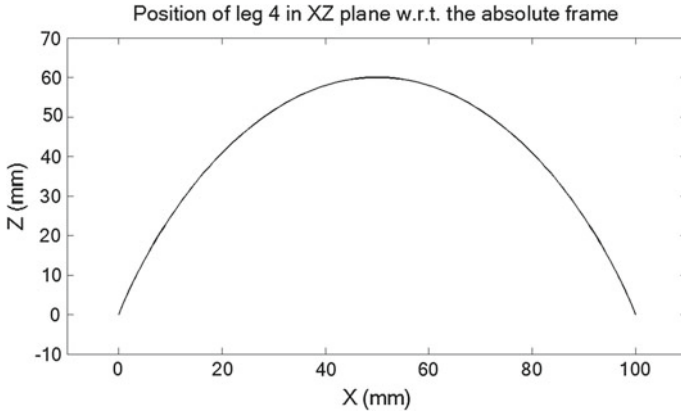


Fig. 3 Swing leg trajectory in the xz plane, for a 100 mm tip displacement and a 60 mm step height

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & t_{id} & t_{id}^2 & t_{id}^3 & t_{id}^4 & t_{id}^5 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2t_{id} & 3t_{id}^2 & 4t_{id}^3 & 5t_{id}^4 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6t_{id} & 12t_{id}^2 & 20t_{id}^3 \end{bmatrix},$$

$$x = [c_0 \ c_1 \ c_2 \ c_3 \ c_4 \ c_5]^T,$$

$$b = [\rho(t_{lo}) \ \rho(t_{id}) \ \dot{\rho}(t_{lo}) \ \dot{\rho}(t_{id}) \ \ddot{\rho}(t_{lo}) \ \ddot{\rho}(t_{id})]^T,$$

then the solution x can be computed as $x = A^{-1}b$. Since each leg is assumed to keep contact with the ground during the support phase, the initial and final velocities and accelerations are set to zero. The final position is computed as the initial position plus the required leg displacement. The swing phase is defined according to β and CT . The lifting event is considered to occur at $t_{lo} = 0$, whereas the touch down time is $t_{td} = (1 - \beta)CT$. A sixth order polynomial is used to describe the motion of the leg tip along the z coordinate of WF. The principle used to compute the curve is very similar to what is done in (5). However, an additional coefficient is added, which is used to define a *via point*, i.e., the maximal step height that is reached in the middle of the transfer phase, $t_m = t_{td}/2$.

An example of the resultant swing trajectory in time and space is shown in Fig. 3. It is noteworthy that, in order to simplify the representation, the transfer leg displacement is set only along the robot heading direction.

3.3 Constraints

Robot motion is constrained by joint limits (as well as joint positions, velocities and accelerations) and body stability. Ground contact conditions are explicitly defined assuming a rigid terrain (i.e., a unilateral reaction force).

Computation of the CoG. The computation of CoG is performed according to the configuration of the robot's legs with respect to BF. For each leg, both link positions and orientations are identified using either the inverse (during planning) or the direct (during execution, using linear displacement sensors on hydraulic actuators) geometric model. Once the configuration of all the links of the leg L_j is known with respect to HF_j, the corresponding leg CoG_{L_j} is estimated as:

$$CoG_{L_j} = \frac{\sum_{l=1}^{\lambda} S_l M_l}{M_{L_j}}, \quad (7)$$

where λ is the number of the L_j links, S_l and M_l are the first momentum and the mass of link l , respectively, whereas M_{L_j} is the total leg mass. It is noteworthy that this computation considers both actuators fixed and mobile parts. In (7), M_{L_j} is given by:

$$M_{L_j} = \sum_{l=1}^{\lambda} M_l. \quad (8)$$

Finally, all CoG_{L_j} are expressed with respect to BF and linearly combined with the CoG of the body to compute the *robot* CoG:

$$CoG = \frac{M_{body} S_{body} + \sum_{j=1}^4 M_{L_j} CoG_{L_j}}{M_{tot}}, \quad (9)$$

where M_{tot} is the total mass computed as:

$$M_{tot} = M_{L_1} + M_{L_2} + M_{L_3} + M_{L_4} + M_{body}. \quad (10)$$

In (10), M_{body} is the first momentum of the robot body with respect to BF. This vector is non-zero since BF is placed at the centroid of the body and not at the barycentre. M_{body} refers to the weight of the robot platform supported by the legs plus the hydraulic and control systems mounted on it.

Static Stability Check. As anticipated, the SSM is used [6]. Every time instant, the CoG is computed according to (7) and then projected onto the horizontal plane. Stability is assessed by checking whether the projection CoG_{proj} lies within the support polygon (Fig. 4). Numbering vertexes of the support polygon in clockwise order, P_i , $i = 1, \dots, 4$, then four vectors v_i are defined, such that $v_i = P_{i+1} - P_i$. From each vertex P_i an additional vector u_i pointing to the CoG projection is defined as

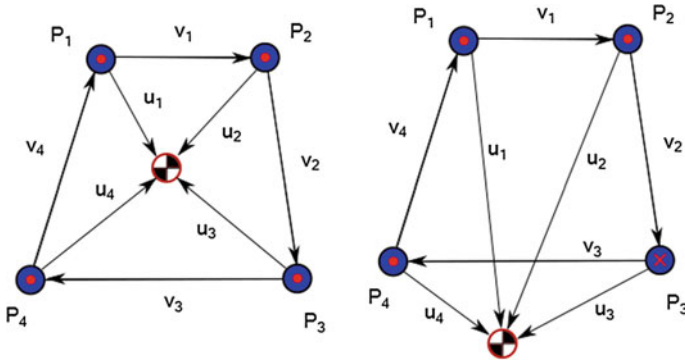


Fig. 4 Static stability checking. *Left* the robot is stable. *Right* the robot is unstable

$u_i = CoG_{proj} - P_i$. If, for each vertex P_i , we compute the vector product $vp_i = u_i \times v_i$, then the position of the CoG is determined on the basis of the vp_i components. If $|vp_i| = 0$ for any i , the robot is unstable because the CoG_{proj} lies on one edge, and therefore the condition $SSM = 0$ holds; although it may be still considered a stable point, real-world robot dynamics effect may disrupt stability. If the z component of vp_i is positive (negative), then CoG_{proj} lies on the right (left) of edge v_i . As a consequence, if vp_i norms are non-zero and the z components of all the vectors are positive, then the robot is *stable* because the CoG_{proj} is within the support polygon. Otherwise, if one null norm exists or one z component is negative, then the robot is *unstable*. The conditions hold for any support polygon, either with three or four legs in support.

3.4 Body Trajectory Planning During One Step

The developed body motion algorithm extends the ESM criterion by taking into account the legs configuration. The main robot's body always stops in the highest reachable ESM poses with four legs in support. The cycle time CT is divided in four equal-length stages (i.e., *leg times*), with the aim of improving the control on the body trajectory planning during the displacement of a target leg, namely (1) *initial body motion*, (2) *leg lifting*, (3) *leg lowering* and (4) *final body motion* (Fig. 5).

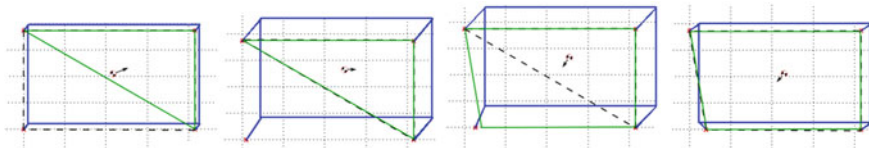


Fig. 5 Phases (1) to (4), from *left* to *right*. In continuous *green*: the support boundary considered to improve ESM. In *dashed black*: the real support polygon for robot stability. The *small circle* is the CoG

During phases (1) and (4), the robot stands with four legs in support, whereas during phases (2) and (3) only three legs are in contact with the ground. The robot’s body trajectory is computed as explained in Sect. 3.1. The trajectory improves the ESM of phases (2) and (4) during phases (1) and (3), respectively.

Phase (1) ensures a stability guard. At the leg lifting, the robot’s body is very close to being unstable, either because it is on one support polygon edges or it has a very small SSM. Therefore, the robot’s body is moved in advance to increase the ESM defined on the future three legs boundary. Phase (2) moves the body according to the triangular support boundary. Phase (3) anticipates body motion toward the best position with four legs in support. Phase (4) completes the cycle time moving the robot’s body after touch down. The body is moved to the highest ESM configuration corresponding to the initial pose for the next cycle. Any leg sequence can be used given the initial and final body motions in the four legs support phases, i.e., phases (1) and (4).

3.5 Walking with Obstacles Along the Path

In this paper, we focus on conveyance tasks constrained to cross *traversable* obstacles. The complex, non-symmetric shape of the leg Workspace is simplified to a

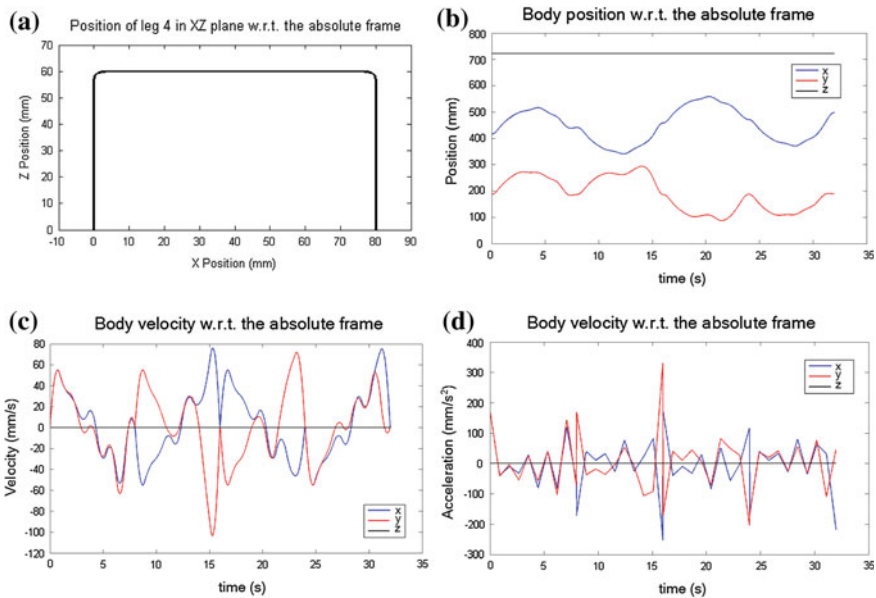


Fig. 6 a Leg trajectory in the xy plane when crossing traversable obstacles; b–d example of body position, velocity and acceleration during a motion cycle

parallelepiped volume characterised by a 150 mm height, a 400 mm length and 350 mm width rectangular base. The step length is bounded in order to keep the leg tip within the rectangular area in the xy plane, whereas the step height is set to the maximum feasible value. Since the robot is not provided with exteroceptive sensors, each leg is lifted as high as possible during the swing phase. In order to reduce the likelihood of collisions during phases (2) and (3), and to limit contact detection only during phase (3), the transfer leg trajectory shape is modified with respect to Fig. 3. The trajectory is depicted in Fig. 6a. It is obtained using two fifth order polynomials in the form of (5) and a first order polynomial to keep the leg tip height along the z axis, as well as one fifth order polynomial along x and y axes computed for a reduced time slice with respect to the transfer time.

4 Experimental Validation

The ESM-based statically stable walking gait is tested using MATLAB, with the aim of checking the algorithm soundness and the final body motion. Complete cycles are executed using different leg sequences and displacing the foot tips in various directions and for arbitrary distances. Tests are conducted at first on flat ground to simplify the analysis. A reference example of body position, velocity and acceleration trajectories during one full motion cycle are shown in Figs. 6b, c and d, respectively. Cycle time is set to $CT = 32$ s, $\beta = 0.875$ and the leg sequence is 4, 2, 3 and 1 (Fig. 1c). Legs are displaced along the x axis for a distance of 80 mm. Trajectories are shown with respect to the absolute frame oriented as the body frame and placed at the leg tip position of the RR leg at the beginning of the motion.

4.1 Body Motion and Gait Stability

As shown in Fig. 6b, the z component of the body position remains constant during the motion. Velocities in x and y are null at the end of each single swing leg time, i.e., 8, 16, 24 and 32 s (Fig. 6c). At the end of the motion, the robot's body advances 80 mm along x and comes back to the original position along y (Fig. 6b). In the final configuration, the body reaches the centroid of the support boundary, which coincides with the support polygon as the motion lies on a horizontal plane. The robot keeps its CoG always within the support polygon displacing the body according to the ESM criterion. It is noteworthy that the planned trajectory sensibly depends on the step size chosen during the gradient-based search (1) and (3) and the sampling performed on the trajectory before the cubic spline interpolation.

Figure 7a shows how the ESM varies along the four phases outlined in Sect. 3.4. During phase (1) the ESM decreases to ensure the stability guard at the beginning of phase (2). In phase (2), the ESM increases like in phase (4) because the support boundary used in the body planning corresponds to the effective one. During phase

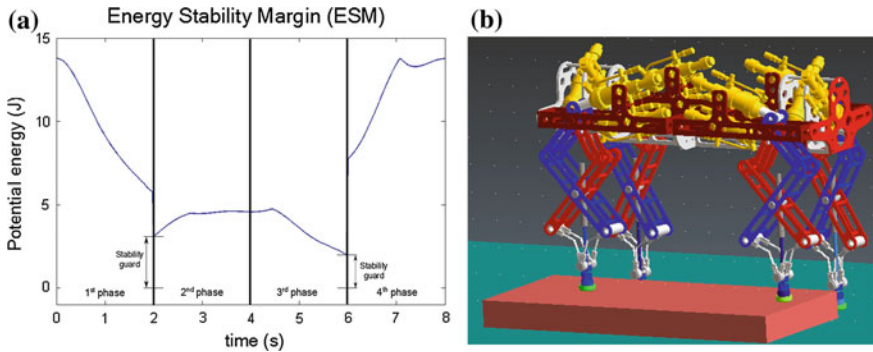


Fig. 7 **a** Evolution of the ESM during the four phases defined for the swing of each leg; **b** obstacle crossing with side legs

(3), the ESM decreases due to the anticipating action to get the body closer to the final configuration. As mentioned before, the stability guards and the body velocity are strictly related to the step size definitions. Furthermore, the final ESM evolution depends on the trajectory sampling performed before the spline interpolation. Since the interpolation of a subset of the trajectory leads to a variation in the body position (Fig. 2), the ESM might not always decrease during phases (1) and (3) and increase during phases (2) and (4). Figure 7a shows this phenomenon, with a focus on phases (3) and (4).

4.2 Robot Motion on Slopes

The gait has been tested on various slopes. Given the limits imposed by the target conveyance task, the leg workspace limits the robot climbing capabilities. Two angles, roll and pitch, are used to define the plane inclination along x and y with respect to WF.

The robot capability to walk on slopes is strictly related to the desired step height and length. The main constraint is induced by the height of the leg tip. Apart from RR, which is located close to the bottom of its workspace, the other legs work at higher positions according to the ground orientation. In addition, considering that the body length is almost twice the width, FL and FR work in higher positions with respect to RL, using the same values for the pitch and roll angles. However, if step height and length were the same for all legs, the robot would be able to walk on steeper slopes along the y axis. For example, setting the step height to 100 mm and the step length to 80 mm, the robot is able to walk on slopes (defined by roll and pitch) between $(-10 \div 10, 0)^\circ$, $(0, -5 \div 5)^\circ$ and $(-5 \div 5, -3 \div 3)^\circ$. The last robot configuration is depicted in Fig. 8, which shows how the robot's body is shifted with respect to the vertical position in order to increase the ESM in the last leg phase.

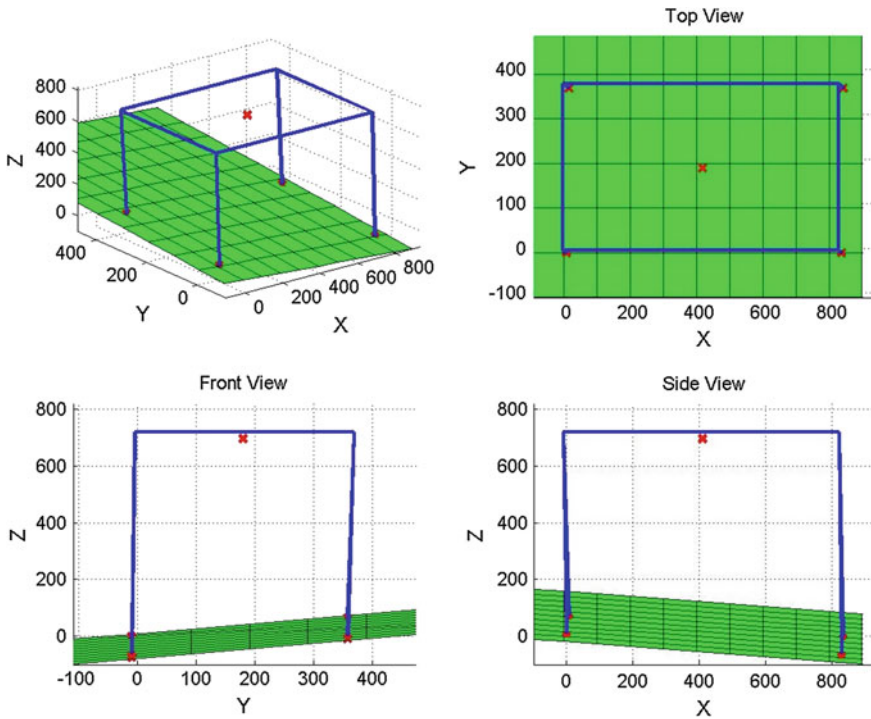


Fig. 8 Final robot configuration on a generic slope. The *red cross* close to the body frame represents the CoG, whereas the *red crosses* on the slope are the leg tips

4.3 Walking on Obstacles

In order to simulate a task where the robot must walk on an obstacle, the planner has been embedded in a Simulink control scheme. The robot dynamic model is implemented in RecurDyn, a multi-body robot simulator [17].

As discussed in Sect. 3.5, collisions are detected only during phase (3), using the contact sensors placed on the foot soles. The robot configuration is initialised to stand with all legs in support vertically aligned to the hips. An obstacle of regular shape is placed in front of the robot with the aim of hindering the motion of the right robot legs. The obstacle is a parallelepiped of 400 mm width, 100 mm height and 500 mm depth. The height is set to a value that can be considered the maximum allowed for a feasible motion. Since the leg can be displaced along z of 150 mm, if the object is 100 mm thick and not all the legs are placed on the obstacle, the legs in transfer lift the tip 50 mm from the object surface. This distance can be considered a sort of safety gap in case the object presents some unexpected irregularities. In case all the legs are on the obstacle, it is possible to keep the original absolute height of the body or lift it to keep a constant distance from the surface of contact. The robot is moved

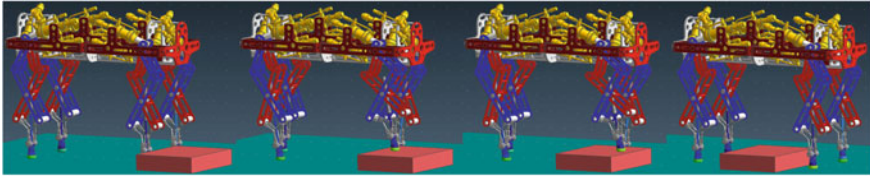


Fig. 9 Obstacle crossing with one leg. Three events characterize the obstacle crossing: step on, walk on and step off

forward with a step of 200 mm and a step height of 150 mm. The robot proves to be able to step on, traverse and step off from the obstacle with one leg (Fig. 9).

The obstacle depth has been extended in order to involve two legs on the same side. Also in this case the robot successfully crosses the object: it steps on the obstacle with one leg first, then it walks with both side legs on the object surface, it steps off from the obstacle with the front leg and then with the rear one. Figure 7b shows the robot with the right legs on the object.

5 Conclusions

This paper describes an architecture to achieve a statically stable gait for quadruped robots in charge of performing conveyance tasks when traversing uneven terrain. The robot uses only proprioceptive sensors. The focus is on the body motion, which is designed to increase the ESM according to the robot stability as well as geometric and kinematic constraints. Fixing the footstep sequence, the walking gait is tested on both flat ground and slopes. Furthermore, tests are carried out on environments with obstacles of given shape and size. Tests have been carried out in simulation. An initial experimental campaign has been conducted on the real platform and is subject of current work (Fig. 10).

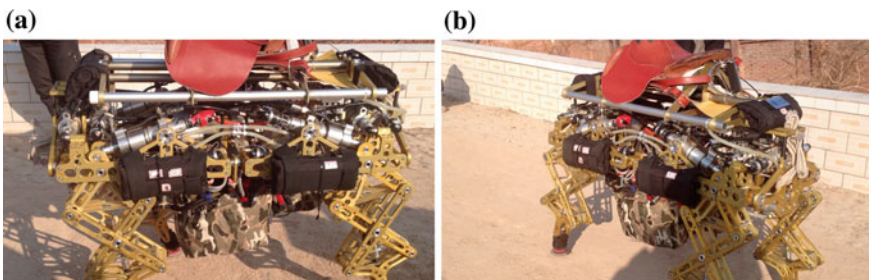


Fig. 10 Two phases of the walking strategy implemented on Baby Elephant

References

1. Wooden, D., Malchano, M., Blankespoor, K., Howardy, A., Rizzi, A., Raibert, M.: Autonomous navigation for Big Dog. In: Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA 2010), Anchorage, Alaska, USA, May 2010
2. Semini, C., Khan, H., Frigerio, M., Boaventura, T., Focchi, M., Buchli, J., Caldwell, D.: Design and scaling of versatile quadruped robots. In: Proceedings of the 15th International Conference on Climbing and Walking Robots CLAWAR, Baltimore, Maryland, USA, July 2012
3. Murphy, M., Saunders, A., Moreira, C., Rizzi, A., Raibert, M.: The Little Dog robot. *Int. J. Robot. Res.* **30**(2), 145–149 (2011)
4. McGhee, R., Frank, A.: On the stability properties of quadruped creeping gaits. *Math. Biosci.* **3**, 331–351 (1968)
5. Zhang, C., Song, S.-M.: Stability analysis of wave-crab gaits of a quadruped. *J. Robot. Syst.* **7**(2), 243–276 (1990)
6. Messuri, D., Klein, C.: Automatic body regulation for maintaining stability of a legged vehicle during rough-terrain locomotion. *IEEE Robot. Autom. Mag.* **1**(3), 132–141 (1985)
7. Hirose, S., Tsukagoshi, H., Yoneda, K.: Normalized energy stability margin and its contour of walking vehicles on rough terrain. In: Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA 2011), Seoul, South Korea, May 2011
8. Garcia, E., Estremera, J., de Santos, P.G.: A comparative study of stability margins for walking machines. *Robotica* **20**, 595–606 (2002)
9. McGhee, R.: Some finite state aspects of legged locomotion. *Math. Biosci.* **2**(1–2), 67–84 (1968)
10. Hirose, S., Nose, M., Kikuchi, H., Umetani, Y.: Adaptive gait control of a quadruped walking vehicle. *Int. J. Robot. Res.* **3**(2), 113–133 (1984)
11. Bares, J., Wettergreen, D.: Dante II: technical description, results and lessons learned. *Int. J. Robot. Res.* **18**(7), 621–649 (1999)
12. Tsukagoshi, H., Hirose, S.: Teaching and programming for robots. The proposal of the intermittent crawl gait and its generation. *J. Robot. Soc. Jpn.* **17**(2), 301–309
13. Zhang, L., Ma, S., Inoue, K.: Several insights into omnidirectional static walking of a quadruped robot on a slope. In: Proceedings of the 2006 IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS 2006), Beijing, China, Sept 2006
14. Pongas, D., Mistry, M., Schaal, S.: A robust quadruped walking gait for traversing rough terrain. In: Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA 2007), Rome, Italy, Apr 2007
15. Johnson, K.: *Contact Mechanics*. Cambridge University Press, Cambridge, MA (1987)
16. Schafer, M.: *Computational Engineering: Introduction to Numerical Methods*. Springer, Heidelberg-Berlin (2006)
17. FunctionBay GmbH: Recurdyn. <http://www.functionbay.de/> (2013)

Part VII
Localization and Path Planning

A Simple and Efficient Path Following Algorithm for Wheeled Mobile Robots

Goran Huskić, Sebastian Buck and Andreas Zell

Abstract A heuristic path following algorithm for wheeled mobile robots is presented. This approach is based on the orthogonal projection to the path and exponential functions for lateral and longitudinal control. It allows smooth and stable navigation in dynamic and cluttered environments, and does not depend on the robot's kinematics. The results are experimentally demonstrated using three different kinematic configurations: omnidirectional, Ackermann- and two-steering.

Keywords Mobile robots · Path following · Control · Kinematics

1 Introduction

The goal of path following is reducing the distance and orientation error between a robot and a geometric reference path. Difference between path following and trajectory tracking is made clear by Aguiar et al. in [1].

Many path following algorithms for wheeled mobile robots have already been developed. A path following algorithm for omnidirectional drives based on model predictive control is presented by Kanjanawanishkul and Zell in [7]. A thorough study on feedback control of car-like vehicles is made by De Luca et al. in [4]. Some work on unicycle control can be found, e.g., in [15] and [16]. Path following solution for skid-steering vehicles using sliding mode control is proposed in [10]. Path following for a multi-axle vehicle is presented in [9]. Line-of-sight path following

G. Huskić (✉) · S. Buck · A. Zell
Chair of Cognitive Systems, Computer Science Department,
University of Tübingen, Sand 1, 72076 Tübingen, Germany
e-mail: goran.huskic@uni-tuebingen.de
URL: <http://www.cogsys.cs.uni-tuebingen.de/>

S. Buck
e-mail: sebastian.buck@uni-tuebingen.de

A. Zell
e-mail: andreas.zell@uni-tuebingen.de

of Dubins paths for different autonomous systems is developed in [6]. A time optimal path following solution for mobile robots with independently steerable wheels can be found in [13]. An efficient model-independent algorithm was proposed by Egerstedt et al. in [5]. This approach was extended by Maček et al. in [11] with an additional velocity control. A comprehensive classification of different mobile robot configurations was done by Campion et al. in [3].

Mojaev and Zell have developed an efficient path following algorithm for a differential drive in [12]. This approach was utilized for omnidirectional drives by Li et al. in [8]. In this article, we are proposing an approach to provide safer, more accurate and stable path following performance, outperforming the algorithm proposed in [12], and offering more features in the linear velocity control than the algorithm proposed in [11]. Our contribution is as follows:

- proposing a novel heuristic linear velocity control,
- introducing this linear velocity control to the approach proposed in [12],
- utilizing the approach for Ackermann- and two-steering drives,
- experimentally evaluating the algorithm performance on
 - an omnidirectional drive (in a competition and in laboratory conditions),
 - Ackermann- and two-steering drives (laboratory conditions).

This article is organized as follows. In Sect. 2 the algorithm proposed in [12] is presented. In Sect. 3, we propose a new longitudinal control. Section 4 describes the proposed path-following approach. Experimental results are presented in Sect. 5. Section 6 concludes.

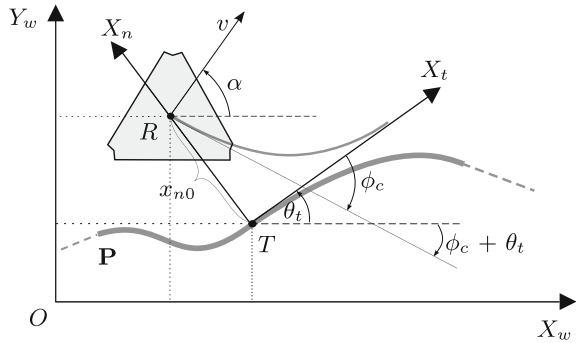
2 Orthogonal Projection Based Control

To show the principle of the approach proposed by Mojaev and Zell in [12], and further developed by Li et al. in [8], Fig. 1 should be observed.

The world frame is defined with the axes X_w, Y_w and the origin O . The robot's center of mass can be seen as point R . The geometric reference path is denoted with P , and the robot's orthogonal projection to the path is point T . The moving reference frame $\{P\}$ is defined with its origin at point T . Abscissa X_t and ordinate X_n are unit tangent and unit normal vectors at point T , respectively. Along the axis X_n , the orthogonal distance from the robot to the path is denoted with x_n , and along the axis X_t , tangential distance with x_t . The tangential angle of the path at point T is denoted with θ_t . This is, in fact, the angle between the moving reference frame and the world frame. The robot's translational velocity is represented with v , and the robot's moving direction angle is α . If we denote x_{n0} as the distance from the robot to the path at point T , it is possible to define the following exponential law

$$x_n = x_{n0} e^{-kx_t}, \quad (1)$$

Fig. 1 Orthogonal projection based path following



where k is a positive constant which regulates the speed of convergence. The orthogonal distance x_n is now an exponential function of the tangential distance, with an initial value x_{n0} at point T. If we denote the angle of the exponential function's tangent as ϕ_c , the following applies: $\tan \phi_c = \frac{dx_n}{dx_t}$. Now, this angle can be expressed as

$$\phi_c = \arctan(-kx_n). \tag{2}$$

If we denote the desired velocity with v_n , the robot's model in the moving reference frame is then

$$\begin{aligned} \dot{x}_t &= v_n \cos \phi_c, \\ \dot{x}_n &= v_n \sin \phi_c. \end{aligned} \tag{3}$$

In [12], a differential drive was controlled using PD control with ϕ_c as error input. Rotational velocity ω was expressed in the moving reference frame as $\omega = K_p \phi_c + K_d \frac{d\phi_c}{dt}$, where K_p and K_d are proportional and derivative gains, respectively.

Omnidirectional drives are able to move in any direction, regardless of their orientation. Because of this, it was possible for Li et al. in [8] to directly use ϕ_c as direction angle, and use separate PD control for rotation. Robot's control model in the world frame is then

$$\begin{aligned} v_x &= v_n \cos \alpha, \\ v_y &= v_n \sin \alpha, \end{aligned} \tag{4}$$

where $\alpha = \phi_c + \theta_t$, and v_x, v_y are robot's translational velocity components. Rotation control is then $\omega = k_1(e_\theta + k_2\dot{e}_\theta)$, where k_1, k_2 are proportional and derivative gains, respectively, and $e_\theta = \theta_d - \theta$ the error between the desired orientation θ_d and measured orientation θ . In the case of nonholonomic robots, θ corresponds with α . Stability analysis of this algorithm can be found in [8].

By using this formulation, the inherent singularity when $x_n = \frac{1}{c_{max}}$ can be avoided, where c_{max} is the maximum curvature of the path. This problem was analysed by Samson in [14].

3 Longitudinal Control

The described algorithm's main limitation is its lack of linear velocity control, since the desired velocity v_n is a positive constant. At higher velocities, the performance would greatly decrease in the following ways:

- at highly curved parts of the path, the tracking accuracy decreases,
- in cluttered and/or dynamic environments, danger of colliding with obstacles increases,
- abruptly going from full velocity to zero at the goal position increases the final positioning error, the danger of tipping over, and damages the actuators,
- driving fast and rotating at the same time introduces unwanted dynamic effects (applies only to omnidirectional drives).

These problems can be solved with a heuristic approach described in the following text.

3.1 Path Curvature

The following idea is similar to the one proposed in [11], but with major differences. First, we define path curvature.

If a 2D curve P is parametrized by the arclength s , then $x_d = p(s)$, and $y_d = q(s)$ are the x and y coordinates of the desired path, respectively. If we denote $p' = \frac{\partial p}{\partial s}$, $q' = \frac{\partial q}{\partial s}$, and $p'' = \frac{\partial^2 p}{\partial s^2}$, $q'' = \frac{\partial^2 q}{\partial s^2}$, then the curvature κ can be computed as

$$\kappa = \frac{p'q'' - p''q'}{(p'^2 + q'^2)^{\frac{3}{2}}}, \quad (5)$$

which is further detailed, e.g., by Toponogov in [17].

Now, we define a free control parameter which we call *path-look-ahead distance*, an arbitrarily chosen distance in path coordinates. It starts at the orthogonal projection T , and ends at most at the goal position. So, if we denote the path-look-ahead distance with l_κ , then $l_\kappa \in [0, s_G - s_T]$. Here, s_T and s_G are the point T and the goal position in path coordinates, respectively. The proposed penalty factor using curvature measure for a discretized path is then

$$\xi_\kappa = e^{-K_\kappa \sum_{i=i_T}^{i_\kappa} |\kappa_i|}, \quad (6)$$

where $K_\kappa > 0$ is a free parameter which determines the speed of convergence, i_T is the index of orthogonal projection T , i_κ is the index of the point positioned at the path-look-ahead distance from the projected point, and κ_i is the curvature in the

current point with the index i . Using this penalty factor to scale the linear velocity as $v = v_n \xi_{\kappa}$, tracking accuracy remains high even in the areas of high path curvature.

3.2 Distance to Obstacles

To be sure that the robot will avoid collisions with static and dynamic obstacles in its environment, a safety measure should be considered, similar to the proposed curvature penalty factor. This penalty factor is proposed as

$$\xi_o = e^{-\frac{K_o}{d_o}}, \quad (7)$$

where d_o is the distance to the obstacles of interest, and K_o is a convergence parameter. This distance to the obstacles of interest can be chosen in different ways. In this work, we use the distance to the nearest obstacle, since it provides satisfying results. Using this penalty factor as $v = v_n \xi_o$, danger of colliding with obstacles is greatly reduced, and the navigation has a much higher safety level.

3.3 Distance to the Goal Position

In order to avoid dangerous sudden braking at the end of the path, it is necessary to introduce asymptotic deceleration, as the robot approaches the goal. This penalty factor can be defined as

$$\xi_g = e^{-\frac{K_g}{d_g}}, \quad (8)$$

where $K_g > 0$ is a parameter which determines the convergence of the asymptote, and d_g is the distance to the goal in path coordinates. By using this penalty factor as $v = v_n \xi_g$, there is no danger of tipping over, tracking accuracy remains preserved at the goal position, and the actuators are not stressed with extreme values.

3.4 Angular Velocity

An omnidirectional robot's linear and angular velocity are kinematically independent, but at higher velocities, dynamic effects occur. If these effects are not explicitly accounted for, they might lead to unwanted behaviour.

We then propose the following rotation penalty factor

$$\xi_{\omega} = e^{-K_{\omega}|\omega|}, \quad (9)$$

where $K_\omega > 0$ regulates the convergence, and the linear velocity is then $v = v_n \xi_\omega$. If the robot needs to change its orientation during the drive, it will shortly decelerate to keep the tracking accuracy, until the rotation is complete. This way, the tracking accuracy is preserved, and there are no unwanted dynamic effects.

3.5 Final Heuristic Longitudinal Control

By using all of the proposed penalty factors for the linear velocity, the final expression is then given as

$$v = v_n \exp \left(- \left(K_\kappa \sum_{i=i_p}^{i_k} |\kappa_i| + K_\omega |\omega| + \frac{K_o}{d_o} + \frac{K_g}{d_g} \right) \right). \quad (10)$$

This expression gives a compact and efficient solution for linear velocity control, which allows the robot to cope with the problems of complex paths in cluttered and dynamic environments.

4 Orthogonal-Exponential Approach

Now we introduce the proposed longitudinal control solution to the path following control presented in Sect. 2. This integral control algorithm we address as *orthogonal-exponential approach*. The robot's model in the moving reference frame is then

$$\begin{aligned} \dot{x}_t &= v \cos \phi_c, \\ \dot{x}_n &= v \sin \phi_c, \end{aligned} \quad (11)$$

where v is defined as in Eq. 10, and $\phi_c = \arctan(-kx_n)$, as in Eq. 2.

Omnidirectional drive can now be directly controlled, similar as in [8]. We use the error angle ϕ_c in robot coordinates, such that ${}^R\phi_c = \phi_c + \theta_t - \theta$.

For the Ackermann- and two-steering drive we propose a PID control with ${}^R\phi_c$ as error input: $\varphi = K_P {}^R\phi_c + K_I \int_0^T {}^R\phi_c dt + K_D \frac{d^R\phi_c}{dt}$, where K_P, K_I, K_D are proportional, integral and derivative gains, respectively. The two-steering drive is assumed to be symmetrical: $\varphi_r = -\varphi_f$, where φ_f and φ_r are front and rear axle steering angles, respectively. The resulting steering is then: $\varphi = \varphi_f - \varphi_r$. With $\{K_P = 1, K_I = 0, K_D = 0\}$, i.e. using a P controller, the steering angle control becomes $\varphi = {}^R\phi_c$. This choice of parameters showed satisfying results and was used in the experiments.

To analyse the stability of the whole system, we follow the work of [8] and choose the Lyapunov candidate function as

$$V = \frac{1}{2}x_n^2 + \frac{1}{2}\phi_c^2, \quad (12)$$

and then the derivative can be found as

$$\begin{aligned} \dot{V} &= x_n \dot{x}_n + \phi_c \dot{\phi}_c \\ &= x_n v \sin(\arctan(-kx_n)) + \arctan(-kx_n) \frac{v \sin(\arctan(-kx_n))}{1 + (kx_n)^2}. \end{aligned} \quad (13)$$

Using the properties of the functions \arctan and \sin , we conclude that $\dot{V} < 0$ and that the equilibrium ($x_n = 0, \phi_c = 0$) is globally asymptotically stable.

The robot's linear velocity v , computed as in Eq. 10, is positive and doesn't vanish with time. The velocity stays in the interval $v \in (0, v_n)$, regardless of the control parameter values. This implies that the global asymptotic stability is preserved by using the longitudinal control proposed in Sect. 3.

5 Experimental Results

The proposed orthogonal-exponential approach was experimentally evaluated in two different manners:

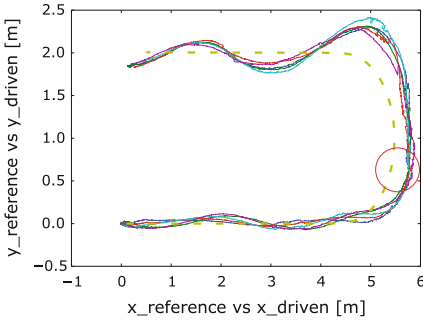
- with an omnidirectional drive, we participated in an international robotics competition SICK Robot Day 2014 and won the 2nd place,
- in our laboratory, path following experiments were conducted using an omnidirectional drive, Ackermann- and two-steering drive.

The robots we used can be seen in Fig. 3. More about the competition, where the proposed algorithm was used, can be found in [2].

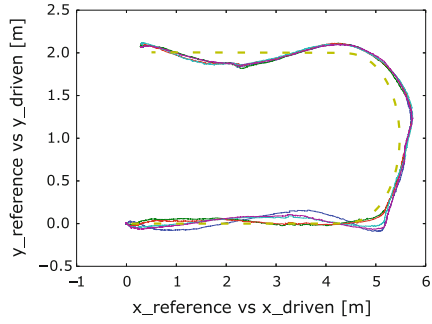
In the following text, results from the laboratory experiments with a U-shaped curve will be explained in detail. All the results in the tables were averaged over 5 experiments. The errors and the velocities are plotted relative to the path. In the area illustrated with a red circle in the following figures, there was a planted obstacle. It was hit every time while driving with the original approach, and avoided every time using the proposed approach. This emphasizes the advantage of inherent safety measures of the proposed approach.

5.1 Omnidirectional Robot

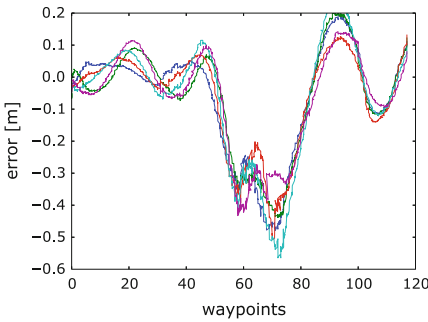
While following the path, the omnidirectional robot simultaneously changed its orientation towards a point set on the planted obstacle. The desired velocity was chosen as a threshold at which the original algorithm shows undesired behaviour ($v_n = 1.3$ m/s). A comparison between the original approach and the proposed approach



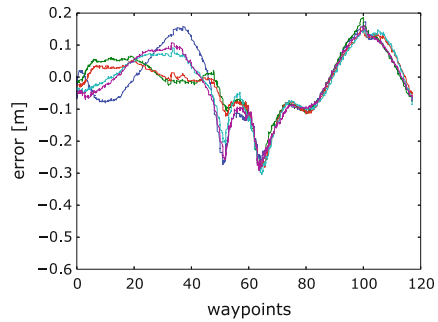
(a) Path following using the original approach



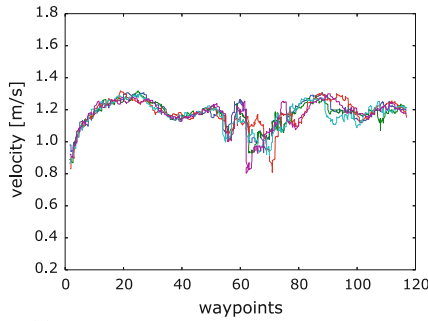
(b) Path following using the proposed approach



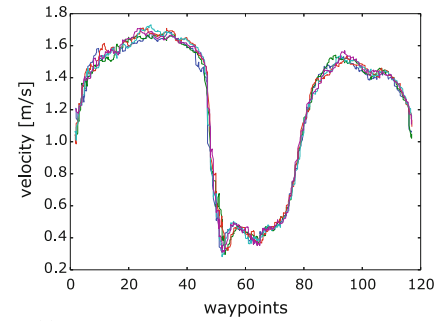
(c) Following error of the original approach



(d) Following error of the proposed approach



(e) Actual velocity using the original approach



(f) Actual velocity using the proposed approach

Fig. 2 Five runs (different colors) of following a U-curve with an omnidirectional robot, where $v_n = 1.3 \text{ m/s}$. The golden dashed line in **a** and **b** represents the desired path

can be seen in Fig. 2. Table 1 shows the improvement the proposed approach offers, 40.7% for the maximum following error and 19.06% for the average error. The proposed approach is more reliable with a 37.51% decrease in standard deviation. The velocities are comparable (Fig. 3).

Table 1 Proposed versus original algorithm: omnidirectional drive

Omni	e_{max} [cm]	e_{avg} [cm]	σ [cm]	v_{max} [$\frac{m}{s}$]	v_{avg} [$\frac{m}{s}$]
Original	48.44	-7.7	17.14	1.32	1.16
Extended	28.73	-6.23	10.71	1.73	0.93
Δ [%]	40.7	19.06	37.51	23.69	20.12

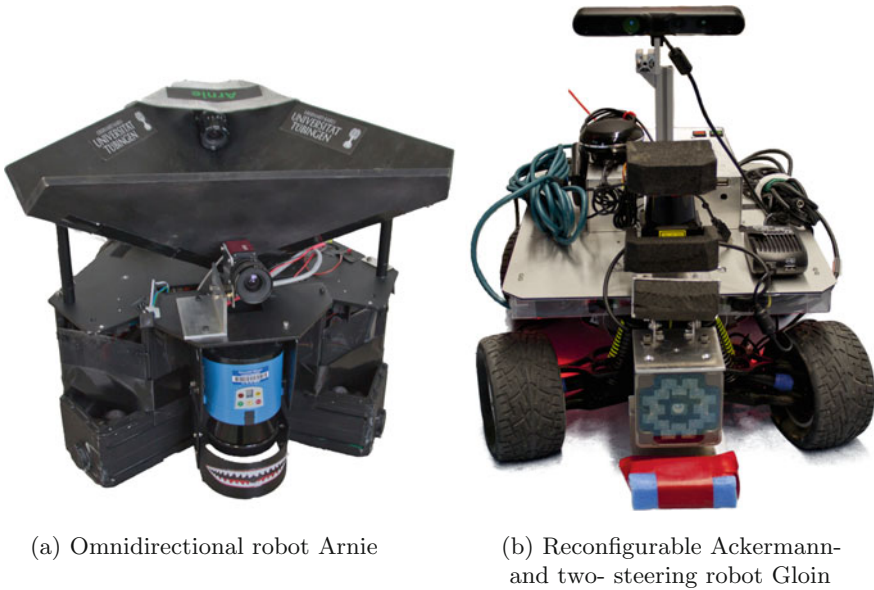


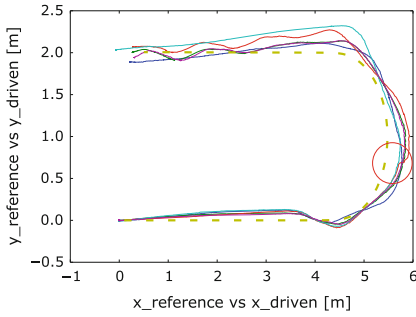
Fig. 3 Robots used for the experiments presented in this paper

5.2 Two-Axle-Steering Robot

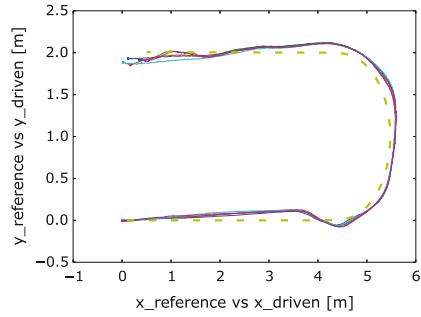
A comparison between the original approach with the command velocity $v_n = 0.7$ m/s, and the proposed approach for the two-axle-steering configuration can be seen in Fig. 4. Table 2 shows how the proposed approach offers an improvement of 52.6% for the maximum following error, 28.13% for the average error and 35.74% for the standard deviation.

5.3 Ackermann-Steering Robot

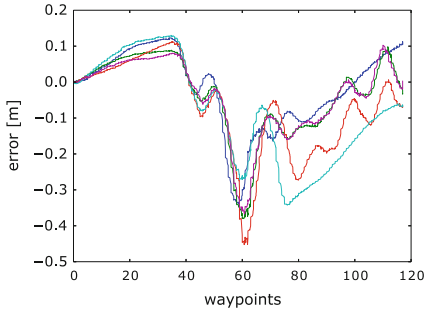
The results of the same experiment with the Ackermann-steering configuration can be seen in Fig. 5. As presented in Table 3, the proposed approach has a 51.85%



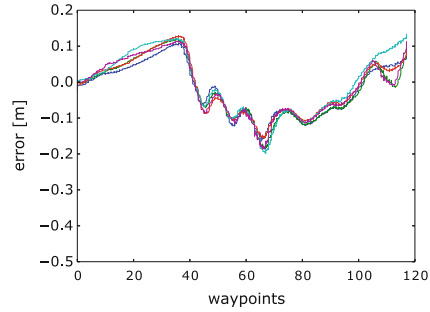
(a) Path following using the original approach



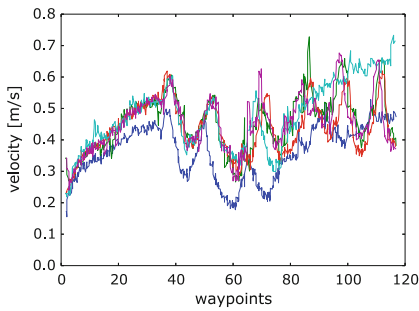
(b) Path following using the proposed approach



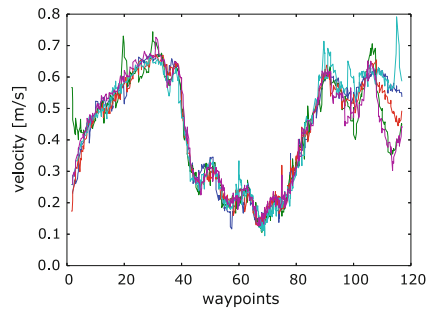
(c) Following error of the original approach



(d) Following error of the proposed approach



(e) Actual velocity using the original approach

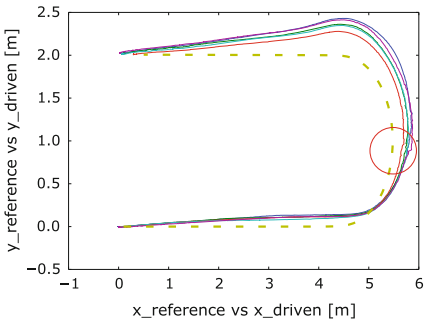


(f) Actual velocity using the proposed approach

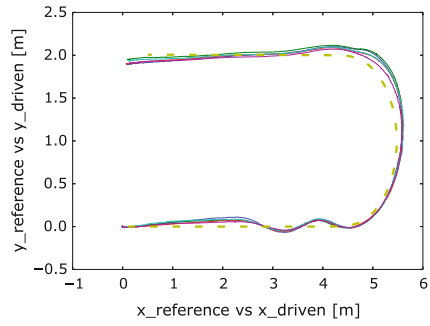
Fig. 4 Five runs of following a U-curve with a 2-axle-steering robot, where $v_n = 0.7$ m/s

Table 2 Proposed versus original algorithm: bi-steerable drive

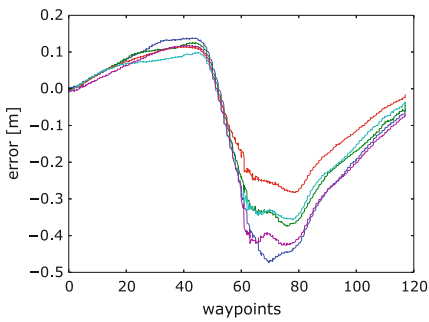
2-steer	e_{max} [cm]	e_{avg} [cm]	σ [cm]	v_{max} [$\frac{m}{s}$]	v_{avg} [$\frac{m}{s}$]
Original	37.62	-5.2	12.34	0.66	0.42
Extended	17.83	-3.74	7.93	0.72	0.37
Δ [%]	52.6	28.13	35.74	8.61	13.68



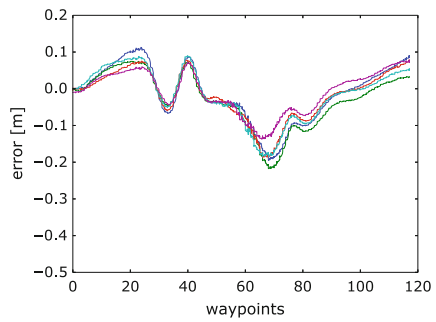
(a) Path following using the original approach



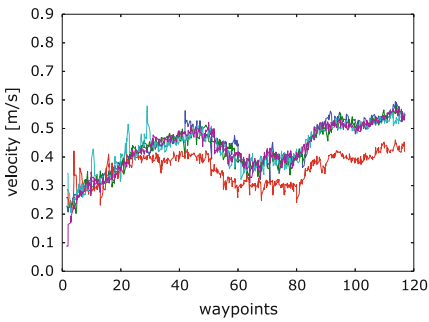
(b) Path following using the proposed approach



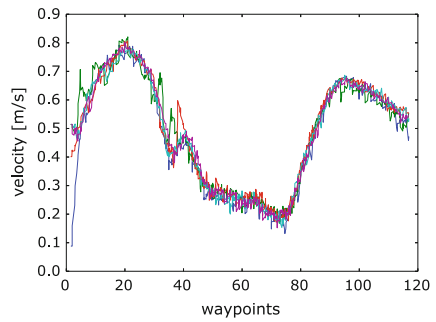
(c) Following error of the original approach



(d) Following error of the proposed approach



(e) Actual velocity using the original approach



(f) Actual velocity using the proposed approach

Fig. 5 Five runs of following a U-curve with an Ackermann-steering robot, where $v_n = 0.6$ m/s

Table 3 Proposed versus original algorithm: Ackermann-steering drive

Ackermann	e_{max} [cm]	e_{avg} [cm]	σ [cm]	v_{max} [$\frac{m}{s}$]	v_{avg} [$\frac{m}{s}$]
original	38.28	-8.42	17.06	0.55	0.4
extended	18.43	-3.618	7.57	0.8	0.41
Δ [%]	51.85	57.03	55.63	30.85	-2.08

smaller maximum error, a 57.03 % smaller average error and a 55.63 % smaller standard deviation. The maximum velocity achieved is 30.85 % higher.

6 Conclusions

We propose a new heuristic longitudinal control for wheeled mobile robots for safe, stable and accurate path following. It is an alternative to complex model-based and predictive controllers which model the dynamics of the robot and the environment. This control is also coupled with the approach proposed in [12], and then utilized for Ackermann- and two-steering drives. Its performance is experimentally evaluated using an omnidirectional robot and a reconfigurable Ackermann- and two-steering robot. The proposed algorithm was also evaluated in an international robotics competition SICK Robot Day 2014, where it coped with a highly dynamic environment and led our team to score 2nd place (detailed in [2]). Recent work includes extending this approach to rough terrain applications.

Acknowledgements Goran Huskić would like to thank the German Academic Exchange Service (DAAD) for his Ph.D. scholarship.

References

1. Aguiar, A.P., Dačić, D.B., Hespanha, J.P., Kokotović, P.: Path-following or reference-tracking? *rn I*, 1 (2004)
2. Buck, S., Hanten, R., Huskić, G., Rauscher, G., Kloss, A., Leininger, J., Ruff, E., Widmaier, F., Zell, A.: Conclusions from an object-delivery robotic competition: Sick robot day 2014. In: 2015 International Conference on Advanced Robotics (ICAR), pp. 137–143. IEEE (2015)
3. Campion, G., Bastin, G., D’Andrea-Novel, B.: Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Trans. Robot. Autom.* **12**(1), 47 (1996)
4. De Luca, A., Oriolo, G., Samson, C.: Feedback control of a nonholonomic car-like robot. In: *Robot motion planning and control*, pp. 171–253. Springer (1998)
5. Egerstedt, M., Hu, X., Stotsky, A.: Control of mobile platforms using a virtual vehicle approach. *IEEE Trans. Autom. Control* **46**(11), 1777–1782 (2001)
6. Fossen, T., Pettersen, K.Y., Galeazzi, R., et al.: Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Trans. Control Syst. Technol.* **23**(2), 820–827 (2015)

7. Kanjanawanishkul, K., Zell, A.: Path following for an omnidirectional mobile robot based on model predictive control. In: IEEE International Conference on Robotics and Automation, 2009. ICRA'09, pp. 3341–3346. IEEE (2009)
8. Li, X., Wang, M., Zell, A.: Dribbling control of omnidirectional soccer robots. In: 2007 IEEE International Conference on Robotics and Automation, pp. 2623–2628. IEEE (2007)
9. Li, Y., He, L., Yang, L.: Path-following control for multi-axle car-like wheeled mobile robot with nonholonomic constraint. In: 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 268–273. IEEE (2013)
10. Lucet, E., Grand, C., Sallé, D., Bidaud, P.: Dynamic sliding mode control of a four-wheel skid-steering vehicle in presence of sliding. In: Proceedings of RoManSy, Tokyo, Japan (2008)
11. Maček, K., Petrović, I., Siegwart, R.: A control method for stable and smooth path following of mobile robots. In: Proceedings of the European Conference on Mobile Robots (2005)
12. Mojaev, A., Zell, A.: Tracking control and adaptive local navigation for nonholonomic mobile robot. In: Intelligent Autonomous Systems (IAS-8), Mar 2004, pp. 521–528. IOS Press, Amsterdam, Netherlands (2004)
13. Oftadeh, R., Ghabcheloo, R., Mattila, J.: Time optimal path following with bounded velocities and accelerations for mobile robots with independently steerable wheels. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 2925–2931. IEEE (2014)
14. Samson, C.: Path following and time-varying feedback stabilization of a wheeled mobile robot (1992)
15. Samson, C.: Control of chained systems application to path following and time-varying point-stabilization of mobile robots. *IEEE Trans. Autom. Control* **40**(1), 64–77 (1995)
16. Soetanto, D., Lapierre, L., Pascoal, A.: Adaptive, non-singular path-following control of dynamic wheeled robots. In: 42nd IEEE Conference on Decision and Control, 2003. Proceedings, vol. 2, pp. 1765–1770. IEEE (2003)
17. Toponogov, V.A.: *Differential Geometry of Curves and Surfaces*. Springer (2006)

Localization Issues for an Autonomous Robot Moving in a Potentially Adverse Environment

Antonio D'Angelo and Dante Degl'Innocenti

Abstract The aim of this paper is to face with the problem of localizing a robot during the navigation in a partially unknown environment. This feature becomes particularly noteworthy especially in the case of a colony of robots, possibly working with humans, inside a scenario where motion issues are crucial. Within this context the focus on self-localization through GPS and INS/SINS integration overtakes merely questions about algorithm efficiency because self-localization is a relevant part of the task. Thus, unlike other approaches, we have focalised on this behavior as an attitude an autonomous system should enhance during the task execution. The tight coupling of GPS and INS sensors is understood as a mechanism which provides the autonomous robot with a refinement of INS use by comparing and/or adjusting the INS performance by exploiting the GPS-INS integration.

Keywords GPS-INS integration · Localization · Neural network · Q-learning

1 Introduction

A problem within the design of the mission control of an autonomous robot is the trajectory adjustment during the execution of the assigned task. Usually this unit executes the task at different levels of abstraction and some of them require specific knowledge about the robot motion which can take place in partially unknown, and even hostile, environments such as military operations or rescue missions.

Thus, the main activity of the robot depends on the specific requirements of the mission at the hand, but its completion relies to the robot ability to find out a safe motion connecting its actual position with the target position. We would notice that in

A. D'Angelo (✉) · D. Degl'Innocenti
Department of Mathematics and Computer Science, Udine University, Udine, Italy
e-mail: antonio.dangelo@uniud.it
URL: <http://www.dimi.uniud.it/~dangelo>

D. Degl'Innocenti
e-mail: deglinnocenti.dante@spes.uniud.it

an hostile environment the working conditions could change very rapidly imposing the autonomous system to adapt accordingly. But, if the dynamics is too fast it can cause undesired behaviors and the robot motion can become definitely unpredictable.

Many countermeasures are taken to avoid such deprecated behaviors. First of all, the use of *global position systems* such as GPS or whatever *fixed star localisation mechanism* capable to validate the actual trajectory covering. Nevertheless, we cannot assume the availability of the GPS signal during all the mission because disturbances can appear as noise at any time or they can results as hostile actions. Hence, the blind motion must be considered as an unavoidable property signalling a possible loss of accuracy in robot motion control.

Usually, when the mission is particularly sensitive, the mission control must cope with this problem in a way the total accumulated noise does not affect the outcome of the mission: remember that *the wider is the absence of GPS signal the wider is the amount of localization uncertainty*. The reference to assigned waypoints is a very useful mechanism to monitor the status of completion of the mission but, especially in hazardous scenarios, the most advantageous approach to estimate robot position in a restricted area, beyond which its localization process is very accurate, is the availability of an *inertial navigation systems*, known as either INS or SINS.¹

Many methods have been proposed in literature to cope with GPS and INS integration, more notably, Kalman filter, either linear or extended, *particle filtering* and *neural networks*: each of them presents advantages and drawbacks. In the follow we shortly discuss such proposals.

1.1 GPS-INS Integration Survey

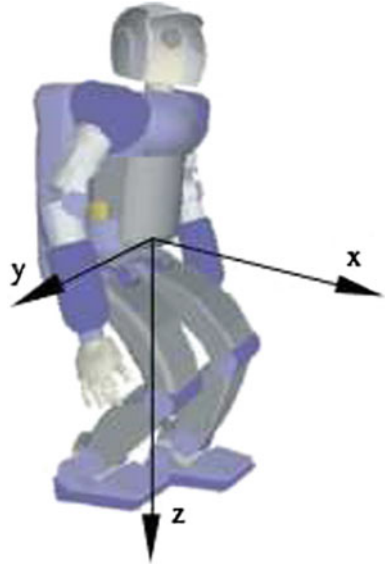
The increasing demand for mobile device localization results in a growing diffusion of a low cost inertial navigation systems. The MEMS² technology seems to meet very well these characteristics but their small dimensions don't scale with their precision due to the large uncertainties of the sensor outputs making them vulnerable to nonlinear error behaviors. For example, Kevin et al. [5] consider the design and implementation of an embedded low cost INS using an inertial measurement unit (IMU), digital compass, GPS, and an embedded computer systems; so, INS provides continuous estimates of a vehicle's position and orientation.

Shin [10] uses a calibration method which does not require IMU aligning to the local level frame and the bias estimation of this method is not affected by the reference gravity error. Most of the positioning error can be removed through accelerometer calibration. Tan [11] considers the design of an accelerometer-based (or gyroscope-free) inertial navigation system that uses only accelerometer measurements to compute the linear and angular motions of a rigid body.

¹Strapdown Inertial Navigation System.

²Micro- electro-mechanical systems.

Fig. 1 Robot body frame



However, to avoid loss of accuracy in robot localization GPS data must be acquired during the mission. So, Sato et al. [9] describe the improvement of the positioning accuracy of a GPS receiver, using software to apply the GPS to compact, hand-held devices. Mao et al. [7] develop UKF³ to model position and velocity estimation for nonlinear filtering to cope with an arbitrary number of satellites; it is especially useful when the visible satellites are frequently variable or less than four (Fig. 1).

Xiaojuan et al. [13] estimate the Hurst parameter of GPS clock difference data based upon the wavelet transform. When $0 < H < 1$, it is taken for a Gaussian zero-mean non-stationary stochastic process which can be considered having the $1/f$ fractal characteristics. Wu et al. [12] review the transformation of coordinates from global positioning system (GPS) signals to 2-D coordinates as a regression problem that derives target coordinates from the inputs of GPS signals directly; a genetic-based solution is proposed and implemented by the techniques of symbolic regression and genetic programming.

Although Kalman filter is used widely in GPS/INS integration, Mao et al. [6] propose an extended Kalman filter approach to estimate the location of a unmanned aerial vehicles (UAV) when its GPS connection is lost, using inter-UAV distance measurements. Mayhew [8] considers several methods for improving the position estimation capabilities of a system by incorporating other sensors and data technologies, including Kalman filtered inertial navigation systems, rule-based and fuzzy-based sensor fusion techniques, and a unique map-matching algorithm. Also, Bin et al. [2] present a GPS/INS integrated navigation system based on the theory of

³Unscented Kalman filter.

multi-sensor data fusion. Error models for the IMU unit are generated and included into the extended Kalman filter for INS.

Azimi-Sadjadi and Krishnaprasa [1] introduce for the first time particle filtering for an exponential family of densities. They prove that under certain conditions the approximated conditional density of the state converges to the true conditional density. In the realistic setting where the conditional density does not lie in an exponential family but stays close to it, they show the estimation error is bounded.

The rest of the paper is organized as follow. In Sect. 2 we shall specify the problem at the hand whereas in Sect. 3 we shall present with some details our proposal to solve the specific problem. At last, in Sect. 4 we shall discuss some experimental results.

2 Problem Position

Let us consider an autonomous robot which can move freely in a partially unknown environment and its mobility is provided by either wheel-driven locomotion or legs or crawlers or whatsoever device especially designed for its mission. Thus, the focus is not on the locomotion itself but on the relation between the robot and the environment and, if they are present, with other robots or even with humans. You can think, for example, to an UGV vehicle which moves inside a hostile environment.

The problem position assumes a mobile robot, which can cooperate with other robots and humans, for a possibly collective task with a specific assigned activity to complete the mission. Here we shall consider the individual task only from the point of view of the path that must be covered towards the target position with the constraints to pass through given way points. They are committed as a part of the task specifications, possibly including to follow assigned paths. We implicitly assume that an environmental map is given at the mission level of abstraction so that some form of self-localization is required.

2.1 Scenario Description

Within this scenario the mission control unit of the robot will provide the connecting path from waypoint to waypoint monitoring its navigation through fixed star localization, such as beacons, GPS constellations, etc. and inertial sensor integration. In the last case, the control system will try to estimate the trajectory by directly accessing the data provided by accelerometers, compass and gyroscopes; however, the raw values need to be processed to correct as best as it is possible, the signal disturbance coming with sensors.

In some sense we could assume that navigation always takes place in a hostile environment⁴ so that the autonomous robot must be able to refine the performance of the task execution while it is taking place. From this point of view, we could introduce a scalar quantity, similar to the DOP⁵ parameter used in satellite navigation, to evaluate the quality of the task execution step by step. However, the term DOP is used to denote the quality of the geometry of the GPS satellite constellation at a particular time: higher DOP results in poor satellite geometry and a potentially less accurate position than a lower DOP.

2.2 Maintaining the Integrity of the Task

During navigation the robot tracks the motion by continuously accessing the raw data coming from inertial sensors by monitoring and comparing them with GPS data when they are available. To overcome such a limitation, the inertial measurement unit (IMU) of the robot needs a dedicated training to continuously adjust the position detection of the robot.

To this aim, let us assume INS reading as a t-upla where each item comes from an independent sensor which usually provides a rotation velocity or an acceleration referred to a given coordinating axis in a cartesian frame of reference. Thus, in most cases we need to monitor and correct up to six independent parameters.

If we focus our attention on one of them, we could observe the course of values $\{x_1, x_2, \dots, x_N\}$ within a short time range of N/f sec. where f is the sampling frequency and N the number of samples. However, because the reading is made on MEMS sensors, we try to obtain a reliable estimate by a weighted average

$$x_e = \frac{w_1 * x_1 + w_2 * x_2 + \dots + w_N * x_N}{N} \tag{1}$$

where the *weight parameters* should be chosen so as to minimize reading errors. In the same range we have the correct values $\{y_1, y_2, \dots, y_N\}$ of the same quantity; its mean value y_c is given by

$$y_c = \frac{y_1 + y_2 + \dots + y_N}{N} \tag{2}$$

and the error ϵ is immediately computed by the difference

$$\epsilon = y_c - x_e \tag{3}$$

⁴Military applications, rescue, hazardous environments such as demining places, pivotal constraints of various nature, and so on.

⁵Dilution of Precision.

which can be easily obtained as mean value of the errors ϵ_k

$$\epsilon = \frac{\epsilon_1 + \epsilon_2 + \dots + \epsilon_N}{N} \quad (4)$$

where each term ϵ_k is given by

$$\epsilon_k = y_k - w_k * x_k \quad (5)$$

In this way the problem of how to obtain reliable INS data moves to the problem of determining satisfactory weight parameters to adjust consistently MEMS readings, by minimizing the *total error* ϵ over N samples. If one minimizes the *local error* ϵ_k for each sample, then he succeeds in minimizing the overall error ϵ . However, it should be noticed that any *inertial system* suffers of the so called *position drift* due to the accumulated errors by the integration over the accelerations and velocities to obtain the current position.

Thus, considering that GPS and INS localizations make use of data which cannot be immediately compared because an integration phase is required on INS data, we choose to focus our attention only on a short range of data at the time. Selecting an appropriate number N of samples we could generate a set of weights $\{w_1, w_2, \dots, w_N\}$ with the property of smoothing the change along the time in a quasi-stationary fashion.

To obtain the required parameters we need a mechanism to generate them: in some sense we would that the IMU can learn from the *GPS observations* how to distill them by *evaluating the position estimate generated by the INS readings*. To this aim, we start by introducing an artificial neuron for each INS sensor which has as many inputs as they are required to simultaneously read sample data in a fixed range, namely, N input weighted connections and an *activation function* $\alpha_{\mathbf{k}}(x)$ defined by

$$\alpha_{\mathbf{k}}(x) = \tanh\left(\frac{\mathbf{k}}{2}x\right) = \sigma(\mathbf{k}x) - \sigma(-\mathbf{k}x) = 2\sigma(\mathbf{k}x) - 1 \quad (6)$$

which represents the *logistic function* where $\sigma(x)$ is the standard sigmoid function. The parameter \mathbf{k} appearing in its definition is directly proportional to its slope at the origin $\mathbf{k} = 2\alpha_{\mathbf{k}}'(0)$. The Fig. 2 shows the shape of the function (6) for different values of the parameter \mathbf{k} accordingly to the following table

$$\begin{aligned} \tanh\left(\frac{x}{2}\right) &\equiv \text{tan-sigmoid}(x) \\ \tanh(20x) &\approx \text{sign}(x) \\ \tanh(x) &\approx \frac{x}{\sqrt{1+x^2}} \end{aligned} \quad (7)$$

but other sigmoid-like functions can be considered. For the aim of this paper we might need to adjust the \mathbf{k} parameter to improve the performance of the neural network to get the *weight parameters*. In fact, the convergence depends on the rule we

Fig. 2 Different sigmoid functions

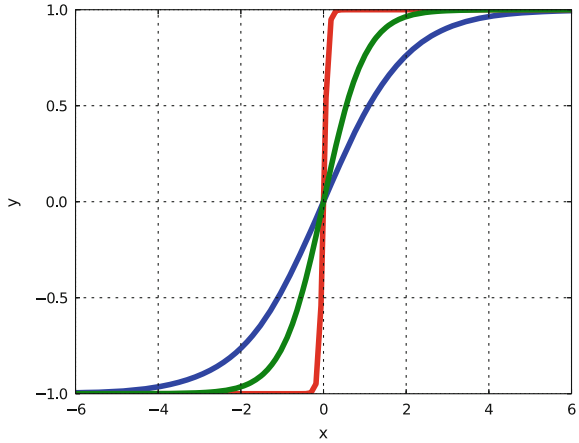
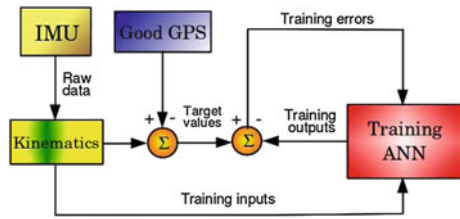


Fig. 3 ANN training architecture



shall adopt to generate the new sequence $\{w_1, w_2, \dots, w_N\}$ of parameters. The choice is really wide and it characterizes the type of neural network to be implemented.

In the rest of the paper we shall consider a *learning rule* borrowed from the *perceptron network* model by modifying the activation rule. In Fig. 3 we have sketched how the perceptron network, labelled ANN, updates the weight parameters to the most reliable estimates from the GPS readings and the current IMU status. For the discussion hereafter the choice is narrowed to find a suitable value for the parameter \mathbf{k} of the sigmoid. Thus, the learning rule becomes

$$\begin{aligned}
 x_e &= w_1 * x_1 + w_2 * x_2 + \dots + w_N * x_N \\
 y_e &= \alpha_{\mathbf{k}}(x_e) \\
 w_k &\leftarrow w_k + Q * (y_c - y_e) * x_k
 \end{aligned}
 \tag{8}$$

where y_c is the correct mean value which must be derived from GPS data, Q is the *learning rate* and its value must be in the range $0 < Q < 1$; its choice characterizes the convergence conditions of the rule.

3 Mission Follower

By considering the noise as an unavoidable component of sensor signal, we need to adjust reading to minimize it. Moreover, noise tends to become the main component just during the time intervals when the GPS signal is dropped out. Thus, the mission planner should be designed to continuously correct its navigation though this adaptive ability could increase the global instability of the robot behavior due to a potential high rate change of direction.

3.1 Motivations

But how could we design the *mission planner* to avoid unwanted asymptotic oscillations by maintaining a smooth trajectory covering? The choice of a model based on neural networks seems very attractive and also suitable to future improvements. Thus, following the idea sketched in the previous section, we shall consider a simplified model of localisation, based on three independent inertial sensors: the planar components a_x and a_y of an accelerometer in addition with the z-axis component ρ_z of a gyroscope, assisted by a GPS receiver.

In the most general case, one needs both a 3-axis accelerometer and a 3-axis gyroscope to have complete access to the inertial forces. However, it can be shown that, with an appropriate rotation and translation of the frame of reference fixed on the IMU device, we can directly access the inertial data as they came from the body-frame of the robot.

As it appears in Fig. 1 this specific frame of reference has x-axis pointing to the forward direction, the z-axis down through the bottom of the robot and the y-axis completes the right-handed orthogonal frame. Within this assumption it can be easily proved the validity of the following identities

$$\begin{aligned}
 a_x &= \frac{dV}{dt} & \rho_x &= 0 \\
 a_y &= V \frac{d\theta}{dt} & \rho_y &= 0 \\
 a_z &= g & \rho_z &= \frac{d\theta}{dt}
 \end{aligned} \tag{9}$$

under the additional assumption that the robot is moving on a plane surface. The scalar quantities V and θ represent, respectively, the forward speed of the robot and its current *bearing*. It is also called the *azimuth*, namely, the horizontal direction on the earth, measured clockwise in degrees around the robot's horizon from north.

Thus, while the GPS receiver is active, we can compare the satellite data with the inertial data and this circumstance is very useful to train the neural network we referred in the previous section to. However, the low quality of MEMS technology has suggested us to consider a set of measures for each sensor device, instead of a single outcome.

Thus, the calibration of the IMU unity is based on the requirement that the errors ϵ_x , ϵ_y , ϵ_z , defined by the following relations

$$\begin{aligned} \epsilon_x &= \frac{dV}{dt} - \overline{a_x} & \overline{a_x} &= \sum_{i=1}^N w_{xi} * a_{xi} \\ \epsilon_y &= V \frac{d\theta}{dt} - \overline{a_y} & \overline{a_y} &= \sum_{i=1}^N w_{yi} * a_{yi} \\ \epsilon_z &= \frac{d\theta}{dt} - \overline{\rho_z} & \overline{\rho_z} &= \sum_{i=1}^N w_{zi} * \rho_{zi} \end{aligned} \quad (10)$$

can be reduced asymptotically to zero. The quantities $\overline{a_x}$, $\overline{a_y}$ and $\overline{\rho_z}$ can be interpreted as either a weighted mean value or a linear combination of input weights. In the latter case, following the idea discussed in Sect. 2 and synthesized by Eq. (8), we shall introduce a neural network with the following learning rule

$$\begin{aligned} w_{xi} &\leftarrow w_{xi} + Q_1 * \left(\frac{dV}{dt} - \alpha_{k_1}(\overline{a_x}) \right) * a_{xi} \\ w_{yi} &\leftarrow w_{yi} + Q_1 * \left(V \frac{d\theta}{dt} - \alpha_{k_1}(\overline{a_y}) \right) * a_{yi} \\ w_{zi} &\leftarrow w_{zi} + Q_2 * \left(\frac{d\theta}{dt} - \alpha_{k_2}(\overline{\rho_z}) \right) * \rho_{zi} \end{aligned} \quad (11)$$

where we have also considered the mean value of the measures provided by GPS. These values are taken on the same number N of measures considered by Eq. (8). The reason is that the mission planner is not required to continuously adjust the tracked trajectory against the followed one though, in principle, it looks the best. On the contrary, the trajectory monitoring should be smooth enough to avoid asymptotic behavior instability, especially when useful adaptations cannot be forecast and only a purely reactive behavior appears admissible. In Fig. 3 we have depicted how this simple schema works: the GPS-INS integration provides the required training of the neural network, which takes action whenever poor or no GPS signal is detected.

4 Experimental Results

The implementation schema presented in Sect. 3 has been motivated by the requirement that the IMU unity could anticipate the long-term behavior of the mission planner, based on past experience and accumulated in the time series that represents the parameters used in the scheme of non-linear regression presented in the paper.

To test the validity of this idea we have considered a simplified experimental setup that highlights the elements of interest only. In this perspective we have assumed the robot is moving on a plane surface or, at least, that it is negligible the variation of height during the motion. To avoid the problem of tuning the IMU output we have

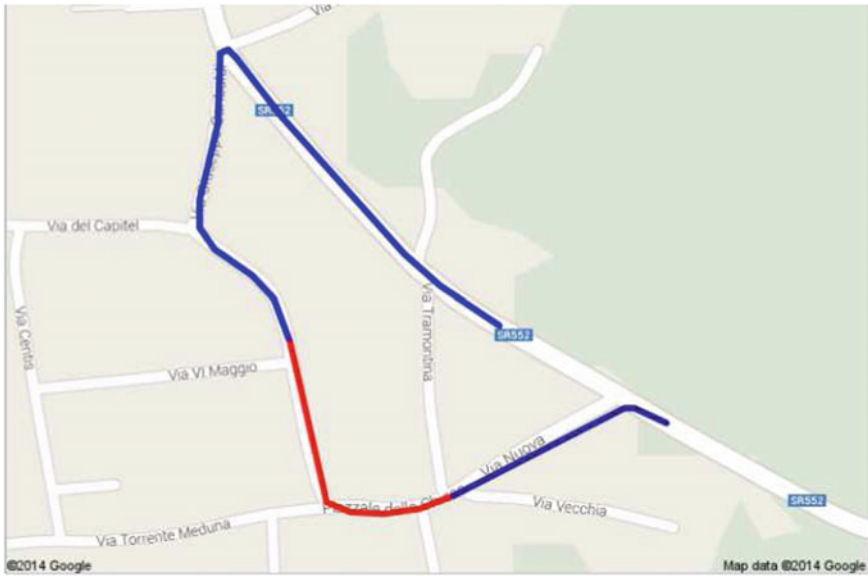


Fig. 4 GPS-INS testbed

centered the mobile terminal on the origin of the body-frame of the robot and to simplify this operation we have chosen to install it on a car. Moreover, the terminal has been connected via USB cable to a portable PC for the required computation.

In the last years many popular smartphones have come to maturation as a mobile terminals equipped with a fully compatible 3-axes accelerometers and gyroscopes assisted by a GPS receiver. More notably the Nexus phones, powered by Google and Samsung, provide a full access to the IMU units along with an android-based development environment to easily implement self-localization applications.

In such a way we have been able to test the performance of the neural network by comparing the self-localization provided by INS sensors with that due to the GPS activity. To this aim we have driven a vehicle, properly equipped with a mobile terminal, inside an area where the GPS quality of satellite signals was good enough to reach the value 5 for the DOP parameter.

It should be remembered that the DOP parameter defines the GPS location accuracy as the radius of 68 % confidence. In other words, if one draws a circle centered at a given location's latitude and longitude, and with a radius equal to the accuracy,⁶ then there is a 68 % probability that the location is inside the circle. This accuracy estimation is only concerned with horizontal accuracy, and does not affect the accuracy of bearing, velocity or altitude if those are included in the location.

In the Fig. 4 one of these experiments is presented: the vehicle travels the colored line where the red part is assumed to have no GPS signal. On the contrary the GPS

⁶5 m in our case.

signal is present along the blue part and, moreover, the longest one is used to train the neural network to actively use the inertial data provided by the IMU unity.

The total traveled distance is 850 m, of which about one-quarter (200 m) is assumed to be not covered by GPS signal. In this setup the mission follower must be trained in the first 490 m and the neural network must be able to determine the weight quantities w_{xk} , w_{yk} and w_{zk} for the next 200 m with an acceptable error. However, a reliable estimate of the measures provided by the IMU unity depends on a number of independent factors, some of which are difficult to determine.

For example, we can assume the error distribution, while inertial data are tracked, is a Poisson process. Thus, the number of measures for each inertial sensor axis can be taken very small. If data are sampling at a frequency of 5 Hz, a number of 5 measures for inertial axis means that the elapsing time ranges over 1 s which can be considered a good setting for a robot which moves at a moderate speed.

In a first set of experiments we have assumed that the sigmoids for the accelerometer and the gyroscope were shaped, respectively, by the factors $\mathbf{k}_1 = 1$ and $\mathbf{k}_2 = 13$. On the contrary, the learning rates for both sensors were taken with the same value, namely, $Q_1 = 0.3$ and $Q_2 = 0.3$.

However, the learning rates Q_1 and Q_2 are critical parameters in what they dramatically affect the convergence conditions expressed by Eq. (11). Also the factors \mathbf{k}_1 and \mathbf{k}_2 are equally crucial but their choice can mitigate the rapid changes of the tracked trajectory by altering the weight average of the INS outcomes.

In Fig. 5 we have reported different estimations of the trajectory traveled by the vehicle under different assignment of the weight parameters provided by different computing cycles as a result of different training sets. In fact, the implementation repeatedly tries to recompute the weight parameters because, as it has been previously pointed out, they must be freshly provided on demand when either a poor DOP or no GPS signal are detected.

However, by comparing Figs. 4 and 5, one can observe that the different colored lines fail to estimate the trajectory has it has been detected by GPS data. The main reason is due to the choice of the shape factors and the learning rates. In fact, in a second set of experiments we have kept $\mathbf{k}_1 = 1$ and $\mathbf{k}_2 = 13$ again but the learning rates were, respectively, $Q_1 = 0.4$ and $Q_2 = 0.5$. After a number of attempts, as a result of different training sets, we have obtained the tracked trajectory depicted in Fig. 6. In this case a different sequence of weight parameters has been generated resulting in a more realistic traveled trajectory and it can be considered a satisfactory result in all the applications where only an acceptable tracking property is required.

This schema of position prediction extends a similar method which has been discussed by Degl’Innocenti [4]. In his master thesis he argues that the restore of the travelled path⁷ can be obtained only looking at the local data coming from the inertial sensors. He tries to correct inertial drifting by tuning kinematic constants and adjusting INS data using specialized non linear functions, such as sigmoid, atan, and so on. He also reviews and compares the experimental results evaluating the geometric properties of the estimated paths with the respect to the travelled ones.

⁷Lost due to lack of GPS signal.

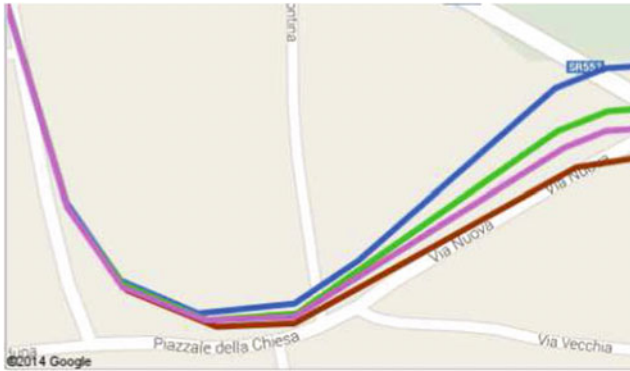


Fig. 5 GPS-INS integration

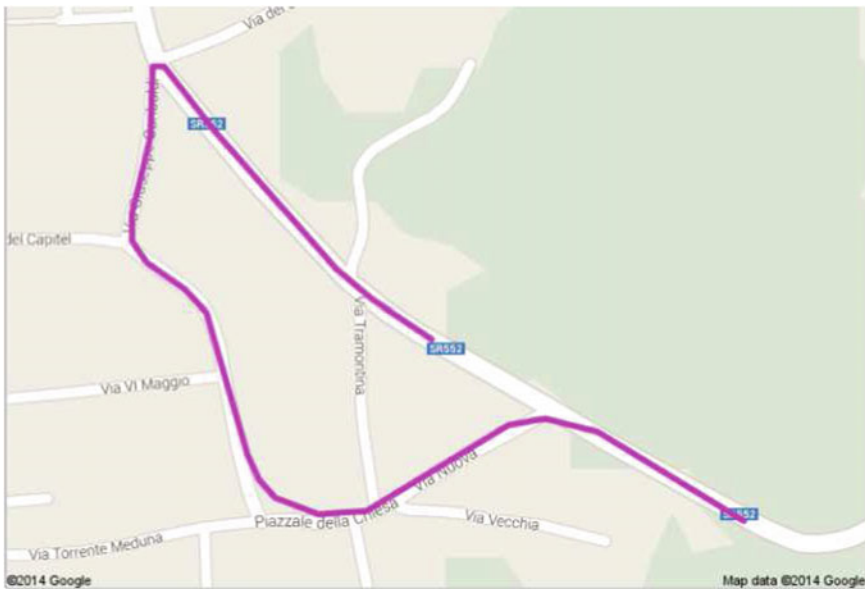


Fig. 6 Complete GPS-INS integration

Following the similar idea but from another point of view, Chiang et al. [3] suggest an hybrid scheme to overcome the limitations of Kalman filter and INS/GPS integration, by introducing an Artificial Neural Network coupled with a Kalman filter. Despite the improvement the authors argue the accuracy requirements of general mobile mapping applications cannot be achieved easily. On the contrary in our approach such requirements are not really necessary so the last presented results of Fig. 6 are largely acceptable with the respect of the problem position.

5 Conclusions

The position estimation proposed in the paper is an attempt to integrate GPS and INS data with a simple neural network by mitigating the computational complexity due to the Extended Kalman Filter and the particle filtering approaches. Some observed oscillating outcomes of the w_j parameters provided by the neural network during repeated experiments suggest to refine the adaptive property of the network by improving the convergence conditions (8).

However, because the approach is based on the idea that a lack of GPS signal requires to estimate the travelling position until fresh GPS signal is available, the only real limitation is the updating rate of the w_j parameters. In some sense, we can consider this approach countercyclical in what it avoids faster response rates when an increment of environmental dynamics is detected.

Thus, the mission unit makes a continuous use of GPS data, when they are available, to train the neural network and the proposed weight parameters are the last successfully computed. This operating condition is really a limitation when the GPS signal is frequently dropped out. We assume that this scenario would happen very rarely. On the contrary, a training support must be considered.

Besides this drawback, the proposed approach appears an intelligent hybrid scheme where a SINS device is directly integrated with a GPS receiver through a neural network whose training is achieved while the robot is normally operating in the environment. However, an initial tuning of the neural network is required. We cannot exclude that a simple extension of its operation is based on an initializing parameter table to be loaded on a case by case basis.

Acknowledgements This work was partially supported by a grant of the University of Padua's Special Project on *Mobility, Perception, and Coordination for a Team of Autonomous Robots* and also by a special grant of *ABTrack Limited Corporation*.

References

1. Azimi-Sadjadi, B., Krishnaprasad, P.S.: Approximate nonlinear filtering and its application in navigation. *Automatica* **41**(6), 945–956 (2005)
2. Bin, W., Jian, W., Jianping, W., Baigen, C.: Study on adaptive gps/ins integrated navigation system. In: *IEEE Intelligent Transportation Systems*. vol. 2, pp. 1016–1021. Shanghai, China (2003)
3. Chiang, K.W., Chang, H.W., Li, C.Y., Huang, Y.W.: An artificial neural network embedded position and orientation determination algorithm for low cost mems ins/gps integrated sensors. *Sensors* **9**, 2586–2610 (2009)
4. Degl'Innocenti, D.: Sviluppo di una Soluzione integrata GPS/INS per Dispositivi Embedded e Android. Master's thesis, Laurea Specialistica in Informatica, Università di Udine (2013)
5. Kevin, J., Michael, C., Nechyba, S., Antonio, A.: Embedded low cost inertial navigation system. In: *Florida Conference on Recent Advances in Robotics*. FAU, Dania Beach (FL) (2003)
6. Mao, G., Drake, S., Anderson, B.: Design of an extended kalman filter for uav localization. In: *IEEE Information, Decision and Control*, pp. 224–229. Adelaide, Qld (2007)

7. Mao, X., Masaki, W., Hideki, H.: Nonlinear gps models for position estimate using low-cost gps receiver. In: *IEEE Intelligent Transportation Systems*, vol. 1, pp. 637–642 (2003)
8. Mayhew, D.M.: *Multi-rate Sensor Fusion for GPS Navigation using Kalman Filtering*. Master's thesis, Virginia Polytechnic Institute and State University, Electrical Engineering Department (1999)
9. Sato, G., Asai, T., Sakamoto, T., Hase, T.: Improvement of the positioning of a software-based gps receiver using a 32-bit embedded microprocessor. *IEEE Trans. Consum. Electr.* **46**(3), 521–530 (2000)
10. Shin, E.H.: *Accuracy Improvement of Low Cost INS/GPS for land application*. Master's thesis, University of Calgary, Geomatic Engineering Department (2001)
11. Tan, C.W.: Design of accelerometer-based inertial navigation systems. *IEEE Trans. Instrum. Meas.* **54**(6), 2520–2530 (2005)
12. Wu, C.H., Chou, H.J., Su, W.H.: A genetic approach for coordinate transformation test of gps positioning. *IEEE Geosci. Remote Sens. Lett.* **4**(2), 297–301 (2007)
13. Xiaojuan, O., Wei, Z., Jianguo, Y.: Study on gps common-view observation data with multi-scale kalman filter based on correlation structure of the discrete wavelet coefficients. In: *IEEE International Frequency Control Symposium and Exposition*, pp. 685–690. Vancouver, BC (2005)

Vector-AMCL: Vector Based Adaptive Monte Carlo Localization for Indoor Maps

Richard Hanten, Sebastian Buck, Sebastian Otte and Andreas Zell

Abstract For navigation of mobile robots in real-world scenarios, accurate and robust localization is a fundamental requirement. In this work we present an efficient localization approach based on adaptive Monte Carlo Localization (AMCL) for large-scale indoor navigation, using vector-based CAD floor plans. The approach is able to use the line segment data of these plans directly. In order to minimize the computational effort, a visibility lookup table is generated, reducing the amount of line segments to process for pose estimation. In addition, we show that the proposed approach performs well in cluttered as well as uncluttered environments. It is compared with grid map-based AMCL and is able to improve its results in terms of memory usage and accuracy.

Keywords Indoor localization · Floor plans · Line segments · CAD

1 Introduction

Self-localization is one the most fundamental tasks in mobile robotics, especially in regard to autonomous operation in various environments. The choice of a suitable localization algorithm strongly depends on the field application and the related requirements.

Large-scale indoor navigation in public buildings, for instance, comes with several difficulties. Since mobile robots are limited to certain computational resources, a

R. Hanten (✉) · S. Buck (✉) · S. Otte · A. Zell
University of Tübingen, Sand 1, 72076 Tübingen, Germany
e-mail: richard.hanten@uni-tuebingen.de

S. Buck
e-mail: sebastian.buck@uni-tuebingen.de

S. Otte
e-mail: sebastian.otte@uni-tuebingen.de

A. Zell
e-mail: andreas.zell@uni-tuebingen.de

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_29

compact map and efficient sensor data incorporation is desirable. Furthermore, such buildings can be considered highly dynamic environments, making it very hard to employ any kind of mapping algorithm.

A suitable solution for the problem is to use a localization algorithm directly working on CAD floor plan data. Nevertheless, the solution should meet certain general requirements: The support of arbitrary floor plans without manual pre-processing is preferable. Besides, a line segment-based localization approach should be able to efficiently incorporate sensor data. More importantly, the algorithm should yield a robust pose estimate. For mobile robots, with limited resources, a compact map representation is of advantage. It is preferable to preserve the continuous CAD information, which is potentially more accurate than discretised representations.

Several localization algorithms using line segment based maps have been introduced in the past. *Simultaneous Localization and Mapping* (SLAM) approaches have been presented by Garulli et al. [1], Sohn et al. [2], Elsenberg et al. [3] and Mazuran et al. [4]. As pointed out before, SLAM algorithms may not be preferable.

The proposal by Cox [5] uses manually generated line maps to improve a rough pose estimate given by robot odometry in an easy office environment. Furthermore, Sohn et al. [6] present a localization algorithm based on iterative vector matching. The approach uses CAD maps with directed line segments, encoding in- and out-sides of walls. Both algorithms do not meet the introduced requirements, due to the necessity of manual pre-processing and the neglect of uncertainty estimations.

Robust localization in such environments can be achieved using a well-known technique called Bayesian filters [7]. Kalman filters, for instance, represent robot pose estimates by Gaussian distributions, which are by definition uni-modal. Particle filters, on the other hand, are able to represent multi-modal distributions, allowing several pose hypotheses at once. That makes them more suitable for the global localization problem, compared to Kalman filters.

He et al. [8] present a multi-hypothesis localization approach. This work gains higher robustness in highly dynamic surroundings through the consideration of uncertainty. A vector map is realized through an interconnected graph of lines. Luo et al. [9] present a single hypothesis pose estimation approach using features extracted of a line segment-based map and the according laser range readings. Both algorithms, however, require costly pre-processing and do not fully support a global localization scheme.

Another particle filter-based approach is presented by Ito et al. [10]. Floor plans are converted to grid maps and Wi-Fi signal strengths are used to improve the pose estimate. However, it is likely that large-scale environments are not fully covered with Wifi hot spots. Besides that, floor plans have to be discretised, resulting in a less compact and more inaccurate map representation.

The approach we present in this work satisfies all the introduced requirements.

- (I) We contribute an efficient vector-based algorithm for large-scale indoor environments.
- (II) It is based on the Adaptive Monte Carlo localization approach introduced in Sect. 2.1, benefiting computational effort and robustness.

- (III) Efficient line segment data incorporation is realized through modified sensor models presented in Sect. 2.3.
- (IV) We introduce a line visibility lookup table for fast, global map access, described in Sect. 2.4. The index reduces the number of map line segments when processing sensor input.

Finally, in Sect. 3, V-AMCL is evaluated in comparison with the original AMCL implementation.

2 Vector-Based Monte Carlo Localization

2.1 Adaptive Monte Carlo Localization (AMCL)

Monte Carlo Localization (MCL) [7, 11] is a widely used technique for estimating the pose of a mobile robot in an already known environment. The location is determined by the posterior belief of the state x_t at time step t , whereas a finite set of particles S_t represents the related posterior distribution.

$$\text{bel}(x_t) = p(x_t | u_{1:t}, z_{1:t}) \quad (1)$$

$$S_t = \{x_t^{[n]} | 1 \leq n \leq N\} \quad (2)$$

The number of particles is denoted by N , also called the sample size. Control inputs, which were used for particle propagation with a motion model until time step t , are denoted $u_{1:t}$. Accordingly, the sensor measurements $z_{1:t}$ were used to weight the particles with an appropriate model. This model yields a measurement probability $p(z_t | x_t^{[n]})$ for each particle $x_t^{[n]}$. Its weight $w_t^{[n]}$ is updated with

$$w_t^{[n]} = p(z_t | x_t^{[n]})w_{t-1}^{[n]}. \quad (3)$$

The weight is basically the integration of measurement probabilities over time. In distinct time intervals, the MCL algorithm conducts a resampling step drawing particles from the current set with replacement. The drawn particles are added to a new set, which is then used to progress. The higher $w_t^{[n]}$, the more likely $x_t^{[n]}$ is introduced into the new set.

While the original MCL approach relies on a fixed sample size N , an adaptive resampling scheme was introduced by Fox [12]. This scheme selects as many samples as necessary to minimize the error between real and sampled posterior with help of the *Kullback-Leibler Distance (KLD)*.

The adaptive sample size is a real benefit for our line-based sensor model, since the application of our line segment-based model requires a lot of geometric computations.

2.2 Motion Models

In this work, we only consider the differential drive *odometry motion model* [7, 11], although our implementation would allow the usage of further models. In Fact, all experiments presented in Sect. 3 used a robot with a differential drive.

2.3 Observation Models

The observation models employed in this work are known by the name *beam* and *likelihood field model* [7]. For a measurement z_t consisting of K 2D laser range readings z_t^k and a particle $x_t^{[n]}$ the probability

$$p(z_t | x_t^{[n]}, m) = \prod_{k=1}^K p(z_t^k | x_t^{[n]}, m) \quad (4)$$

is calculated. This is done for the whole particle set S_t and applies to both, grid-based and line segment-based maps m . For each particle a set of hypothetical range readings z_k^{f*} is generated which is then compared with the corresponding laser range readings of z_k^f . The key idea of the beam model is to use a mixture probability distribution as shown in Fig. 1a. It is defined over the interval from 0 m to z_{max} , the maximum distance the used laser scanner can measure. It consists of probability distributions for random, short and maximum readings as well as a Gaussian distribution for a hit. Hit denotes the case, that there is only a very small difference between the real range reading z_t^k and the generated range reading z_t^{k*} . In addition to that, z_t^{k*} also determines the position of the maximum of the mixture distribution.

In contrast to the beam model, the likelihood field model uses Gaussian distributions around obstacles to model a hit by a laser ray with uncertainty. Figure 1b

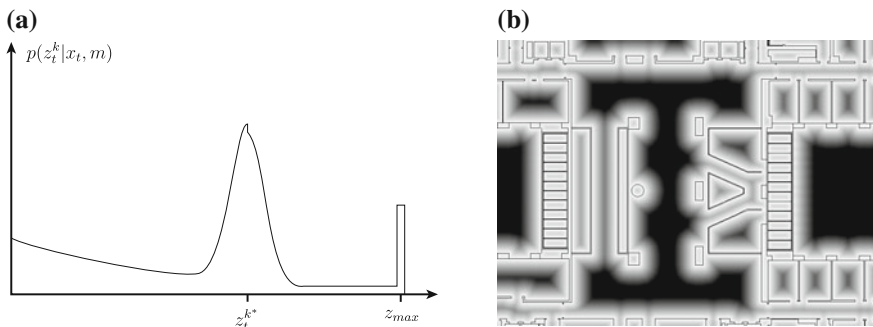


Fig. 1 **a** A mixture distribution employed by the beam model with the highest peak around z_t^{k*} . **b** A likelihood field around obstacles, in this case walls

illustrates an example for that. Range measurements z_t^k of z_t are used to calculate ray end points for each particle. The distance to the nearest obstacle is calculated and then used to draw a probability from the mentioned distribution.

2.4 Vector Maps

The concepts of the sensor models are directly applicable to a line segment-based map representation. Solely the map data interpretation differs. Working with vectors has the benefit of avoiding a discretised obstacle representation, but also leads to higher arithmetic expense. That also applies to the goal to work with arbitrary CAD based floor plans, especially in large-scale environments. A major issue in this work, was to find a solution for the computation of the line segment information efficiently.

A large architectural map $\mathcal{M} = \{s_1, \dots, s_M\}$ can contain many thousands of line segments s_i . Without further pre-processing, we would have to intersect each of those lines with every hypothetical laser beam, which is clearly not feasible. We therefore construct a spatial index \mathcal{I} to speed up the laser scan generation by performing a filtering step in an offline phase.

\mathcal{I} is implemented as a coarsely resolved, three dimensional lookup table over the x - y position and the yaw-orientation θ . Every cell c of the lookup table is characterized by its bounding box $bb(c)$, an angular range $\Delta\theta(c)$ and a set $\mathcal{V}(c) = \{s_1, \dots, s_c\}$ of all the line segments that are visible to any point in c . To generate a virtual laser beam at pose $p = (x, y, \theta)$, we look up the cell c_p that p is located in and then perform intersection tests using $\mathcal{V}(c_p)$ instead of \mathcal{M} .

The calculation of a cell c of \mathcal{I} is split into multiple phases. We construct c by keeping track of a set of possibly visible lines c' , starting with all segments

$$c' \leftarrow \{s \mid s \in \mathcal{M}\}, \quad (5)$$

and then consecutively filtering segments that are not visible. Let bb be the axis-aligned bounding box of c and r_{max} be the maximum range of the laser scanner in use. As a first step, we discard segments that are too far away and only keep the segments

$$c' \leftarrow \{s \mid s \in c', \text{mindist}(s, bb) \leq r_{max}\}, \quad (6)$$

for which the closest distance to the four edges of bb is smaller than the laser scanners range. This alone results in a massive gain in performance for maps that are far larger than r_{max} .

In a second step, we remove hidden line segments, by iterating s' and removing those that are completely hidden to any point in bb . We calculate a *shadow* polygon $\mathcal{P}_s(s_i)$ for each segment s_i , as visualized in Fig. 2b. Every segment that completely lies inside the shadow of another will be removed from c'

$$c' \leftarrow c' \setminus \{s \mid s \in c', \text{shadowed}(s)\}. \quad (7)$$

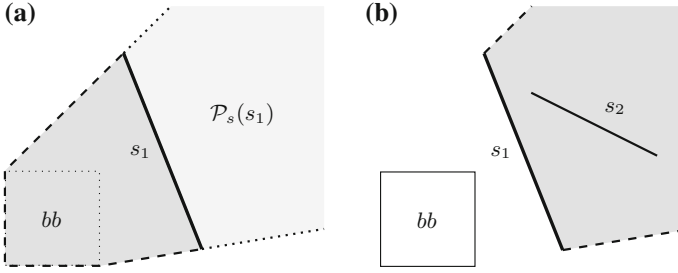


Fig. 2 **a** Calculation of the Shadow $S(s_1)$ by constructing the convex hull of bb and s_1 and extending the newly added segments. **b** Segment s_2 is invisible to bounding box bb because it is in the shadow of s_1 . Therefore, it is removed from the index \mathcal{I}

The shadow $\mathcal{P}_s(s_i)$ of segment s_i is calculated by calculating the convex hull of bb and the endpoints of s_i and then extending the newly added line segments into infinite lines, as demonstrated in Fig. 2a.

As a final step, we approximately calculate the set of possibly visible lines. Hence, we sample the cell on a grid with a resolution of $\delta = 0.5$ m and perform ray casting at each sample point to detect visible lines by sweeping a ray $r = (\cos(\theta), \sin(\theta))$ for $\theta \in \Delta\theta(c)$ with an angular resolution $\alpha = 1^\circ$. Every segment that is hit by any one of these rays is kept

$$c' \leftarrow \{s \mid s \in c', \text{hit}(s)\}. \quad (8)$$

This step is only an approximation, since small segments at large distance and structures far smaller than δ can be missed, due to the grid sampling scheme. Nevertheless, this approximation reduces the amount of segments per cell.

3 Experiments

For a comparative performance evaluation of V-AMCL, we compare it to AMCL using occupancy grid maps generated from discretised floor plans. We conducted several experiments on simulated as well as real world data. A simulated data set was generated by ourselves, directly using floor plan information. Real world experiments were conducted on a data set provided by the Rawseeds Project [13]. The project also specifies mandatory and optional benchmarking metrics [14], where we used the ATE and the RPE, which are explained in Sect. 3.1.

We evaluated all experiments with the continuous and discretised beam and likelihood field model implementations. The different models were introduced in Sect. 2.3. For best performances of the tested algorithms, we chose to optimize the sensor model parameters as good as possible. This is a necessary step since behaviour changed due to the usage of a continuous data representation. The robot platform in all experiments used a differential drive, we therefore employed the previously men-

tioned differential drive odometry motion model for particle propagation. Parameters for that model were chosen once for real world and for simulation.

We chose to examine the influence of adaptive sample sizes and ran each experiment several times, calculating a mean outcome. Since V-AMCL and AMCL use a randomized resampling scheme and Gaussian distributed noise for the odometry model, this was a reasonable step to take. The CAD floor plans had to be rasterised for AMCL, where resolutions of 0.05, 0.10, and 0.2 m were chosen. For V-AMCL we considered visibility lookup table resolutions of 1.0, 2.0, and 4.0 m. Furthermore, a range of 22.5° was used for angular binning in the lookup table.

In Tables 1, 2, 3, 4, 5, 6, 7, and 8 best results are highlighted.

3.1 Performance Measures

3.1.1 Absolute Trajectory Error (ATE)

The ATE is a mandatory benchmarking metric [15] defined by the Rawseeds Project. It is the absolute error between ground truth and computed trajectory, expressing quality of overall localization. Only the 2D position of the robot is taken into account, while the robot orientation is dismissed. At time step t_j , the *translation error* d_j is calculated between the computed pose x_j and the ground truth pose x_j^{GT} with

$$d_j = ||\text{trans}(x_j) - \text{trans}(x_j^{GT})||. \quad (9)$$

The translational component of the poses is denoted by $\text{trans}(x_j)$ and $\text{trans}(x_j^{GT})$. In the end, the mean translative error \bar{d}_j and the standard deviation σ_d are calculated.

3.1.2 Relative Pose Error (RPE)

The RPE is a recommended benchmarking metric [16], which measures the relative localization accuracy. It consists of the *Translational Relative Pose Error* (T-RPE) and the *Rotational Relative Pose Error* (R-RPE). Calculation takes place for a pair poses (x_i, x_j) of consecutive time steps t_i and t_j . The relative pose-to-pose transformation is denoted by

$$x_{ij} = x_j \ominus x_i. \quad (10)$$

Here, \ominus is the inverse of the standard motion composition operator, transforming the pose x_i onto x_j . The error to the expected ground truth transformation x_{ij}^{GT} is given by

$$d_{ij} = x_{ij} \ominus x_{ij}^{GT}, \quad (11)$$

the relative displacement. The T-RPE is calculated by

$$\text{T-RPE} = \frac{1}{N} \sum_{ij} (\text{trans}(d_{ij}))^2, \quad (12)$$

where N denotes the number of transformation pairs and trans refers to the first two components of d_{ij} . Respectively, the R-RPE is determined by

$$\text{R-RPE} = \frac{1}{N} \sum_{ij} (\text{rot}(d_{ij}))^2, \quad (13)$$

where rot refers to the third component of d_{ij} .

3.2 Evaluation

3.2.1 Simulated Data

The simulated data set was generated with Gazebo, a well-known 3D simulator for robotics and is based on the floor plan of our research chair at the University of Tuebingen. It contains a clutter free environment and therefore provides optimal conditions for localization using line segments. Figure 3 shows the 189.99 m long trajectory for which we recorded simulated odometry, sensor measurements, and ground truth positions. Gaussian distributed noise was added to both simulated sensors and odometry. Table 1 shows the results for V-AMCL and AMCL using the likelihood field model with a fixed sample size of $N = 500$. V-AMCL achieved a smaller localization error compared to AMCL in terms of the ATE. R-RPE and T-RPE are small in both cases. This indicates that the odometry motion model was configured the right way and the sensor models worked well. Highly weighted hypotheses were kept by the resampling scheme.

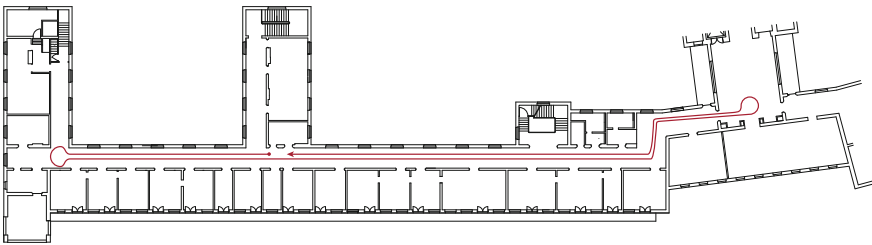


Fig. 3 The floor plan of our research chair with an extent of 30–115 m

Table 1 Overall localization error on virtual data using the likelihood field model. Sample sizes of AMCL and V-AMCL are fixed to $N = 500$

	Res. (m)	\bar{d}_j (m)	σ_{d_j} (m)	T -RPE (m)	R -RPE (deg)
AMCL	0.05	0.1024	0.0668	0.0015	0.1227
	0.10	0.1045	0.0629	0.0015	0.1222
	0.20	0.1091	0.0664	0.0015	0.1228
V-AMCL	1.0	0.0674	0.0527	0.0015	0.1221
	2.0	0.0656	0.0464	0.0016	0.1224
	4.0	0.0717	0.0541	0.0016	0.1223

Table 2 Overall localization error on virtual data using the beam model. Sample sizes of AMCL and V-AMCL are fixed to $N = 500$

	Res. (m)	\bar{d}_j (m)	σ_{d_j} (m)	T -RPE (m)	R -RPE (deg)
AMCL	0.05	0.0447	0.0389	0.0015	0.1237
	0.10	0.0484	0.0357	0.0015	0.1237
	0.20	0.0777	0.0581	0.0016	0.1230
V-AMCL	1.0	0.0658	0.0547	0.0017	0.1221
	2.0	0.0670	0.0552	0.0017	0.1220
	4.0	0.0575	0.0461	0.0016	0.1225

Table 3 Overall localization error on virtual data using the likelihood field model. Sample sizes of AMCL and V-AMCL are dynamic with $N \in [500, 1000]$

	Res. (m)	\bar{d}_j (m)	σ_{d_j} (m)	T -RPE (m)	R -RPE (deg)
AMCL	0.05	0.0901	0.0613	0.0015	0.1230
	0.10	0.0982	0.0644	0.0015	0.1226
	0.20	0.1079	0.0661	0.0015	0.1228
V-AMCL	1.0	0.0640	0.0485	0.0016	0.1223
	2.0	0.0680	0.0496	0.0016	0.1218
	4.0	0.0735	0.0534	0.0016	0.1211

Table 2 illustrates the results for the utilized beam model implementations. Results differ significantly from the previous experimental setup. While V-AMCL behaved almost the same, it is outperformed by AMCL regarding the ATE. R-RPE and T-RPE are comparable. This implies, that either the discretised map representation works better with the beam model or that the parameters for the line segment-based beam model could have been optimized more aggressively.

Tables 3 and 4 show the results of AMCL and V-AMCL on simulated data using an adaptive sample size of $N \in [500, 1000]$. Adaptive sample sizes do not influence the localization accuracies for the synthetic experiments.

Table 4 Overall localization error on virtual data using the beam model. Sample sizes of AMCL and V-AMCL are dynamic with $N \in [500, 1000]$

	Res. (m)	\bar{d}_j (m)	σ_{d_j} (m)	$T\text{-RPE}$ (m)	$R\text{-RPE}$ (deg)
AMCL	0.05	0.0451	0.0389	0.0015	0.1237
	0.10	0.0503	0.0374	0.0015	0.1236
	0.20	0.0734	0.0526	0.0016	0.1232
V-AMCL	1.0	0.0616	0.0505	0.0017	0.1217
	2.0	0.0626	0.0513	0.0017	0.1211
	4.0	0.0610	0.0476	0.0017	0.1228

Table 5 Overall localization error on real world data using the likelihood field model. Sample sizes of AMCL and V-AMCL are fixed to $N = 500$

	Res. (m)	\bar{d}_j (m)	σ_{d_j} (m)	$T\text{-RPE}$ (m)	$R\text{-RPE}$ (deg)
AMCL	0.05	0.2131	0.1406	0.0670	0.7721
	0.10	0.2164	0.1289	0.0744	0.7788
	0.20	0.2229	0.1295	0.0867	0.7840
V-AMCL	1.0	0.1869	0.0888	0.0684	0.7702
	2.0	0.1894	0.0919	0.0705	0.7637
	4.0	0.2054	0.1036	0.0674	0.7794

3.2.2 Real World Data

The Rawseeds Project provides several indoor and outdoor data sets containing sensor data and ground truth robot trajectories.

For our experiments we chose one of five available data sets, identified by *Bicocca1_2009-02-25b*.¹ Figure 4 shows the floor plan as well as the ground truth trajectory. Contrary to the simulation, the environment is massively cluttered and provides more difficult conditions for V-AMCL and AMCL, which use floor plans only.

Tables 5 and 6 illustrate the localization accuracies for fixed sample sizes. As to be expected, ATE, R-RPE and T-RPE turn out much higher compared to the evaluation on simulated data. AMCL and V-AMCL are both confronted with a challenging environment containing a lot of book shelves, objects and structural changes to the building compared to the floor plan. Table 5 shows that V-AMCL outperforms AMCL using the likelihood field approach. The ATE is lower, especially concerning the standard deviation. Applying line segment based sensor models seems to result in better particle weights compared to the grid map based ones. For both algorithms T-RPE and R-RPE are alike, but reveal difficulties caused by the cluttered environments.

¹A demonstrative video of V-AMCL performing on the data set can be found at <https://youtu.be/YwekqsCRQFE>.

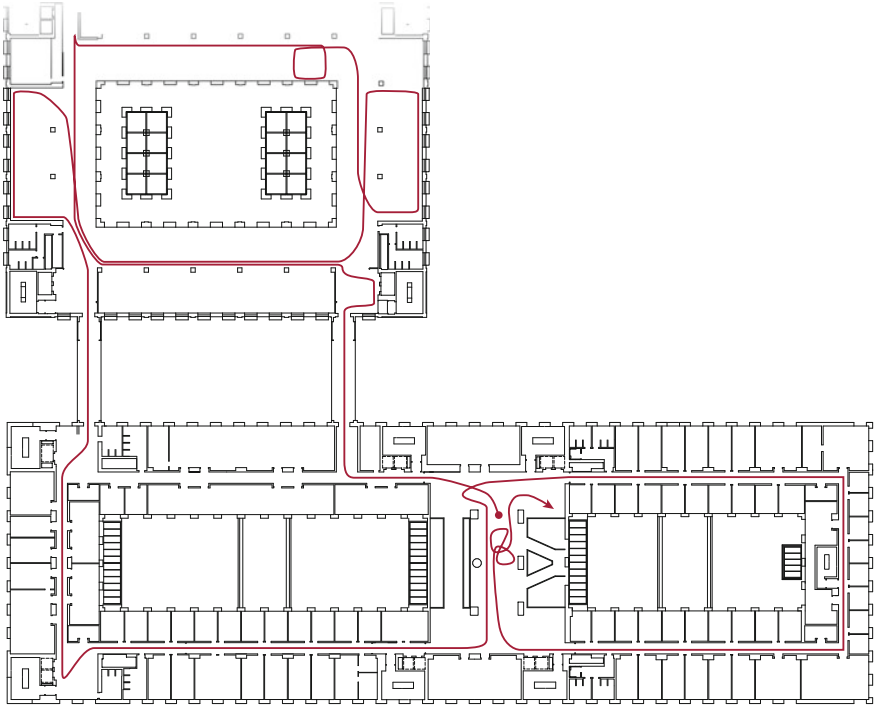


Fig. 4 Part of the floor plan of the Bicocca University building, used for gathering the data. While the shown ground truth trajectory is about 740 m long, the extent of the complete floor plan is about 140 m × 240 m

Table 6 Overall localization error on real world data using the beam model. Sample sizes of AMCL and V-AMCL are fixed to $N = 500$

	Res. (m)	\bar{d}_j (m)	σ_{d_j} (m)	$T-RPE$ (m)	$R-RPE$ (deg)
AMCL	0.05	0.2019	0.1248	0.0755	0.8134
	0.10	0.2035	0.1168	0.0880	0.8104
	0.20	0.2197	0.1219	0.1070	0.9099
V-AMCL	1.0	0.1956	0.1028	0.0701	0.7991
	2.0	0.1919	0.1019	0.0709	0.7083
	4.0	0.1880	0.0945	0.0719	0.7932

Table 6 contains results for AMCL and V-AMCL employing beam models. Contrary to the evaluation on simulated data, the performance of AMCL does not improve significantly, although results are better than achieved with the likelihood field model. V-AMCL performs better than AMCL, though using the beam model did not improve the results compared to the likelihood field model. T-RPE and R-RPE are nearly the same for both algorithms and their model implementations. Tables 7

Table 7 Overall localization error on real world data using the likelihood field model. Sample sizes of AMCL and V-AMCL are dynamic with $N \in [500, 1000]$

	Res. (m)	\bar{d}_j (m)	σ_{d_j} (m)	$T-RPE$ (m)	$R-RPE$ (deg)
AMCL	0.05	0.2172	0.1434	0.0678	0.7980
	0.10	0.2236	0.1430	0.0746	0.8129
	0.20	0.2392	0.1429	0.0902	0.8254
V-AMCL	1.0	0.1886	0.0941	0.0680	0.7739
	2.0	0.1885	0.0892	0.0682	0.7657
	4.0	0.2044	0.1031	0.0671	0.7756

Table 8 Overall localization error on real world data using the beam model. Sample sizes of AMCL and V-AMCL are dynamic with $N \in [500, 1000]$

	Res. (m)	\bar{d}_j (m)	σ_{d_j} (m)	$T-RPE$ (m)	$R-RPE$ (deg)
AMCL	0.05	0.2017	0.1248	0.0770	0.8413
	0.10	0.2040	0.1181	0.0850	0.8324
	0.20	0.2262	0.1304	0.1182	0.9686
V-AMCL	1.0	0.1922	0.1017	0.0706	0.7890
	2.0	0.1937	0.1040	0.0674	0.7766
	4.0	0.2007	0.1134	0.0854	0.8050

and 8 show the influence of varying sample sizes. The results do not differ significantly compared to a fixed sample size, still accuracy slightly decreases for V-AMCL and marginally increases for AMCL.

To conclude, V-AMCL seems to perform better than AMCL, both employing the likelihood field model. In both cases, this model type appears to work better with higher resolutions.

Regarding the beam model, V-AMCL performs better on real world data than AMCL. In contrast to that AMCL works better on simulated data using this model type. For both algorithms, the beam model seems to benefit from decreased resolutions. Performances of beam and likelihood field models interfere with adaptive sample sizes yield slightly worse or better results depending on the model.

3.3 Runtimes and Memory Consumption

Table 9 shows the map memory consumption as well as the mean sensor update runtimes on the real world data set for a fixed sample size for comparability. A sensor update consists of the measurement probability calculations on the whole particle set.

Table 9 Map memory consumption and mean runtimes $\bar{t}_{Likelihood}$ for the likelihood field models and \bar{t}_{Beam} for the beam models for a fixed sample size of $N = 500$

	Res.	Size (MB)	$\bar{t}_{Likelihood}$ (ms)	\bar{t}_{Beam} (ms)
AMCL	0.05	12.9	2.8	13.8
	0.10	3.6	2.6	7.3
	0.20	0.8	2.6	5.3
V-AMCL	1.0	7.7	3.3	6.3
	2.0	4.2	3.6	7.8
	4.0	2.7	4.9	11.5

Where lower grid map resolutions lead to a faster sensor update of AMCL, lower visibility lookup table resolutions lead to slower sensor updates of V-AMCL, because more line segments have to be processed. Memory consumption for grid based maps and our representation decreases proportionally to the resolution. Although V-AMCL works with vector arithmetic, we are able to achieve at least comparable and depending on the setting even faster computation times.

4 Conclusions and Future Research

We present an efficient and robust localization approach suitable for large-scale indoor environments. Furthermore, the approach fulfils several requirements introduced by the navigation problem in public buildings. Experiments on real world data reveal that the algorithm can perform very well in immensely cluttered environments utilizing CAD floor plans as map information. Although the standard implementation of AMCL performs very well in the given scenario, we still could improve results with our line segment based extension to it. Our approach achieved a higher accuracy, demands less memory and less computation time for sensor updates for higher map resolutions. Evaluation results also imply good scalability for large-scale environments.

In future work it is desired to extend the implementation with sensor models which use a point to line metric for matching likelihoods. In an indoor scenario it is most likely, that this will lead to better results. Since the field of application is indoor navigation and the algorithm works with CAD maps, one could think of introducing permeable line segments. Those could be used, for instance, to describe glass walls, as they can be found within the Bicocca data set. Besides that, it would be possible to assign an arbitrary amount of attributes to different geometrical objects in a CAD map. For example, one could enter door positions. These possibilities allow the introduction of further sensor models.

Acknowledgements The presented work is funded by the German Federal Ministry of Education and Research (BMBF Grant 01IM12005B).

References

1. Garulli, A., Giannitrapani, A., Rossi, A., Vicino, A.: Mobile robot slam for line-based environment representation. In: 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05, pp. 2041–2046 (2005)
2. Sohn, H.J., Kim, B.K.: VecSLAM: an efficient vector-based SLAM algorithm for indoor environments. *J. Intell. Rob. Syst.* **56**(3), 301–318 (2009)
3. Elseberg, J., Creed, R.T., Lakaemper, R.: A line segment based system for 2D global mapping. In: 2010 IEEE International Conference on Robotics and Automation, pp. 3924–3931 (2010)
4. Mazuran, M., Amigoni, F.: Matching line segment scans with mutual compatibility constraints. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 4298–4303 (2014)
5. Cox, I.J.: Blanche—an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Trans. Robot. Autom.* **7**(2), 193–204 (1991)
6. Sohn, H., Kim, B.: An efficient localization algorithm based on vector matching for mobile robots using laser range finders. *J. Intell. Rob. Syst.* **51**(4), 461–488 (2008)
7. Thrun, S., Burgard, W., Fox, D.: *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press (2005)
8. He, T., Hirose, S.: A global localization approach based on Line-segment Relation Matching technique. *Robot. Auton. Syst.* **60**(1), 95–112 (2012)
9. Luo, R., Li, J.X., Chen, C.T.: Indoor localization using line based map for autonomous mobile robot. In: IEEE Workshop on Advanced robotics and Its Social Impacts. ARSO 2008, pp. 1–6 (2008)
10. Ito, S., Endres, F.: W-RGB-D: floor-plan-based indoor global localization using a depth camera and WiFi. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2014)
11. Frank, D., Burgard, W., Dellaert, F., Thrun, S.: Monte carlo localization: efficient position estimation for mobile robots. In: Proceedings of the National Conference on Artificial Intelligence and the Innovative Applications of Artificial Intelligence Conference. AAAI '99/IAAI '99, American Association for Artificial Intelligence, pp. 343–349 (1999)
12. Fox, D.: Adapting the sample size in particle filters through kld-sampling. *I. J. Robotic Res.* **22**(12), 985–1004 (2003)
13. RAWSEEDS: The rawseeds project (2015). <http://www.rawseeds.org/home/>
14. Bonarini, A., Burgard, W., Fontana, G., Matteucci, M., Sorrenti, D.G., Tardos, J.D.: RAWSEEDS: Robotics advancement through web-publishing of sensorial and elaborated extensive data sets. In: Proceedings of IROS'06 Workshop on Benchmarks in Robotics Research (2006)
15. RAWSEEDS: Absolute trajectory error (ate) (2015). <http://www.rawseeds.org/rs/methods/view/9>
16. RAWSEEDS: Relative pose error (rpe) (2015). <http://www.rawseeds.org/rs/methods/view/13>

A Virtual Force Guidance Law for Trajectory Tracking and Path Following

Xun Wang, Jianwei Zhang, Daibing Zhang and Lincheng Shen

Abstract This paper presents a virtual force guidance law for trajectory tracking of autonomous vehicles. Normally, three virtual forces are designed to govern the vehicles. The virtual centripetal force counteracts the influence of the reference heading rate. The virtual spring force pulls the vehicle to the reference trajectory and the virtual drag force prevents oscillations. When local obstacles are detected, an extra virtual repulsive force is designed to push the vehicle away from its way to get around the obstacles. Using the guidance law, the reference trajectory can be straight line, circle and general curve with time-varying curvature. The guidance law is directly applicable to path-following problem by redefining the reference point. The use of artificial physics makes the guidance law be founded on solid physical theory and computationally simple. Besides, the physical meanings of the parameters are definite, which makes it easy to tune in application. Simulation results demonstrate the effectiveness of the proposed guidance law for problems of trajectory tracking, path following, and obstacle avoidance.

Keywords Trajectory tracking · Path following · Obstacle avoidance · Artificial physics · Virtual force

1 Introduction

Accurately tracking of a predefined trajectory is a basic requirement for autonomous vehicles in application. Thus, the problems of trajectory tracking and path following has been hot topics for many years. Although many approaches have been proposed

X. Wang (✉) · D. Zhang · L. Shen
College of Mechatronics and Automation,
National University of Defense Technology, Changsha 410073, China
e-mail: xunwang@nudt.edu.cn

X. Wang · J. Zhang (✉)
Group TAMS, Department of Informatics,
University of Hamburg, Hamburg 22527, Germany
e-mail: zhang@informatik.uni-hamburg.de

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_30

for these problems, there is still some work to do when considering curved trajectories, local obstacles, and real-time computing. There are various types of autonomous vehicles, namely, autonomous underwater vehicles (AUVs), unmanned surface vehicles (USVs), unmanned ground vehicles (UGVs), and unmanned aerial vehicles (UAVs) (Fig. 1). Although the dynamics associated with each type of vehicle are different, the kinematics of different vehicles with appropriate low-level controller are similar. By appropriately designing the low-level controller, kinematics of many autonomous vehicles can be approximated to a unicycle with different constraints or delays. In this paper, we focus on the problems of trajectory tracking and path following for unicycle-type autonomous vehicles, considering curved trajectories, local obstacles, and real-time computing.

Various strategies have been proposed in the literature for trajectory tracking and path following. Two main categories are control theories based approaches and geometry based approaches. Several types of control-theoretic techniques have been developed for trajectory tracking of autonomous vehicles. Some of the well-known techniques are adaptive control [1], sliding mode control [2], and linear quadratic regulator [3]. The main advantage of these control-theoretic approaches is the guarantee of the stability and the convenience for performance analysis in theory. While, obstacles avoidance during trajectory tracking remains a challenge in such control-theoretic frames. Although model predictive control based approaches [4] consider obstacles as constraints, the computational-complexity is high and all the obstacles should be known in prior. Another common drawback of control-theoretic approach is the longer man-hours associated with the controller's implementation comparing with the geometric methods.



Fig. 1 Various types of autonomous vehicles, with appropriate low-level controller, the kinematics of which can be approximated to a unicycle with different constraints or delays

Geometric guidance laws, such as pure pursuit [5] and line-of-sight (LOS) [6], were mainly proposed to address path-following problem. These guidance laws were mainly used to follow straight-line and circular paths. Although the nonlinear guidance law (NLGL) in [7] was claimed suitable for curve tracking, the cross-track error is still considerable when the curvature varies [8]. Besides, obstacles avoidance can't be address using these geometric techniques. Another popular geometric technique is the vector field (VF) based approach. Vector fields can be designed for path following [9] or obstacle avoidance during the vehicle moving towards the goal [10]. It is difficult to design the vector field when combining trajectory tracking and obstacle avoidance. Since all these techniques use the geometric approach, computing the desired heading angle (or rate) is quick and they are easy to implement. While, there is still some work to do when considering curve trajectories and the combination of trajectory tracking and obstacles avoidance.

Motivated by the above considerations, we focus on the trajectory-tracking problem on the base of artificial physics. The concept of artificial physics was proposed by Spears et al., when they worked on the problem of swarm robotics in [11]. The basic idea was from natural physics. The agents or robots react to virtual forces, which are motivated by natural physical laws. One advantage of artificial physics based approach is the solid scientific principles foundation. Inspired by the concept of artificial physics, we propose a virtual force guidance law for trajectory tracking. Normally, the vehicle is governed by three virtual forces, namely, a virtual centripetal force, a virtual spring force and a virtual drag force. The virtual centripetal force counteracts the influence of the reference heading rate. The virtual spring force pulls the vehicle to the reference point and the virtual drag force prevents oscillations. When local obstacles are detected, an extra virtual repulsive force is designed to push the vehicle away from its way to get around the obstacles. After that, other virtual forces guide the vehicle back to the reference trajectory. Using the proposed guidance law, the reference trajectory can be any sufficiently smooth time-varying bounded curve. From the view of control theory, the guidance law is equivalent to a proportional-derivative controller when tracking a straight line and is similar to a feedback linearization method when tracking circular or curved trajectories. By redefining the reference point, the guidance law is directly applicable for path-following problem.

We discuss the stability and convergence of the resulting closed-loop. We also include a comparison analysis of our approach with the NLGL approach previously published in [7]. NLGL has been proved have better performance than the PID, VF, and PLOS [8, 12]. Obtained simulation results demonstrate the superiority of the proposed guidance law when tracking a curved trajectory with time-varying curvature.

The main contribution of this work is a new virtual force based guidance law for autonomous vehicles trajectory tracking. There are four advantages with the proposed guidance law. First of all, the guidance law can be used to accurately track curved trajectory with time-varying bounded curvature. Secondly, the guidance law is computationally simple, and directly applicable in path-following problem.

Thirdly, it is convenient to combine trajectory tracking and obstacle avoidance in the frame of the virtual force guidance law. Fourthly, the use of virtual forces makes the guidance law be founded on solid physical theory, and the physical meanings of the parameters are definite, which makes the parameters easy to tune in application.

This paper is organized as follows. In the next section, we formulate the problem. In Sect. 3, we first present the virtual force guidance law along with the analysis of stability and convergence, then the guidance law is applied in path-following problem, after that, we modify the guidance law to avoid obstacles. In Sect. 4, simulation results are given to demonstrate the effectiveness of the proposed approach. In the last section, we conclude the paper.

2 Problem Formulation

In this section, we briefly describe the vehicle model and the reference trajectory, then we give the definition of the problems of trajectory tracking and path following.

2.1 Vehicle Model

We consider a unicycle-type autonomous vehicle as follows

$$\dot{x} = v \cos(\psi), \quad \dot{y} = v \sin(\psi), \quad \dot{\psi} = \omega \quad (1)$$

where the state $S = (x, y, \psi)$ denotes the inertial position and heading of the vehicle. The velocity v and the heading rate ω are control inputs.

2.2 Reference Trajectory

The desired reference trajectory S_r is produced by a dynamic trajectory smoother as follows

$$\dot{x}_r = v_r \cos(\psi_r), \quad \dot{y}_r = v_r \sin(\psi_r), \quad \dot{\psi}_r = \omega_r \quad (2)$$

where $S_r = (x_r, y_r, \psi_r)$ denotes the reference position and reference heading. v_r and ω_r are piecewise continuous.

In this paper, we use (2) to generate the desired reference trajectories and reference paths, including straight line, circle and general curve with time-varying curvature.

2.3 Problem Definition

The major difference between the trajectory-tracking problem and the path-following problem lies in the formulation of the reference path. In the trajectory-tracking problem, the reference path is time parameterized, while, in the path-following problem, time is not considered. Thus, the problems considered in this paper can be stated as follows:

Definition 1 Trajectory-tracking problem: Let $S_r(t) = (x_r(t), y_r(t), \psi_r(t))$ generated by (2) be a given reference trajectory. The purpose is to design the velocity and heading rate command inputs (v_{cmd}, ω_{cmd}) such that all the closed-loop signals are bounded and the tracking error $\|S(t) - S_r(t)\|$ converges to a neighborhood of the origin that can be made arbitrarily small.

Definition 2 Path-following problem: $S_r(\gamma)$ is a reparameterization of $S_r(t)$ by a continuous function $\gamma = \gamma(t)$. Let $S_r(\gamma)$ be a given reference path. The purpose is to design the velocity and heading rate command inputs (v_{cmd}, ω_{cmd}) such that the closed-loop signals are bounded, the tracking error $\|S(t) - S_r(\gamma)\|$ and the velocity error $|v(t) - v_r(\gamma)|$ converges to the neighborhood of the origin that can be made arbitrarily small.

2.4 Virtual Force

In our work, virtual forces are designed to govern the autonomous vehicle. The word “virtual” here means the forces are a kind of artificial forces, which don’t exist in real systems. We also use the word “virtual” to mean although we are motivated by natural physical forces, we are not restricted to them. The virtual force may have some features that are beyond the “actual” physics. For example, the virtual spring is an ideal spring and the rest length is 0. Although the forces are virtual, the vehicle acts as if they were real.

3 Approach

In this section, we first present the virtual force guidance law along with the stability and convergence analysis without considering obstacles. Then, the guidance law is applied on path-following problem. At last, we consider local obstacles during trajectory tracking.

3.1 The Virtual Force Guidance Law

For trajectory tracking, we design three virtual forces to govern the vehicle, namely, a virtual centripetal force F_c , a virtual spring force F_t and a virtual drag force F_d , as shown in Fig. 2. F_c points to the reference center $O = (x_o, y_o)$, which is determined by the reference point P and the reference radius r .

$$\begin{cases} x_o = x_r + r \cos(\psi_r + \frac{\pi}{2} \text{sgn}(\omega_r)) \\ y_o = y_r + r \sin(\psi_r + \frac{\pi}{2} \text{sgn}(\omega_r)) \end{cases} \quad (3)$$

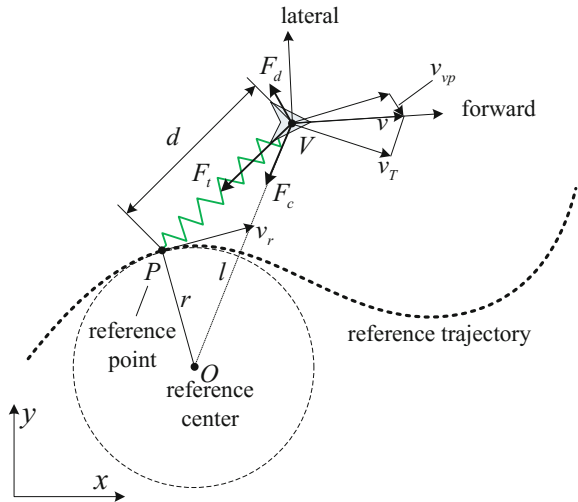
$$r = \begin{cases} |\frac{v_r}{w_r}|, & w_r \neq 0 \\ +\infty, & w_r = 0 \end{cases} \quad (4)$$

F_t points to P , and F_d points to the opposite of the velocity of the vehicle V relative to P . Assume the vehicle has a unit mass, i.e., $m = 1$, then the absolute value of the virtual forces are given by

$$F_t = kd, \quad F_d = cv_{vp}, \quad F_c = \frac{v_T^2}{l} \quad (5)$$

where $k > 0$ is the spring constant, d is the distance from the vehicle to the reference point, $c > 0$ is the drag coefficient, v_{vp} is the velocity of the vehicle relative the reference point, v_T is the velocity component in the normal direction of OV , and l is the distance between V and O . Among these variables, only k and c are the control parameters, and the others can be calculated according to the reference and the actual states of the vehicle.

Fig. 2 Diagram of the virtual force guidance law for trajectory tracking



Decomposing the virtual forces in the forward and lateral direction of the vehicle, we get the resultant forces in the two directions

$$F_f = F_{tf} + F_{df} + F_{cf}, \quad F_l = F_{tl} + F_{dl} + F_{cl} \tag{6}$$

where the subscripts *f* and *l* mean the components in the forward and lateral direction. Then the virtual force guidance law for trajectory-tracking problem is given by

$$v_{cmd} = v + \Delta T F_f, \quad \omega_{cmd} = \frac{F_l}{v} \tag{7}$$

where ΔT is the time step in application.

3.2 Stability and Convergence Analysis

Straight-Line Tracking When the vehicle tracks a straight line, $\omega_r = 0$. Then the reference radius $r = \infty$, and $F_c = 0$. Accordingly the heading rate and the velocity command inputs are determined only by the virtual spring force F_l and the virtual drag force F_d as shown in Fig. 3.

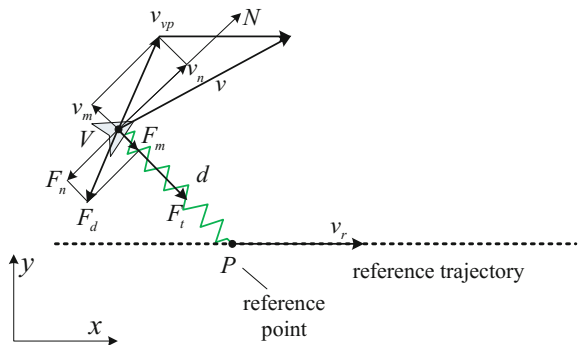
By decomposing the virtual drag force F_d and the relative velocity v_{vp} along VP and its normal N , we can analyze the dynamics of the vehicle in the two vertical directions.

In the direction of N , the vehicle has an initial velocity $v_n(0) = v_{n0}$ and a drag force $F_n = cv_n$. The direction of the drag force is always opposite to v_n . By solving the following equation

$$\dot{v}_n(t) = -cv_n(t), \quad v_n(0) = v_{n0} \tag{8}$$

we get

Fig. 3 Diagram for straight-line trajectory tracking



$$v_n(t) = v_{n0}e^{-ct} \quad (9)$$

For $c > 0$, v_n exponentially converges to 0.

Then the dynamics in the VP direction can be given by

$$\ddot{d} = -kd - c\dot{d} \quad (10)$$

Let $\mathbf{X} = [x_1, x_2]' = [d, \dot{d}]'$, the differential equation (10) is transformed to

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X}, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ -k & -c \end{bmatrix} \quad (11)$$

Then, let $\dot{\mathbf{X}} = 0$, we get the only equilibrium point $\hat{d} = \hat{x}_1 = 0$, and $\hat{\dot{d}} = \hat{x}_2 = 0$. The characteristic equation of (11) is given by

$$\det(\lambda\mathbf{I} - \mathbf{A}) = \lambda^2 + c\lambda + k = 0 \quad (12)$$

Accordingly, the characteristic roots are

$$\lambda = \frac{-c \pm \sqrt{c^2 - 4k}}{2} \quad (13)$$

For the spring constant $k > 0$, and the drag coefficient $c > 0$, we always have $\text{real}(\lambda) < 0$, which means the system (11) is asymptotically stable. Consequently, the tracking error globally converges to 0 using the guidance law (7).

From (13), we know that: the tracking error d converges to 0 without any overshoot or oscillation, if the parameters satisfy $c \geq 2\sqrt{k}$ and $k > 0$. It is worthwhile to mention that the virtual force guidance law is equivalent to a proportion differential controller with $e = d$, $\dot{e} = -\dot{d}$, $K_p = k$, and $K_D = c$.

Circular Trajectory Tracking When the vehicle tracks a circular trajectory, $r = |\frac{v_x}{\omega_r}| = \text{const}$ is the constant radius of the reference circle. Without the virtual centripetal force, which means $F_c = 0$, F_t and F_d force the vehicle to track the reference circular trajectory if the spring force F_t is big enough to provide the centripetal force, but there will be a steady tracking error d_s as shown in Fig. 4a, b.

By force analysis, we know that $d_s \geq r_s \geq 0$, and $d_s \rightarrow r_s$ when the positive drag coefficient c is small enough and approaches 0. By solving the following equation

$$kr_s = \frac{v_T^2}{l} = l\omega_r^2 = (r + r_s)\omega_r^2 \quad (14)$$

we get

$$r_s = \frac{r\omega_r^2}{k - \omega_r^2} \quad (15)$$

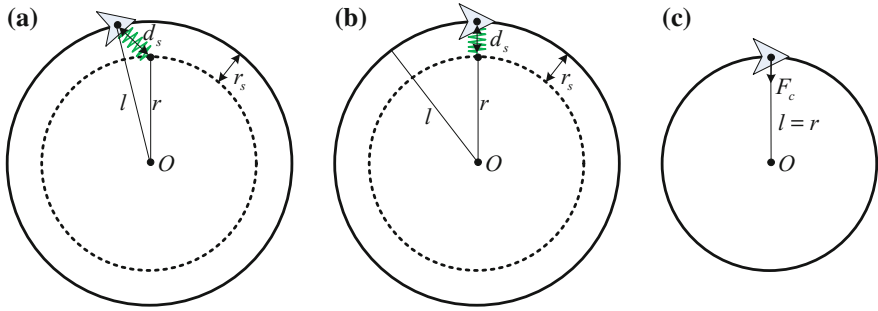


Fig. 4 Diagram for the equilibrium states of circular trajectory tracking with: **a** $F_c = 0$ and $c > 0$, **b** $F_c = 0$ and $c \rightarrow 0$, **c** $F_c = \frac{v_r^2}{l}$

We know that $r_s > 0$. It requires the spring constant satisfies

$$k > \omega_r^2 \tag{16}$$

From Eq. (15), we know that $r_s \rightarrow 0$ when $k \rightarrow \infty$. While, in reality, the value of k is a reflection of the maneuverability of the vehicle, which means k is bounded. Accordingly, the steady tracking error $d_s > 0$ always exists without designing the virtual centripetal force.

On the contrary, by the virtual centripetal force $F_c = \frac{v_r^2}{l}$, the vehicle orbits a concentric circle with a radius of l . Then the virtual spring force will pull the vehicle close to the concentric reference circle. Meanwhile, the virtual drag force prevents oscillations. When the vehicle converges to the reference trajectory, the virtual forces $F_t = F_d = 0$ due to $d = 0$ and $v_{vp} = 0$. Then the virtual centripetal force $F_c = \frac{v_r^2}{r}$ ensures the vehicle orbits the reference circle.

For circular trajectory tracking, the virtual centripetal force counteracts the influence of the reference heading rate, so that we can use the virtual spring force F_t and drag force F_d to guide the vehicle. Then the vehicle tracks a circle like tracking a straight line, and the dynamics about the tracking error d is similar to the straight-line case. Accordingly, the tracking error d globally converges to 0 using the guidance law.

Curved Trajectory Tracking When the vehicle tracks a general curve, the reference radius $r = |\frac{v_r}{\omega_r}|$ varies. Then the reference center $O = (x_o, y_o)$ determined by (3)

moves with the reference point. The virtual centripetal force $F_c = \frac{v_r^2}{l}$ pointing to the reference center O still provides the centripetal force that ensures the vehicle orbits the moving reference center. Similar to circle case, the virtual centripetal force counteracts the influence of the reference heading rate, the virtual spring force F_t pulls the vehicle close to the reference point, and the virtual drag force prevents oscillations.

From the view of control theory, the virtual force guidance law is similar to a feedback nonlinearization method, when tracking a circular or curved trajectory. The feedback linearization approach involves coming up with a transformation of the nonlinear system into an equivalent linear system through a change of variables and a suitable control input. In the virtual force guidance law, the virtual centripetal force F_c is a nonlinear function of the states of the vehicle and the reference point. By introducing the nonlinear virtual centripetal force as a feedback and analyzing the dynamics of the tracking error d , the control input is equivalently transformed to \ddot{d} . Then the resulting closed-loop system about d is a second order linear system, which is equivalent to a PD controller. Analogously, there will be no overshoot during the convergence of d , if the parameters satisfy $c \geq 2\sqrt{k}$ and $k > 0$.

3.3 Application in Path-Following Problem

The main difference between the trajectory-tracking problem and the path-following problem lies in the formulation of the reference path. In the trajectory-tracking problem, the reference path is time parameterized, while, in the path-following problem, time is not considered. Then we have to redefine the reference point on the reference path.

For the path-following problem, the closest point to the vehicle is selected as the reference point on the reference path. Then, the proposed virtual force guidance law is directly applicable for the problem of path following without any modification. The stable and the convergence analysis above are also applicable for the path-following problem.

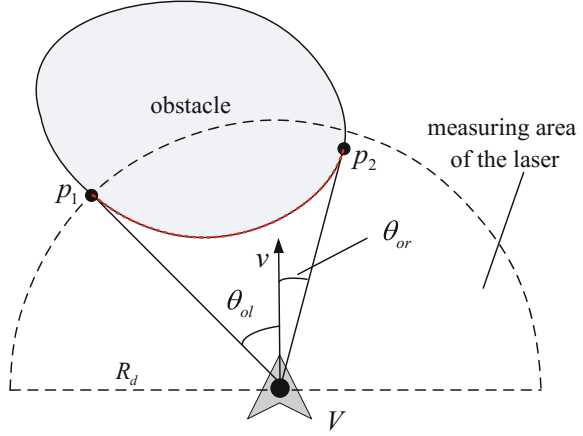
3.4 Obstacle Avoidance

It is important for an autonomous vehicle to dynamically avoid local obstacles when tracking a predefined trajectory. It requires the flexibility to move out of its way to get around the obstacles. After that, it should converge back to the reference trajectory. Based on the virtual force guidance law, the ability of obstacle avoidance can be easily achieved by properly designing a virtual repulsive force from the obstacle.

We assume that the vehicle is equipped with a laser to sense local obstacles. We get a point set $P_o = \{(d_o, \theta_o) | d_o \leq R_d, -\frac{\pi}{2} < \theta_o < \frac{\pi}{2}\}$ on the surface of the obstacle in the local polar coordinates, when an obstacle is detected, where the positive direction of θ_o is counter clockwise, and R_d is the detect radius of the laser as shown in Fig. 5.

Accordingly, two points $p_1 = (d_{ol}, \theta_{ol})$ and $p_2 = (d_{or}, \theta_{or})$ can be found from P_o , where θ_{ol} and θ_{or} are the leftmost and the rightmost point detected by the laser from the view of the vehicle.

Fig. 5 Diagram for the obstacle avoidance



We design an extra virtual repulsive force that repels the vehicle to the lateral direction, which means the obstacle only affects the heading rate of the vehicle. Without any prior knowledge of the world, it is reasonable for the vehicle to avoid the obstacle along the side with a smaller angle of view. Accordingly, the obstacle repels the vehicle to the right when $|\theta_{or}| \leq |\theta_{ol}|$, to the left when $|\theta_{or}| > |\theta_{ol}|$.

Considering the unit mass assumption, we design the extra virtual repulsive force F_o as follows

$$F_o = k_o l_o \tag{17}$$

$$l_o = \begin{cases} -(2d_{saf} + d_{ol}\theta_{ol}) & \theta_{or} + \theta_{ol} \leq 0, 2d_{saf} + d_{ol}\theta_{ol} > 0 \\ (2d_{saf} - d_{or}\theta_{or}) & \theta_{or} + \theta_{ol} > 0, 2d_{saf} - d_{or}\theta_{or} > 0 \\ 0 & \text{else} \end{cases} \tag{18}$$

where k_o is the repulsive force constant, and d_{saf} is the allowed minimum clearance between the agent and the obstacle. In most cases, collision with an obstacle is destructive for the vehicle. Thus, the repulsive force constant k_o should much larger than the spring constant k (i.e. $k_o \gg k$).

Then, the heading rate command input ω_{cmd} in the guidance law (7) can be modified by the extra virtual repulsive force as follows

$$v_{cmd} = v + \Delta TF_f, \quad \omega_{cmd} = \frac{F_l}{v} + \frac{F_o}{v} \tag{19}$$

This is not the only way to modify the virtual force guidance law to avoid obstacles, one can design other kinds of repulsive force to achieve it. This simple modification is to illustrate the convenience to consider obstacle avoidance in the frame of the virtual force guidance law.

4 Simulation

This section presents simulation results to illustrate theoretical results proposed in this paper. The simulation includes three parts, namely, trajectory tracking, path following, and obstacle avoidance. In the path-following simulation, we include a comparison analysis of our approach with the NLGL approach. Throughout the simulations, the unit for distance, velocity, angle and angle rate are m, m/s, rad and rad/s (2) is used to generate the reference trajectories. The reference velocity keeps $v_r = 20$. The guidance law (7) and (19) are applied.

4.1 Trajectory Tracking

The reference trajectories of straight line, circle, and curve are generate by (2) with $\omega_r = 0$, $\omega_r = -0.1$, and $\omega_r = -0.1 \sin(\frac{\pi t}{30})$, respectively. Different k and c are applied in (7) to evaluate the performance of the guidance law for trajectory tracking. Simulation results for straight line, circle and curve tracking are shown from Figs. 6, 7, 8 respectively.

From Figs. 6, 7, 8, we find that, with all of the c and k , the vehicle converges to the reference trajectories including straight line, circle and curve with time-varying curvature. While, the convergence processes are different with different c or k . Figures 6a, 7a, 8a shows that: there are overshoots before the convergence of d , when $c = \sqrt{k}$ and $c = 1.5\sqrt{k}$. The vehicle converges to the reference trajectories without any overshoot, when $c = 2\sqrt{k}$ and $c = 3\sqrt{k}$. It takes a longer time to converge when $c = 3\sqrt{k}$ than $c = 2\sqrt{k}$. From Figs. 6b, 7b, 8b, we find that the convergence time can be shorten by increasing k and the guidance law has a large parameter adaptation.

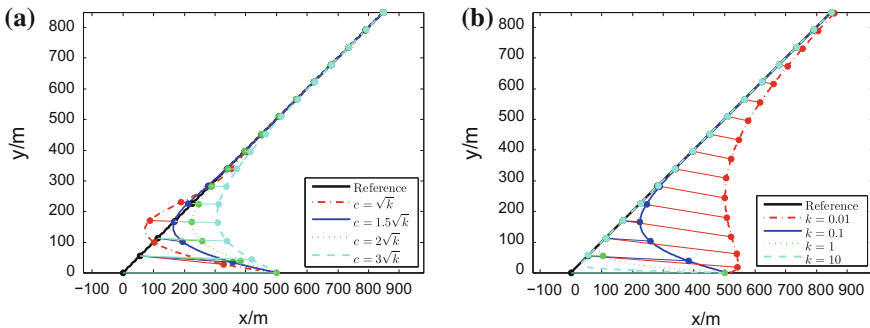


Fig. 6 Vehicle trajectories in the simulation for straight-line tracking with $S(0) = (500, 0, 0)$ and: **a** $k = 0.1$ and different c , **b** $c = 2\sqrt{k}$ and different k

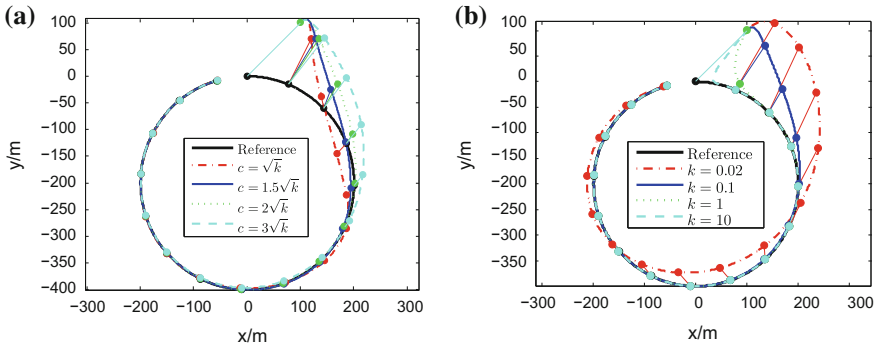


Fig. 7 Vehicle trajectories in the simulation for circle tracking with $S(0) = (100, 100, \frac{\pi}{4})$ and: **a** $k = 0.1$ and different c , **b** $c = 2\sqrt{k}$ and different k

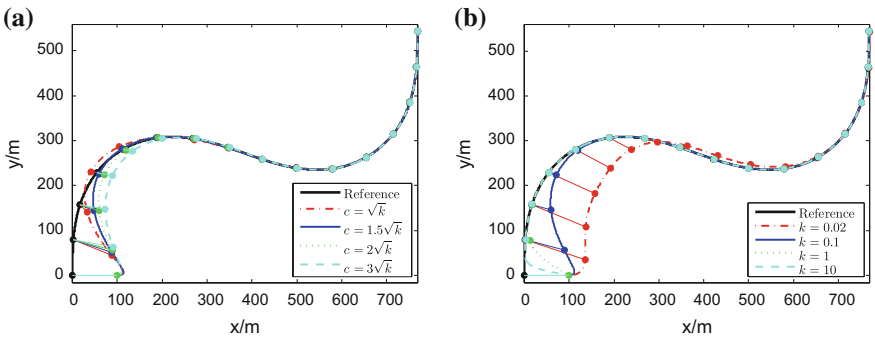


Fig. 8 Vehicle trajectories in the simulation for curve trajectory tracking with $S(0) = (100, 0, 0)$ and: **a** $k = 0.1$ and different c , **b** $c = 2\sqrt{k}$ and different k

4.2 Path Following

The virtual force guidance law can be directly applied to address the path-following problem. To illustrate the advantage of the proposed guidance law, we compare the virtual force guidance law with the NLGL proposed in [7]. The performance of NLGL was evaluated on fixed-wing UAVs in [8, 12], and it performed over the PID, VF, and PLOS. For the NLGL, we use the recommend parameters $L = 0.6r$ [8] in the simulation. For the virtual force guidance law, we use $k = 0.1$ and $c = 2\sqrt{k}$. The reference path is generated by (2) with

$$\omega_r(t) = \begin{cases} 0 & , t \in [0, 4] \\ -0.1 & , t \in (4, 30] \\ 0.1 & , t \in (30, 60] \\ 0.1 + 0.1 \sin(\frac{2\pi(t-60)}{80}) & , t \in (60, 80] \\ 0.2 & , t \in (80, 90] \end{cases} \quad (20)$$

Figure 9 shows the trajectories of the vehicle using the two guidance laws. The peak cross-track error of the NLGL is found to be over 15 m. For the virtual force guidance law, the cross-track error is 0 throughout the simulation. The result shows that the virtual force guidance law performs over the NLGL approach, especially when the reference curvature varies.

Actually, in the trajectory-tracking and path-following problems, the reference trajectory (or path) is predefined by the user or generated by path planning algorithms, which means the reference path is always fully known. Accordingly, the reference heading rate ω_r is also accurately known in advance. Therefore, it is reasonable to make use of the reference heading rate to improve the tracking accuracy when the reference is a curve with time-varying curvature. In the virtual force guidance law, the virtual centripetal force determined by the reference point and the reference heading rate counteracts the influence of the varying reference heading rate. Consequently, the guidance law can be applied to accurately track a general curve. While, the reference heading rate is not considered in the NLGL. It is the reason for the large cross-track error when the NLGL is applied in curve tracking, especially when the reference curvature varies.

Fig. 9 Trajectories in the comparison between the virtual force guidance law and NLGL

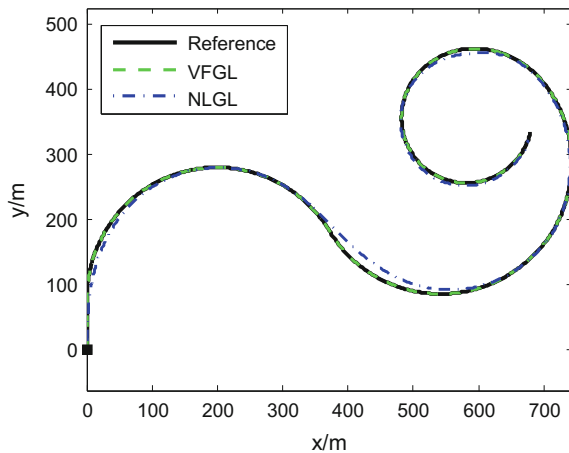
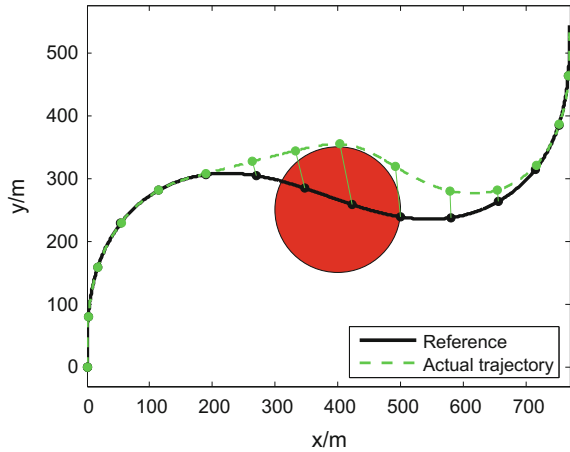


Fig. 10 Vehicle trajectory in the simulation of trajectory tracking with obstacle avoidance



4.3 Combined Obstacle Avoidance

In this scenario, an obstacle with a radius of 100 is located at (400, 250). The reference trajectory is same with the curve in Sect. 4.1. The modified virtual force guidance law (19) is applied with parameters as follows:

$$k = 0.1, c = 2\sqrt{k}, k_o = 10, d_{saf} = 5, R_d = 200 \quad (21)$$

Figure 10 shows the result of obstacle avoidance during trajectory tracking. The vehicle is pushed out of its way by the repulsive force to get around the obstacle. After that, it converges back to the reference curve.

5 Conclusion

In this paper, we focused on the guidance of unicycle-type autonomous vehicles. A virtual force guidance law was proposed for trajectory tracking and path following. Virtual forces were designed to determine the velocity and heading rate command inputs. The use of artificial physics made the guidance law be founded on solid physical theory and computationally simple. Besides, the physical meanings of the parameters are definite, which makes it easy to tune in application. Simulation results demonstrated the performance of guidance law for trajectory tracking, path following and obstacle avoidance. The noneconomic constraints and the dynamic environment with dynamic obstacles will be considered in the future work. Further results will be tested on the platform developed in [13, 14].

References

1. Brezoescu, A., Espinoza, T., Castillo, P., Lozano, R.: Adaptive trajectory following for a fixed-wing uav in presence of crosswind. *J. Intell. Robot. Syst.* **69**(1–4), 257–271 (2013)
2. Chwa, D.: Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates. *IEEE Trans. Control Syst. Technol.* **12**(4), 637–644 (2004)
3. Ratnoo, A., Sujit, P., Kothari, M.: Adaptive optimal path following for high wind flights. In: 18th IFAC World Congress, Milano, Italy, vol. 2 (2011)
4. Howard, T.M., Green, C.J., Kelly, A.: Receding horizon model-predictive control for mobile robot navigation of intricate paths. In: *Field and Service Robotics*, pp. 69–78. Springer (2010)
5. Conte, G., Duranti, S., Merz, T.: Dynamic 3d path following for an autonomous helicopter. In: *Proceedings of the IFAC Symposium on Intelligent Autonomous Vehicles* (2004)
6. Fossen, T.I., Breivik, M., Skjetne, R.: Line-of-sight path following of underactuated marine craft. In: *Proceedings of the 6th IFAC MCMC, Girona, Spain*, pp. 44–249 (2003)
7. Park, S., Deyst, J., How, J.P.: A new nonlinear guidance logic for trajectory tracking. In: *Proceedings of the AIAA Guidance, Navigation and Control Conference*, pp. 1–16 (2004)
8. Park, S., Deyst, J., How, J.P.: Performance and lyapunov stability of a nonlinear path following guidance method. *J. Guid. Control Dyn.* **30**(6), 1718–1728 (2007)
9. Nelson, D.R., Barber, D.B., McLain, T.W., Beard, R.W.: Vector field path following for miniature air vehicles. *IEEE Trans. Robot.* **23**(3), 519–529 (2007)
10. Kazem, B.I., Hamad, A.H., Mozael, M.M.: Modified vector field histogram with a neural network learning model for mobile robot path planning and obstacle avoidance. *Int. J. Adv. Comp. Techn.* **2**(5), 166–173 (2010)
11. Spears, W.M., Spears, D.F., Hamann, J.C., Heil, R.: Distributed, physics-based control of swarms of vehicles. *Auton. Robots* **17**(2–3), 137–162 (2004)
12. Sujit, P., Saripalli, S., Borges Sousa, J.: Unmanned aerial vehicle path following: a survey and analysis of algorithms for fixed-wing unmanned aerial vehicles. *IEEE Control Syst.* **34**(1), 42–59 (2014)
13. Wang, X., Kong, W., Zhang, D., Shen, L.: Active disturbance rejection controller for small fixed-wing uavs with model uncertainty. In: *2015 IEEE International Conference on Information and Automation*, pp. 2299–2304. IEEE (2015)
14. Wang, X., Zhang, J., Zhang, D., Shen, L.: Uav formation: from numerical simulation to actual flight. In: *2015 IEEE International Conference on Information and Automation*, pp. 475–480. IEEE (2015)

A 2D Voronoi-Based Random Tree for Path Planning in Complicated 3D Environments

Zheng Fang, Chengzhi Luan and Zhiming Sun

Abstract Path planning in complicated 3D environments with narrow passages and rooms is a challenging problem, which is usually time consuming for geometric searching methods or incomplete for sampling-based methods. Focusing on these issues, this paper presents a new algorithm named 2D Voronoi-based Random Tree which combines the completeness of the voronoi diagram based 2D path searching methods and the efficiency of sampling based path planning methods. In this method, 2D voronoi diagram is created and used to guide the growth of random trees in complicated 3D environments. In each iteration of random trees growth, a new node on voronoi edges is selected by moving forward to the target with a fix step length along edges. And then, the 3D particles are distributed locally around this node and be selected according to a cost function for the random trees growth. By doing so, this method can find a valid path in complicated 3D environments with narrow passages and rooms while improving its efficiency and completeness. To demonstrate its effectiveness, efficiency and robustness, the proposed method is examined and compared with RRT algorithm in various practical complex 3D environments.

Keywords Path planning · 2D voronoi diagram · Random trees · Complicated 3D environment

1 Introduction

In robotics, path planning is a fundamental research topic which focus on finding a path from the start point to the target point while avoiding obstacles in the way. And nowadays, path planning not only plays an important role in automated

Z. Fang (✉) · C. Luan (✉) · Z. Sun
State Key Lab of Synthetical Automation for Process Industries,
Northeastern University, Shenyang, China
e-mail: fangzheng@mail.neu.edu.cn

C. Luan
e-mail: sdwflcz@163.com

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_31

manufacturing and mobile robots, but also has been widely used in computer animations, medical surgery, and molecular biology [1]. There are many kinds of path planning algorithms. According to the environmental dimension, these methods can be divided into 2D, 2.5D, 3D and higher dimensional path planning algorithms. Among them, 2D geometric searching algorithms are the most studied and maturest path planning methods, such as Dijkstra [2], A* [3], Theta* [4], 2D voronoi digram based path planning methods [5], etc. Although the 2D geometric searching methods can keep the completeness of algorithms and get a valid final path timely, it request that the space must be a 2D or 2.5D environment. However, the real world usually is a more complicated 3D environment. Therefore, it will be time consuming when those 2D geometric searching methods are used in practical 3D environments. At the same time, it also consumes a lot of computing and storage resources which are limited especially on the onboard computer system like a micro aerial vehicle. For solving these problems, many sampling based path planning methods have been proposed to effectively reduce the resource consumption while improving computational efficiency. The commonly using algorithms are the variants of Probabilistic Roadmap Method (PRM) [6], Rapidly-Exploring Random Trees (RRT) [7] and Fast Matching Trees (FMT) [8]. However, due to the randomness and nondeterminacy of sampling techniques, sampling based path planning methods could not guarantee the completeness of an algorithm, which means that the algorithm maybe can not find a valid path even though it actually exists. Especially, sampling based methods could not find a valid path even after multiple iterations in environments with narrow passages and rooms. Therefore, the main problem for path planning in complicated 3D environments is the contradiction between completeness and efficiency.

Focusing on these issues, we propose a 2D voronoi-based random trees algorithm which can solve these issues in complicated 3D environments. For this method, there are two phases in each iteration of 3D random trees growth. Firstly, a new node on 2D voronoi edges is selected by moving forward to the target with a fix step length along edges. Secondly, 3D particles are distributed locally around this node and be selected according to a cost function for the random trees growth. By combining the 2D voronoi diagram and sampling techniques, this algorithm can find the near-shortest and safest path while keeping the property of completeness of voronoi based path planning algorithms and efficiency of sampling based path planning algorithms.

2 Related Work

As introduced in Sect. 1, there are many kinds of path planning algorithms which are designed for different environments. For giving an over view and comprehensive understanding, some algorithms are discussed in the following three parts.

2.1 2D Geometric Searching Methods

Dijkstra is a classical path planning algorithm for finding the shortest path between two points due to its optimization capability [2]. By combining the advantages of Dijkstra and Best-First Search methods (BFS), A* [3] algorithm is faster than Dijkstra algorithm. However, both Dijkstra and A* method will consume a lot of time when the map is large or the grid resolution is high. Basic Theta* and Angle-Propagation Theta* are both variants of A* which propagate information along grid edges to achieve a short runtime without constraining paths to grid edges for finding any-angle paths. However, Basic Theta* and Angle-Propagation Theta* can not guarantee to find the true shortest paths [4]. Voronoi based path planning algorithms are combinational path searching methods which can significantly reduce the search problem and can be used to generate the n-best paths [9]. However, it is computationally expensive to create a voronoi graph for 3D environments, which is not suitable for robots with limited onboard computing resources like a micro aerial vehicle.

2.2 3D Sampling-Based Path Planning Algorithms

PRM [6] is easier to implement and more efficient than traditional geometric searching methods for solving the path planning problems for robots with many degree of freedoms. If optimality is relaxed, Sequential Roadmap Spanner (SRS) [10] can produce sparser graphs by not including all edges. The PRM* [11] can guarantee that solutions converge to optimal as iteration increasing. ANC-Spatial [12] uses localized learning to select appropriate connection during PRM roadmap construction. However, because of the randomness and nondeterminacy of sampling techniques, these method cannot guarantee to find a valid path in narrow passage environments.

RRT [7] can quickly search high-dimensional spaces that have both algebraic and differential constraints to find out a valid path. However, this algorithm also could not guarantee the optimality. As a modification of RRT, Lower Bound Tree-RRT (LBT-RRT) [13] is asymptotically near-optimal. And, RRT* can ensure asymptotic optimality of the solution while maintaining the probabilistic completeness and the computational efficiency of the RRT [11]. Even though the performance of RRT algorithm has been improved greatly, there are still some problems need to be solved. For example, narrow passages are very common in indoor environments. However, the random trees will not grow obviously even after many iterations in this kinds of region, so this is still an open problem.

2.3 Hybrid Methods

In 2001, Mark presented a hybrid path planning algorithm by combining the advantages of 3D generalized voronoi graph (GVG) and sampling techniques [14]. Though this method can find out a valid path, it is computationally expensive for GVG

computation and point path searching in high dimensional environments. Hybrid PSO-GSA Robot Path Planning Algorithm [15] which can generate an optimal collision-free trajectory in static environments is proposed by combining Gravitational Search Algorithm (GSA) and a Particle Swarm Optimization (PSO) algorithm. However this method runs off-line, which is not suitable for autonomous robots. A hybrid strategy based on interfered fluid dynamical system (IFDS) and improved rapidly-exploring random tree (IRRT) is proposed in [16]. However this algorithm is computationally expensive and consumes lots of storage resources.

3 2D Voronoi-Based Random Trees

Inspired by hybrid path planning methods, we propose a path planning algorithm based on 2D GVD and 3D random particles to improve the efficiency and reduce the computational consumption while keeping the completeness of a path planning algorithm. In this algorithm, firstly we generate a 2D grid map by cutting a slice from the 3D Octomap at a specified height. Then, we generate a 2D generalized voronoi diagram from the 2D grid map. After that, we can use the generated 2D GVD to guide the growth of random trees. However, if we grow the random trees like that, the random trees can only grow at a specified height of the 3D environments. Therefore, some valid path may be missed in complicated 3D environments. For example, as shown in Fig. 1, if we cut the 2D slice like the red line, the table will be considered as an obstacle. However, if we cut the 2D slice like the green or blue line, it will be considered as free. And it is very difficult to know what height is the best position to cut the 2D slice. For solving this problem, a number of 3D random particles will be distributed locally along the selected 2D GVD edge nodes. And then these particles will be selected by a cost function to grow the random trees in each iteration. By

Fig. 1 Obstacles in complicated 3D environment: if we cut the 2D slice like the *red line*, the table will be considered as an obstacle when creating the 2D GVD. However, if we cut the 2D slice like the *green line* or *blue line*, it will be considered as free



doing so, we actually grow the random tree in local 3D space (not constrained to the 2D GVD), therefore it will enhance the possibility of finding a valid or optimal path.

3.1 2D Voronoi Diagram Creating

The Generalized Voronoi Diagram (GVD) is defined as a set of points in free space to which the two closest obstacles have the same distance. This characteristic significantly reduces the search problem and can be used to generate the n-best paths for offering route alternatives to users or for optimization-based motion planning. However, creating a 3D generalized voronoi graph is computationally expensive, especially for the onboard computer system of robots. Therefore, 2D generalized voronoi diagram is used to reflect the real 3D indoor environments in this paper. The processes of creating 2D GVD are shown below, and the results of the implementation are shown in Fig. 2.

1. Extract the voxels at a specified height of the world map (3D OctoMap) [17] to create a 2D grid map.
2. Traverse the whole 2D grid map to check whether it is free or occupied for each grid cell.
3. Extract the boundaries of the static obstacles which are represented by occupied cells.
4. Create a distance map (DM) with the brushfire algorithm and the obstacle boundary cells.
5. Create a 2D generalized voronoi diagram (GVD) according to the DM.
6. Prune the GVD with the skeletonization-based approach to get a simplified GVD.

3.2 3D Random Trees Growing

The edges of 2D voronoi diagram correspond to the safest routes for robots moving with respect to obstacles. However, this kinds of edges can not be used to search for a valid path in complicated 3D environments directly, because there may be many

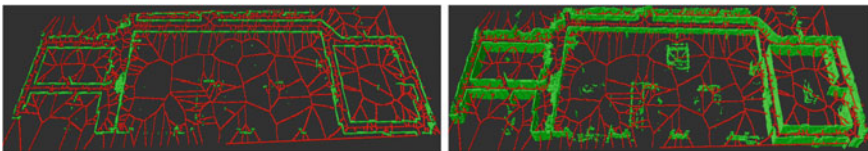


Fig. 2 2D voronoi diagram in 2D and 3D maps: the *left picture* represent the 2D grid map extracted from a 3D environment shown in the *right picture*, in which, the *green cells* is the obstacles and the *red lines* are the voronoi edges

different kinds of obstacles in indoor environments, such as tables, chairs, walls with doors or windows and so on. Although the robots can pass through some kinds of obstacles from top, middle or bottom, the 2D voronoi diagram at specified height could not cover all possible ways for all of those different obstacles, which will result in failure for path searching in complicated 3D environments. But, the 2D voronoi edges are still valuable for guiding the path searching in some senses. For solving this problem, 3D random trees are constructed to help to pass through this kinds of obstacles which 2D voronoi diagram could not. The strategy to construct a 3D random trees is that a number of 3D random particles are distributed locally around 2D voronoi edge nodes and be selected according to the cost function (1) to grow the random trees in each iteration.

$$F_1(x_i) = G_1(x_i) + H_1(x_i) + D_1(x_i) \quad (1)$$

where x_i is the current selected particle, $G(x_i)$ is the actual cost relative to the Euclidean distance from the start point to the current particle x_i , $H(x_i)$ is the estimate cost relative to the Euclidean distance from the current particle x_i to the target point which is a heuristic function, $D(x_i)$ is the cost relative to the Euclidean distance from the current particle x_i to the nearest random trees node, $F_1(x_i)$ is the total cost of the selected particle x_i . And for finding the optimal path, the particle which has the minimal total cost value will be selected as the next random trees node.

Considering that there are many voronoi edge nodes in a 2D voronoi diagram, this paper tries to select the best node which can help the random tree to grow more quickly toward the target by computing the cost function (2):

$$F_2(k) = G_2(k) + H_2(k) \quad (2)$$

where k is the current voronoi edge node, $G_2(k)$ is the actual cost relative to the Euclidean distance from the start point to the current voronoi edge node k , $H_2(k)$ is the estimate cost relative to the Euclidean distance from the current voronoi edge node k to the target point which is a heuristic function, $F_2(k)$ is the total cost of the selected voronoi edge node k . And for reaching the target as soon as possible, the node with minimal cost $F_2(k)$ will be selected as the next searching branch.

3.3 Algorithm Structure and Pseudo-Code

For comprehensive understanding, the structure of 2D voronoi-based random trees algorithm is described as below:

1. Extract the point cloud at a specified height of the 3D environments and create a 2D grid map.
2. Create the generalized voronoi diagram (GVD) of the 2D grid map.

3. Search the 2D GVD edges from the start point to the target point with a cost function and distribute 3D random particles locally.
4. Select the best particle and check collision to confirm whether this particle can be used to grow the random trees in each iteration.
5. Query the random trees from the target point backward to start point to get the shortest path.
6. Shorten and smooth this path to obtain a feasible one.

The details of our algorithm are shown in the pseudo-code Algorithm 1. In which, *cloud* is the 3D map represented as point cloud, x_{init} and x_{goal} are start and target point respectively, r is the radius for local particles distribution and voronoi diagram edge node searching, k is the max iteration number, τ is the variable which store the random trees for path querying, *path* is the queried path and *spath* is the final feasible path. The variable *temp* is used to store all the candidate edge nodes in each iteration, p is the selected best particle from ps which have the minimal cost value of $F_1(k)$ in Eq. (1), n is the best node selected from *temp* which has the minimal cost value of $F_2(k)$ in Eq. (2), the *branch* is used to record the nodes that have not been used in each iteration and select the best node when we can not find a valid node in *temp*.

Algorithm 1 : 2D Voronoi-based Random Trees

```

VB_RT(cloud,  $x_{init}$ ,  $x_{goal}$ ,  $r$ , smooth_path)
1  gm = EXTRACT_2D_GRID_MAP(cloud);
2  vm = GET_VORONOI_DIAGRAM(gm);
3   $\tau$  = SET_TO_TREE( $x_{init}$ );
4  temp = FIND_VNODE(vm,  $x_{init}$ );
5   $n$  = BETS_NODE(temp,  $x_{goal}$ );
6  for  $i \leftarrow 1$  to  $k$ 
7      do
8          ps = DISTRIBUTE_PARTICLES( $n$ ,  $r$ );
9           $p$  = BEST_PARTICLES( $\tau$ , ps,  $x_{init}$ ,  $x_{goal}$ );
10          $\tau$  = SET_TO_TREE( $p$ );
11         if IS_FINISH( $\tau$ ,  $x_{goal}$ )
12             then
13                  $\tau$  = SET_TO_TREE( $x_{goal}$ );
14                 path = FIND_PATH( $\tau$ );
15                 smooth_path = SMOOTH_PATH(path);
16                 return smooth_path;
17             else
18                 branch = BRANCH(temp);
19                 temp = GET_VNODE( $n$ );
20                  $n$  = BETS_NODE(temp,  $x_{goal}$ );
21         smooth_path.clear();
22     return 0

```

4 Experiments

4.1 Experiments Setup

In this section, we tested our method and compared it with basic RRT algorithm on a computer with 3.6 GHz Inter® Core™ CPU and 8 GByte RAM. We used Ubuntu 14.04 LST as the operating system and developed our algorithm using C++. In order to reduce the consumption of memory, we use OctoMap to represent the real 3D environments. And in our examinations, the parameters we used are listed in Table 1. We carried out two experiments to show the efficiency and robustness of our algorithm respectively and the experiments video can be found in the following link: <https://www.youtube.com/watch?v=iFCHn2Yxw1o>.

4.2 Efficiency Validation

For efficiency, we tested our algorithm and compared it with basic RRT method in the same environment. The environment is composed of more than 8,700,000 points with a size of length 60 m, width 20 m and height 3 m. And there are rooms, long corridors and an atrium in this environment, which is actually a complicated 3D environment, as shown in Figs. 3, 4 and 5. Both of the two algorithms were repeated 100 times and the average results are listed in Table 2. For simplification, our 2D voronoi based random trees path planning algorithm is denoted as VB-RT.

For validating the completeness of VB-RT algorithm which is a precondition of efficiency, the start and the target points are set in right and left room respectively as shown in Figs. 3 and 4. There are long corridors around these two rooms, which is the most difficult problem for random based path planning algorithms. As shown in Table 2, we can see from the first two rows that the VB-RT algorithm just needed few iterations to construct a bigger random trees while basic RRT algorithm reached the max iterations and failed to find a feasible path. The final path of our algorithm is shown in Fig. 4 and the traversed voronoi edges are marked with white color as

Table 1 Experiments parameters: VN-R is the voronoi edge node searching radius, V-H is the 2D voronoi diagram high in 3D environments, P-N is the local particle number, P-R is the particle distributed radius, I-N is the maximum iteration number for random trees growth, C-R is the collision checking radius, SI-N is the smoothing iteration number, max and min represent the maximum and minimum coordinate of the environment in Z axis

VN-R (m)		V-H (m)	P-N	P-R (m)			I-N	C-R (m)	SI-N
x	y			x	y	z			
0.4	0.4	(max+min)/2	40	0.5	0.5	(max+min)/2	5000	0.25	5

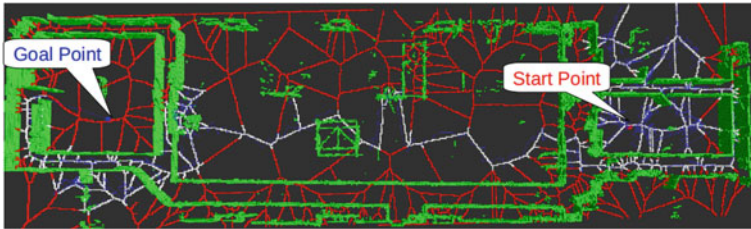


Fig. 3 Random tree growing process: the red lines are 2D voronoi diagram edges, the white lines are the edges traversed by path searching progress, the blue lines are the random trees constructed according to the white lines and local random particles

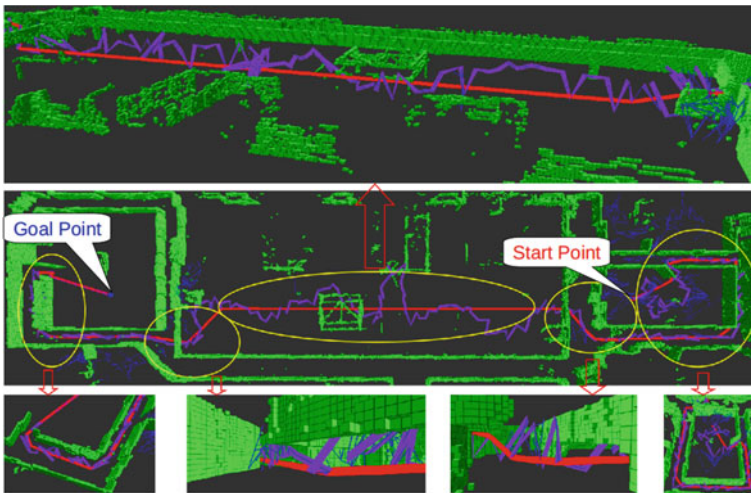


Fig. 4 2D voronoi based random trees Path Planning Algorithm in complicated 3D environments: At the beginning, a 2D voronoi diagram was created at the middle height of the complicated 3D map. Then, in order to pass through the 3D obstacles, a random trees was constructed by distributing random particles locally along the 2D voronoi edge nodes and searching among them to get the best particle for the trees growth toward the target point. By doing so, the random trees was constructed and represented with the blue lines. The purple line is the path queried from random trees, and the red line is the final shortened and smoothed path

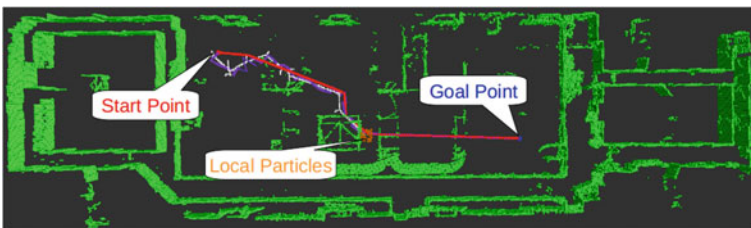


Fig. 5 Efficiency examination in middle spacious space: red point is the start point, the white lines are the edges traversed for path searching, the blue point is target point, the purple line is the path queried from random trees, the red line is the final shortened and smoothed path

Table 2 The average result of 100 times of efficiency examinations: *iterations* records the average iterations used for the algorithm to find out a valid path, *nodes* record the average numbers of random trees nodes when procedure stopped, *time* is the average time for running this algorithm one time and *success* records the state whether a valid path is found for each repetition

Algorithm	Start position	Target position	Iterations	Nodes	Time/s	Success
VB-RT	Right room	Left room	1,003	1,003	72.7	91/100
RRT			5,000	316	5.2	0/100
VB-RT	Middle spacious space	Middle spacious space	51	51	4.0	100/100
RRT			2,071	790	4.3	100/100

shown in Fig. 3, where we can see that the random trees can not only pass through narrow passages but also get out of and get into rooms. Even though our algorithm consumed a little more time than basic RRT, VB-RT algorithm can find a path from one room to another room with probabilistic of 91 % while basic RRT is 0 within 5,000 iterations.

For validating the efficiency of VB-RT algorithm, both the start and target points are set in the middle spacious place as shown in Fig. 5. And in Table 2, the last two rows are the average testing results of VB-RT and basic RRT algorithms, from which we can see that both VB-RT and basic RRT can always find a valid path very fast in this kinds of uncluttered 3D environments. And our algorithm is a little faster than basic RRT, which proves the efficiency of our algorithm. As to the *iterations* and *nodes*, nearly 40 times of the number of iterations and 15 times of the number of nodes are needed by basic RRT than VB-RT algorithms which means that the VB-RT algorithm consume fewer computing resources and storage resources. It should be noted that current implemented code of the proposed algorithm has not been optimized, we think if we use some advanced data-structure, the speed could be improved dramatically.

4.3 Robustness Demonstration

For robustness, we tested our VB-RT algorithm in various kinds of environments. And all of those environments are constructed by a 3D reconstruction equipment in real environments which have rooms, corridors, tables and so on. The experimental results are shown in Fig. 6, in which the white lines are voronoi edges traversed for path searching, the blue lines are the random trees, the yellow particles are the local distributed points, the purple line is the path queried from random trees and the red line is the final shortened and smoothed path. To test the performance of the proposed method in normal indoor and corridor environments, we carried out two tests in the

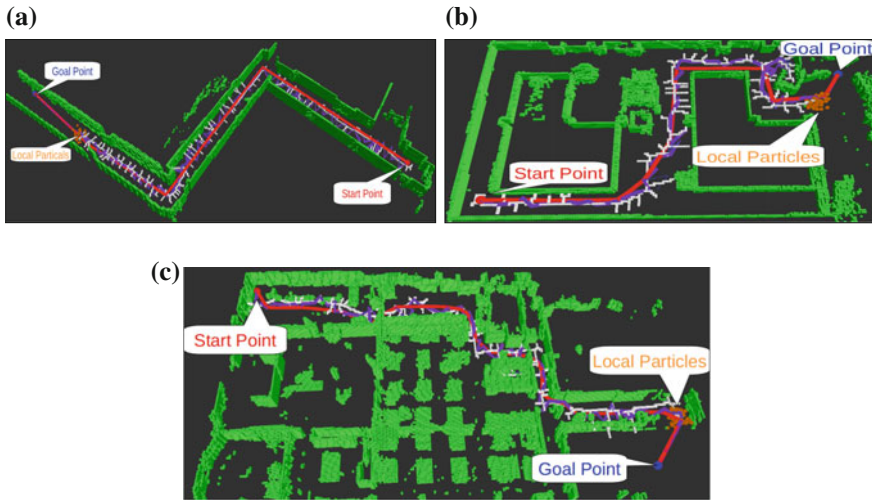


Fig. 6 Robustness validation in complicated indoor environments: **a**, **b** are simple 2.5D environments, **c** is a restaurant which is an extremely complicated 3D environment with many rooms, corridors, tables. In these figures, the *white line* is the voronoi edges traversed for path searching, the *blue lines* are the random trees, the yellow particles are the local distributed points, the *purple line* is the path queried from random trees and the *red line* is the final shortened and smoothed path

cluttered environments. The testing results are shown in Fig. 6a, b. From the figures, we can see that the environments are relatively simple 3D environments, in which the random trees grew toward target quickly without too many redundant nodes. That is because two cost functions are used to select the next best voronoi edge node and particle respectively to help to grow the random trees. As a result, these two tests show that VB-RT algorithm perform well in simple 3D environments.

To test the performance of the proposed method in complicated 3D indoor environment, we carried out an experiment in a cluttered environment. The experimental result is shown in Fig. 6c. As you can see, the test environment is an extremely complicated indoor environment which contains many rooms, corridors, tables and other obstacles. And searching in this kinds of environments have been proved to be the most difficult problem for sampling based path planning algorithms. But, we can see that our algorithm can still find a valid path easily without too many redundant trees nodes. That is because 2D voronoi diagram can always guide the random trees to get out of rooms and pass through corridors easily. And the good performance of VB-RT algorithm in this kinds of 3D environment not only proves the robustness, but also the efficiency.

At the same time, we also tested the efficiency of the proposed algorithm in this test. The results are listed in Table 3, where we repeated our algorithm 100 times in each kind of 3D environment. From Table 3, we can conclude that the proposed algorithm can still keep its efficiency and completeness in various kinds of environments.

Table 3 The average result of 100 times of robustness examinations

Test environment	Start position	Target position	Iterations	Nodes	Time/s	Success
Figure 6a	Right	Left	157	157	0.715	100/100
Figure 6b	Bottom left corner	Top right corner	67	67	0.189	100/100
Figure 6c	Top left corner	Bottom left corner	157	157	7.149	94/100

5 Conclusion

This paper presents an improved sampling-based path planning algorithm for complicated 3D environments, which uses 2D voronoi diagram to guide the growth of the 3D random trees. This algorithm was examined and compared with RRT algorithm in various kinds of practical environments to show its efficiency and robustness. From the experimental results, it is obviously that the proposed algorithm is very efficient to grow a random trees in various 3D indoor environments and find a valid path with a high possibility. Meanwhile, our algorithm can not only find a valid path in 3D environments with long narrow passages, but also work in very complicated 3D environments with cluttered obstacles and corridors with narrow entrances. And in the future, the random particle selecting technique and collision checking method will be improved to further enhance the performance of the proposed method.

Acknowledgements This work was supported by Natural Science Foundation of China under Grant Number 61573091 and 61300020.

References

1. Latombe, J.-C.: Motion planning: a journey of robots, molecules, digital actors, and other artifacts. *Int. J. Robot. Res.* **18**(11), 1119–1128 (1999)
2. Fadzli, S.A., Abdulkadir, S.I., Makhtar, M., Jamal, A.A.: Robotic indoor path planning using Dijkstra's algorithm with multi-layer dictionaries, pp. 1–4 (2015)
3. Eraghi, N.O., Lopez-Colino, F., De Castro, A., Garrido, J.: Path length comparison in grid maps of planning algorithms: HCTNav, A and Dijkstra. In: *Proceedings of the 2014 29th Conference on Design of Circuits and Integrated Systems, DCIS 2014* (2015)
4. Daniel, K., Nash, A., Koenig, S., Felner, A.: Theta*: any-angle path planning on grids. *J. Artif. Intell. Res.* **39**, 533–579 (2010)
5. LaValle, S.M.: *Planning Algorithms*, vol. 842. Cambridge University Press (2006)
6. Kavraki, L., Kavraki, L., Svestka, P., Svestka, P., Latombe, J.-C., Latombe, J.-C., Overmars, M., Overmars, M.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **12**(4), 566–580 (1996)
7. LaValle, S.: Rapidly-exploring random trees: a new tool for path planning. TR 98–11. Iowa State University, Computer Science Department (1998)

8. Janson, L., Schmerling, E., Clark, A., Pavone, M.: Fast marching tree: a fast marching sampling-based method for optimal motion planning in many dimensions, pp. 1–60 (2013)
9. Lau, B.: Improved updating of Euclidean distance maps and Voronoi diagrams. In: *Intelligent Robots and Systems*, pp. 281–286 (2010)
10. Dobson, A., Bekris, K.E.: Improving sparse roadmap spanners, pp. 4091–4096 (2013)
11. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **30**(7), 846–894 (2011)
12. Ekenna, C., Uwacu, D., Thomas, S., Amato, N.M.: Improved roadmap connection via local learning for sampling based planners. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3227–3234 (2015)
13. Salzman, O., Halperin, D.: Asymptotically near-optimal RRT for fast, high-quality, motion planning. *CoRR* **255827**(255827), 4680–4685 (2013)
14. Foskey, M., Garber, M., Lin, M., Manocha, D.: A Voronoi-based hybrid motion planner. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 1, pp. 55–60 (2001)
15. Purcaru, C., Precup, R.E., Iercan, D., Fedorovici, L.O., David, R.C.: Hybrid PSO-GSA robot path planning algorithm in static environments with danger zones. In: *2013 17th International Conference on System Theory, Control and Computing, ICSTCC 2013; Joint Conference of SINTES 2013, SACCS 2013, SIMSIS 2013 - Proceedings*, pp. 434–439 (2013)
16. Yao, P., Wang, H., Su, Z.: Hybrid UAV path planning based on interfered fluid dynamical system and improved RRT, pp. 829–834 (2015)
17. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap : an efficient probabilistic 3D mapping framework based on octrees. *Auton. Robots* 189–206 (2013)

3D FieldLut Algorithm Based Indoor Localization for Planar Mobile Robots Using Kinect

Xiaoxiao Zhu, Qixin Cao and Wenshan Wang

Abstract The FieldLut algorithm is a widely used localization algorithm in the RoboCup MSL (Middle Size League). It is now used for indoor mobile robot localization, but it can only use 2D range data. This paper improves the FieldLut algorithm to allow the use of 3D range data for indoor localization and uses Kinect sensor as the input sensor. The core of our improvement is the creation of a 3D LUT (lookup table). The 3D LUT is created as a multi-layer 2D LUT. Additionally, a memory optimization method is proposed. Experimental result shows real-time performance at video rates and high accuracy; for example, using Kinect sensor, the localization error is below 15 cm in a 13×8 m room and the repeat localization is below 6 cm.

Keywords FieldLut algorithm · Kinect sensor · Indoor localization

1 Introduction

FieldLut [1] is a successful localization method developed in the RoboCup MSL field [2]. Although this algorithm was developed to use an omni-directional camera sensor, it uses a camera as a 2D range sensor. Therefore, FieldLut is a 2D localization algorithm using a 2D range sensor.

FieldLut has recently been used for indoor environment localization with an infrared range sensor and a laser scan finder [3]. In their experiment, they treated the

X. Zhu (✉) · Q. Cao (✉) · W. Wang (✉)
State Key Lab of Mechanical Systems and Vibration,
Research Institute of Robotics, Shanghai Jiao Tong University,
Shanghai 200240, China
e-mail: ttl@sjtu.edu.cn

Q. Cao
e-mail: qxcao@sjtu.edu.cn

W. Wang
e-mail: Amigo@sjtu.edu.cn

wall as the white line in the soccer field in RoboCup. The result shows good accuracy and robustness. They also improved the global localization method of the original algorithm.

The release of the Kinect sensor [4] at the end of 2010 attracted considerable attention in many domains. Many researchers treat the Kinect as a RGB-D (color and depth) sensor and have developed some localization algorithms such as RGBDSLAM [5, 6] and RGB-D ICP based localization [7]. Depth and color information can be achieved at the same time, which makes the algorithm simpler and more robust than the traditional RGB sensor. However, those algorithms are still somewhat influenced by light conditions. Others use the Kinect as a 3D range sensor. The Kinect Fusion [8] algorithm uses plane-to-plane ICP for localization, which is a map based method and is only suitable for relative small maps because the map is stored in the display memory. A plane filter is performed and those plane points are then matched to a 2D map to get localization information.

The work most relative to ours is that of [9], who was the champion team of RoboCup 2007. They treat the Kinect as 2D range sensor by selecting the fixed high points from the input point cloud and then directly applying the FieldLut algorithm. Their experiment shows good accuracy and robustness, but the robustness of their algorithm is lower than using an omni-directional camera or laser range sensor because of the limited Horizontal view of Kinect. The fixed height is at the ceiling level, so it is always far from the robot and the input data will contain much noise, which decreases the accuracy of the localization.

In this paper, we aim to improve the method proposed in [9]. The main idea is to use the vertical range to compensate for the limitation in the horizontal comparison to the laser range finder. A 3D FieldLut algorithm is proposed that can use the 3D point cloud from Kinect for 2D indoor mobile robot localization. In the following sections, the FieldLut algorithm is first briefly introduced and then our improved 3D FieldLut algorithm is presented, followed by the experiment and conclusion.

2 FieldLut Algorithm

2.1 Basic Principle of FieldLut Algorithm

In the RoboCup MSL, all soccer robots need to self-localize. In 2005, Tribots (Champions of the RoboCup 2006–2007) developed the FieldLut algorithm, which based on matching the white line points captured by the visual system (usually using an omni-directional camera). The FieldLut algorithm involves two coordinate systems: the field coordinates system (FCS) and the robot coordinates system (RCS). The idea of the FieldLut is that those white points (as shown in Fig. 1a) can be transformed from the RCS to the FCS if the exact robot pose in the FCS is known and those white points will then move to the white line on the field, as

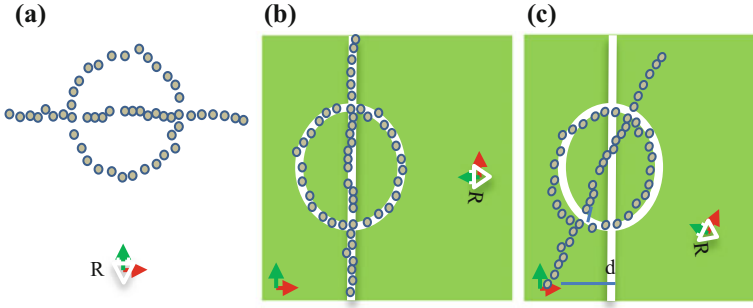


Fig. 1 a White line points in RCS. b, c The transformed white line points are coincide (or not coincide) with the white line on the field, given a correct (or wrong) pose of robot

shown in Fig. 1b. Otherwise, if the robot pose is wrong, not all the prospective white points will coincide with the line, as shown in Fig. 1c.

FieldLut defines this inconsistent as a robot pose error function for every white line point, as in (1). The minimum total error E value in (2) will give an optimized robot pose.

$$err = 1 - \frac{c^2}{c^2 + d(p + \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} s_i)^2} \quad (1)$$

$$\underset{p, \varphi}{\text{minimize}} E = \sum_{i=1}^n err \left(d \left(p + \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} s_i \right) \right) \quad (2)$$

where the $d(\cdot)$ computes the distance from a certain point on the field to the closest white line mark, c is a constant reducing the influence of faraway points, p and φ are the robot pose and the orientation, and s_i is the position of the white point in the RCS.

The minimization task of (2) is solved by RPROP algorithm [10]. Due to the quick convergence and high robustness of RPROP, FieldLut uses 10 iterations of RPROP to solve the minimization task [1]. RPROP uses only the sign of the derivation of the variables to determine the direction and step length of the next search, so its implementation is very simple [11]. The FieldLut workflow is as follows:

- (a) Use the motion model and odometer information to compute a robot prediction pose.
- (b) Use the RPROP method to compute the optimized pose, starting from the prediction pose.
- (c) Repeat step (a) to step (b) 10 times for every frame.

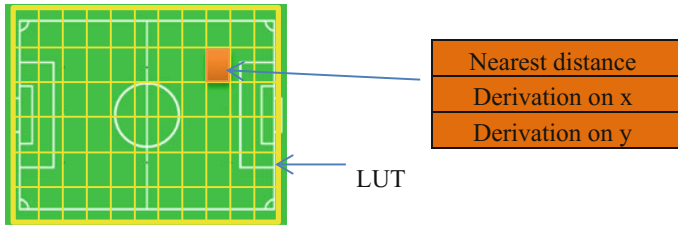


Fig. 2 The LUT for the game field

Notice that RPROP method needs to compute the $d(\cdot)$ and the derivations, which are time consuming. To save computing time, FieldLut defines a LUT for the game field; one cell of the LUT stores three values: the nearest distance and two derivations at the center of the cell, shown in Fig. 2. This is the origin of the name.

2.2 Relation to the ICP (Iterative Closest Point) Algorithm

The general ICP [12] method also iteratively finds the nearest points and then computes the optimized transform, but the two methods are different in two areas. On one hand, they use different error functions, and so their optimization methods are different. On the other hand, the FieldLut using a LUT is faster because it does not need to compute the nearest points every time, although this comes at the cost of higher memory requirements.

3 Improved 3D FieldLut

To use 3D range data as input, the error function (1) was changed to (3), in which $d(\cdot)$ and s_i change from 2D to 3D means that a 3D LUT needs to be built. The creation of the 3D LUT is the kernel of our improvement, but unlike the football field, we usually have no knowledge of the geometry of the room. Therefore, we first build a 3D point cloud model of the room as Fig. 3 and then build a 3D LUT based the 3D model.

$$err = 1 - \frac{c^2}{c^2 + d\left(\begin{matrix} p_x + s_{ix} \cos \varphi - s_{iy} \sin \varphi \\ p_y + s_{ix} \sin \varphi + s_{iy} \cos \varphi \\ s_{iz} \end{matrix}\right)_i^2} \tag{3}$$

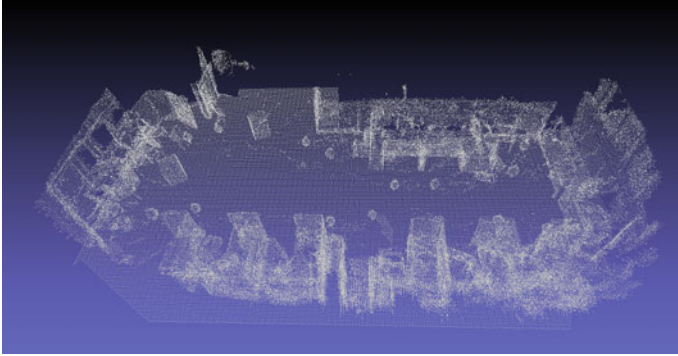


Fig. 3 A point cloud map

3.1 3D LUT Building

In 2D, every cell of the LUT stores the nearest distance to the environment points. It can be seen from the distance mapping function ($d(\cdot)$ in (3)) that the nearest cell distance in 3D does not represent the distance to a global nearest point, but the nearest point with the same cell height, so a 3D LUT is in fact equal to a multi-layer 2D LUT. The building steps are as follows:

- (a) Build a volume which is a little bigger than the bounding box of the model point cloud, such as the cyan box in Fig. 4a.
- (b) Divide all the points of the point cloud of the room model into different slices based on their z component. This yields one slice, as shown in Fig. 4a, b.
- (c) Build a grid for every slice and look for the nearest distance of each slice to every cell on the grid. Kd-tree based nearest point searching is used to accelerate this process. There is the problem of having to decide on the volume resolution. High resolution leads to high accuracy, but according to [13], it makes no sense to make the resolution of the grid higher than the point cloud's. The LUT for one slice is displayed in gray-level image as Fig. 4c, where the darker the smaller the nearest distance value.
- (d) Compute the derivation. This is discussed in Sect. 3.4.

3.2 Derivations

The derivations of E in (2) can be easily computed using a chain rule, as in (4).

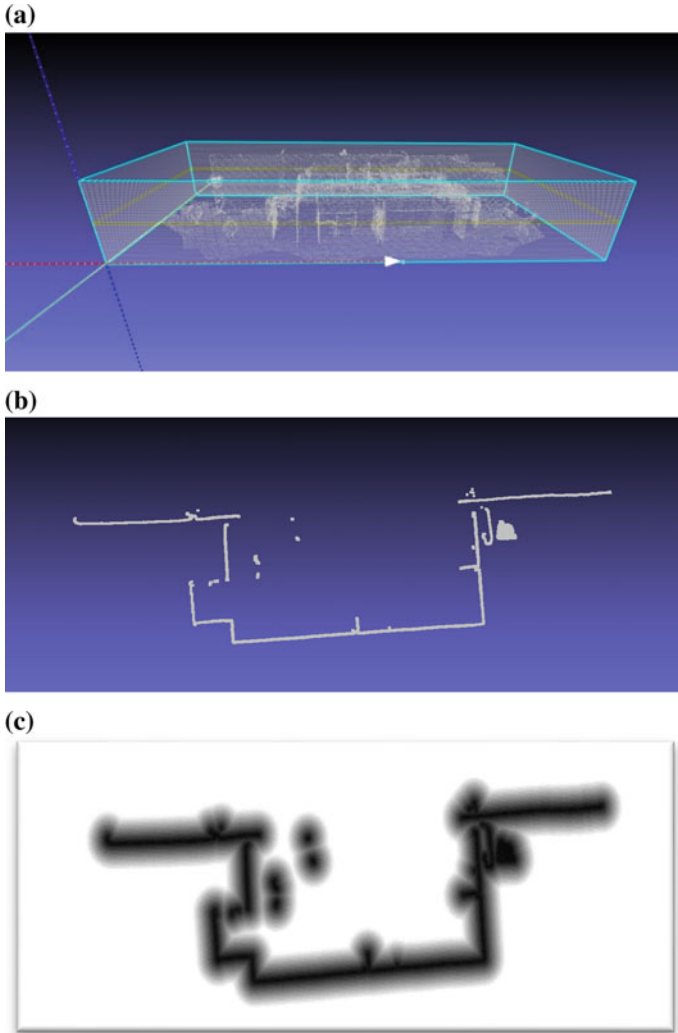


Fig. 4 a Volume of the 3D LUT and the slices. b Points and grid in one slice. c Gray-level display of the LUT of one slice

$$\begin{cases} \frac{\partial E}{\partial p_x} = \sum_{i=1}^n \frac{2c^2}{(c^2 + d^2)^2} \frac{\partial d}{\partial u} \\ \frac{\partial E}{\partial p_x} = \sum_{i=1}^n \frac{2c^2}{(c^2 + d^2)^2} \frac{\partial d}{\partial v} \\ \frac{\partial E}{\partial \varphi} = \sum_{i=1}^n \frac{2c^2}{(c^2 + d^2)^2} \left[\frac{\partial d}{\partial u} (-s_{ix} \sin \varphi - s_{iy} \cos \varphi) + \frac{\partial d}{\partial v} (s_{ix} \cos \varphi - s_{iy} \sin \varphi) \right] \end{cases} \quad (4)$$

where $u = p_x + s_{ix} \cos \varphi - s_{iy} \sin \varphi$, $v = p_y + s_{ix} \sin \varphi + s_{iy} \cos \varphi$.

Fig. 5 Neighborhood of the current cell d_5

d_1	d_2	d_3
d_4	d_5	d_6
d_7	d_8	d_9

$\frac{\partial d}{\partial u}$ and $\frac{\partial d}{\partial v}$ are computed with the Sobel kernel. Assume the 3×3 neighborhood in the distance LUT at the $z = s_{iz}$ slice is shown in Fig. 5.

Using the Sobel kernel, we have

$$\begin{cases} \frac{\partial d}{\partial u} = 0.125 \times (-d_1 + d_3 - 2d_4 + 2d_6 - d_7 + d_9) / \text{grid_size_u} \\ \frac{\partial d}{\partial v} = 0.125 \times (-d_1 - 2d_2 - d_3 + d_7 + 2d_8 + d_9) / \text{grid_size_v} \end{cases} \quad (5)$$

where grid_size_u , grid_size_v means the size of one cell.

3.3 Memory Optimization

In the original algorithm, the single cell in the LUT uses the float type to represent the nearest distance, d , and two derivations, so one LUT cell requires 12 bytes. In 3D LUT, the cost of memory increases. For example, a $1024 \times 1024 \times 300$ LUT requires 3.6 GB of memory. To save memory, the nearest distance is represented by one byte in mm, so the maximum distance that can be represented is 255 mm. This limitation has the same bounding effect as the constant value, c , in the error function (3), which is set as 250 mm in FieldLut [1]. However, the original function (3) is still used to normalize the error. The RPROP optimization algorithm just needs the sign of the derivations ($\frac{\partial E}{\partial p_x}$, $\frac{\partial E}{\partial p_y}$ and $\frac{\partial E}{\partial p}$ in (4)) and not the exact value, so if $\text{grid_size_u} = \text{grid_size_v}$, Eq. (5) can be modified to (6). The varying distances between neighboring cells are not greater than the cell size (grid_size_u or grid_size_v), which is 1–10 mm according to the resolution of the LUT; therefore, 1 byte can be used to represent each derivation. After optimization, one LUT cell requires just 3 bytes.

$$\begin{cases} \frac{\partial d}{\partial u} = -d_1 + d_3 - 2d_4 + 2d_6 - d_7 + d_9 \\ \frac{\partial d}{\partial v} = -d_1 - 2d_2 - d_3 + d_7 + 2d_8 + d_9 \end{cases} \quad (6)$$

3.4 External Parameters Calibration of the Kinect

In the previous section, it was assumed that the input point position, S_i , is in the RCS. In fact, the original input points are in the Kinect coordinate system (KCS).

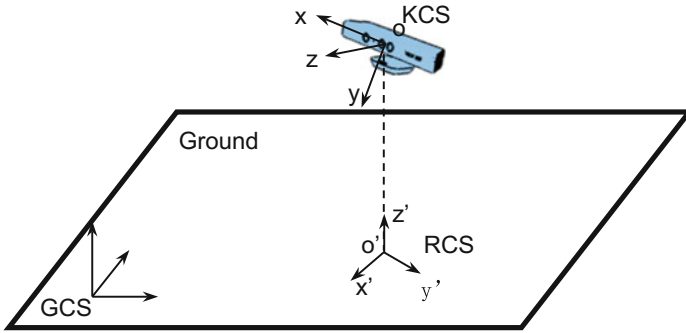


Fig. 6 Definition of RCS

Therefore, a transformation from the KCS to RCS needs to be determined. The RCS can be defined as the coordinate $o' - x'y'z'$ in Fig. 6. The o' is the projection of the KCS origin in the ground plane and the projected z axis is the new x' axis. The z' axis is parallel to the normal direction of the ground plane. Assuming the plane equation of the ground in KCS is $\hat{n}x = -p, p \geq 0$, the transform equation is

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ p \end{pmatrix} + \begin{pmatrix} \hat{I}_x & \hat{I}_y & \hat{I}_z \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{7}$$

where \hat{I}_* and \hat{I}_* , $*$ = x, y, z is the unit vector of the axis. $\hat{I}_z = \hat{n}, \hat{I}_y = \hat{I}_z \times \hat{I}_z, \hat{I}_x = \hat{I}_y \times \hat{I}_z$.

4 Experiment

A soccer robot with a Kinect mounting on the top was used in the experiment, as shown in Fig. 7. The 3D FieldLut algorithm was implemented in C++ under Linux and ran on a single core 2.5 GHz CPU. The size of the LUT was $13 \times 8 \times 3$ m and the grid resolution was $1300 \times 800 \times 300$. To achieve real time performance, the input points were sampled and 64×48 points were used for localization. The process time for every frame was 18 ms.

The first experiment is designed to test the repeat accuracy. We set a fix trajectory and control the robot to go through it for 6 times. The ground truth of the trajectory of the robot is measure by a Hokuyo URG laser sensor. The result is shown in Fig. 8. The repeat localization error is below 6 cm.

The second experiment is to test the localization accuracy. We compare the localization result of the 3D FieldLut algorithm with the laser sensor's. The comparison results are shown in Fig. 9. The maximum error is about 15 cm, which is



Fig. 7 Robot and the sensors used in the experiment

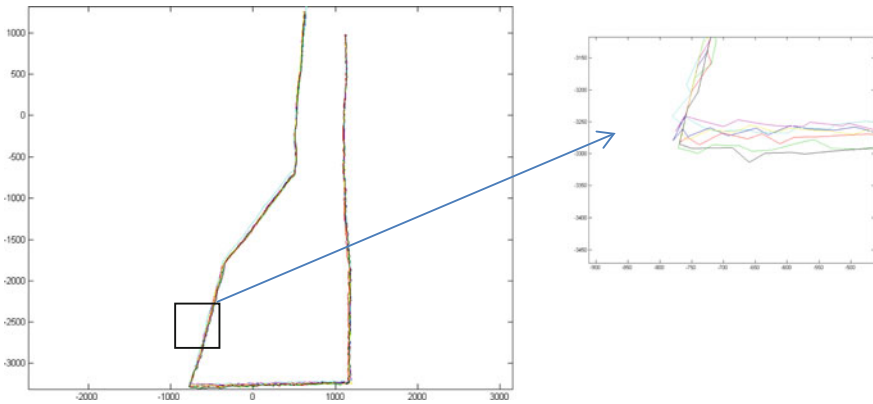


Fig. 8 The result of repeat localization accuracy test

generated mostly by the error of the map. The lack of smoothness of the trajectory can be tackled using the Kalman filter, as in [1].

Additionally the FPS of the algorithm is compared with the traditional look-up-table based Fast-ICP method [14]. As the result shown in Table 1, the 3D LUT algorithm is 10 times fast than the traditional Fast-ICP method.

Fig. 9 The result of localization accuracy test

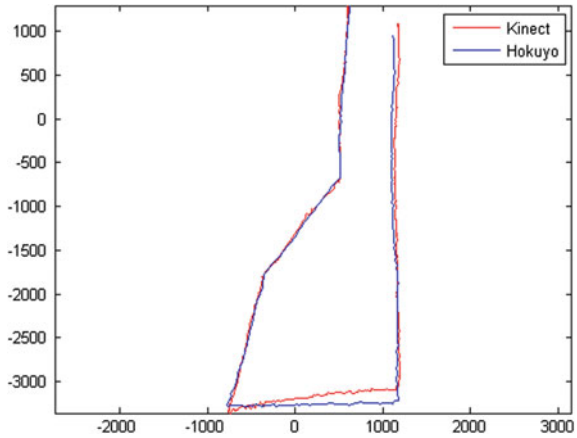


Table 1 The comparison result of FPS with the traditional Fast-ICP

	Number of points	FPS
Fast-ICP	3072	16.5
	25000	2.2
3DLUT	3072	170
	25000	20

5 Conclusion

In this paper, an improved 3D FieldLut algorithm that can make use of 3D range data from Kinect was proposed. A KinectFusion based sub-map mapping method for building a 3D indoor environment model was presented, with emphasis on the consistency of the ground plane. The experimental results show that the accuracy the final 3D map is less than that in a room with dimensions of 13×8 m. A 3D LUT was then built and the memory optimization method was proposed. The localization experiment shows that the localization error is below 15 cm, which should be sufficient for most indoor tasks. However, all the experiments were conducted in perfect static environment, so many other problems need to be solved for real-world use. One of these is the processing of moving objects, such as people. We already have some ideas for dealing with moving objects, such as the use of people detection algorithms to detect people and delete the pixels of people before performing localization. Another issue is that the FieldLut algorithm is just a local localization method requiring an initial starting position, so the global localization problem must also be considered. Global localization is discussed in the original FieldLut algorithm and we will perform experiments on them in future work.

Acknowledgements We thank the support of China Postdoctoral Science Foundation, No. 2015M571561 and the National Natural Science Foundation of China, No. 61273331.

References

1. Lauer, M., Lange, S., Riedmiller, M.: Calculating the perfect match: an efficient and accurate approach for robot self-localization. *Lecture Notes Comput. Sci. J.* **4020**(2006), 142–153 (2005)
2. wiki. Middle Size League of RoboCup. http://wiki.robocup.org/wiki/Middle_Size_League (3.21)
3. Gouveia, M., Moreira, A. P., Costa, P. et al.: Robustness and precision analysis in map-matching based mobile robot self-localization. In: *Fourteenth Portuguese Conference on Artificial Intelligence*, pp. 243–253 (2009)
4. Wiki. Kinect. <http://en.wikipedia.org/wiki/Kinect> (4.26)
5. Engelhard, N., Endres, F., Hess, J. et al.: Real-time 3D visual SLAM with a hand-held RGB-D camera. In: *RSS 2010 Workshop on RGB-D Cameras* (2010)
6. Henry, P., Krainin, M., Herbst, E. et al.: (2010) RGB-D mapping: using depth cameras for dense 3D modeling of indoor environments. In: *RSS 2010 Workshop on RGB-D Cameras*
7. Fioraio, N., Konolige, K.: Realtime visual and point cloud SLAM. In: *RSS 2010 Workshop on RGB-D Cameras* (2011)
8. Izadi, S., Newcombe, R. A., Kim, D. et al.: Kinectfusion: real-time dynamic 3d surface reconstruction and interaction. In: *ACM SIGGRAPH 2011* (2011)
9. Cunha, J., Pedrosa, E., Cruz, C. et al.: Using a depth camera for indoor robot localization and navigation. In: *RSS 2010 Workshop on RGB-D Cameras* (2011)
10. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: *IEEE International Conference on Neural Networks*, pp. 586–591 (1993)
11. Riedmiller, M., Braun, H.: Rprop-description and implementation details. Citeseer (1994)
12. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 239–256 (1992)
13. Yan, P., Bowyer, K.W.: A fast algorithm for ICP-based 3D shape biometrics. *Comput. Vis. Image Underst. J.* **107**(3), 195–202 (2007)
14. Tubic, D., Hébert, P., laurendeau, D.: A volumetric approach for interactive 3D modeling[A]. In: *the 1st International Symposium on 3D Data Processing, Visualisation and Transmission (3DPVT02)[C]*, Padova, pp. 150–158 (2002)

A Feature-Based Mutual Information and Wavelet Method for Image Fusion

Yulong Liu, Yiping Chen, Cheng Wang and Ming Cheng

Abstract Accurate image fusion is an essential technique to obtain more information from remote sensing image in different sensors. This paper presents a method for fusion delineating objects from multiple sensors. The proposed algorithm partitions feature-based mutual information into the maximization as the requirement for fusion, which consists of entropy in the image. The wavelet transform decomposes the maximum value of the mutual information for image fusion. To evaluate the validity of the proposed method, experiments were conducted using two types of remote sensing images. The overlapping, correctness, and quality of the fusion object are over 98 %, 95.3 %, and 95.1 % respectively, which proves the proposed method is a promising solution for registration and fusion from two remote sensing images.

Keywords Image registration · Image fusion · Mutual information · Entropy · Wavelet transform · Multiple sensors

1 Introduction

Increasing rates of information have led to a growing demand for image fusion. Normally, we collect data at the same time and same position in difficulty. Image registration is an image processing to find the matrix to pair two or more images from different sensors, different view position or different times. Efficient techniques for data fusion in the field of remote sensing have been gained popularity in aligning coastline or road and the object recognition. Many studies have developed

Y. Liu · Y. Chen (✉) · C. Wang · M. Cheng
Fujian Key Laboratory of Sensing and Computing for Smart Cities,
School of Information Science and Engineering, Xiamen University,
Xiamen FJ 361005, China
e-mail: ypchenhk@gmail.com

Y. Liu
e-mail: ylliuXu@163.com

to information fusion from remote sensing images, these methods are mutual information [1], pixel migration [2], implicit similarity [3], feature consensus [4] and contour-based [5]. Considering the image features from different sensors have low similarity, an intensity-based image registration and the mutual information registration from information theory have been introduced. Applied to rigid registration of multiple sensors, it shows great promise like less limitation, high robust and high accuracy. Suri and Reinartz [6] proposed an improved mutual information method of TerraSAR-X and IKONOS images based registration. Guo and Lu [7] takes advantage of mutual information based on gradient vector flow to do multi-modality image registration, and [8] combined SIFT and mutual information to describe Coarse-to-Fine image matching algorithm, etc. We improved the mutual information method by calculating the maximal gradient direction of images to estimate the angular point as the coarse registration.

The relative area of images for fusion can be gained by acquired transformation matrix according to transform, scale, rotate the image in image registration. Hence, Image fusion is a process of combining information precisely from one image to another image to extent the content, which can extract information from each channel to improve image quality. In recent years, lots of articles on image fusion were published [9–13]. In general, image fusion from low to high level can be divided into three categories: pixel level fusion, feature level fusion, and decision level fusion [14]. The wavelet transform is a local transformation of space (time) and frequency, which can effectively extract information from the signal. And wavelet transform can be divided into two categories: Continuous Wavelet Transform (CWT) and Discrete Wavelet Transform (DWT). CWT is used in theoretical analysis because it is performed in an ideal condition. Nevertheless, DWT is applied to practice mainly. The perfect reconstruction ability to ensure that there is no information loss and redundant information in the process of decomposition, and easy to extract the structure information and details of the original image. Wavelet analysis provides a selective image in the direction of the human visual system. As pixel level fusion, wavelet transform is a popular method in fusion domain. Li et al. [15] had made a summary about multi-sensor image fusion by using wavelet transform. Recently, [16] use wavelet fusion to enhance weather degraded video sequences, [17] proposed a method using wavelet fusion on ratio imaged for change detection in SAR images. El-Khamy et al. [18] used wavelet fusion in area of LMMSE image super-resolution. Multi-sensor image registration and fusion has attracted researchers' attention in areas of medical, remote sensing, computer vision, and military nowadays. The objective of this paper aims to develop a matching and fusion work for multiple images by maximizing mutual information and discrete wavelet transform (DWT) respectively. The rest of this paper is arranged as follows. Section 2 shows details of the algorithm proposed in this paper. Section 3 shows and discusses the experimental results. Conclusions are presented in Sect. 4.

2 Method

The image information of remote sensing usually contains a huge number of details with varied intensity and texture characteristic. The proposed method improved the fusion performance by increasing additional feature from maximal gradient direction value to mutual information method for image registration from multiple sensors data. The method mainly includes the following two steps: image registration using improved feature-based maximization of mutual information and image fusion by wavelet transform.

2.1 Feature-Based Mutual Information Registration

Shannon put forward the concept of mutual information in information theory. Image registration based on mutual information similarity measure is used as the criterion of image registration. The definition of mutual information is

$$MI(A, B) = H(A) + H(B) - H(A, B) \quad (1)$$

where $H(A)$ is the image A 's individual entropy; $H(B)$ is the image B 's individual entropy; $H(A, B)$ is the joint entropy of the images A and B and it means the information variable A contains random variable B . They can be further described as

$$H(A) = - \sum_a p_A(a) \log p_A(a) \quad (2)$$

$$H(B) = - \sum_b p_B(b) \log p_B(b) \quad (3)$$

$$H(A, B) = - \sum_{a,b} p_{AB}(a, b) \log p_{AB}(a, b) \quad (4)$$

where $p_A(a)$ and $p_B(b)$ represent the marginal probability of image A and B , and $p_{AB}(a, b)$ represents the joint marginal probability of A and B . $p_A(a)$ and $p_B(b)$ are calculated by the simple marginal histograms of their images. $p_{AB}(a, b)$ is obtained by the joint marginal histograms of their overlapping parts.

To reduce the time of calculating the mutual information, we choose the angular point of the building as the corresponding image points as the feature point first from two input images for coarse matching. We select the several approximating maximal gradient direction value to choose the angle point of building as feature point.

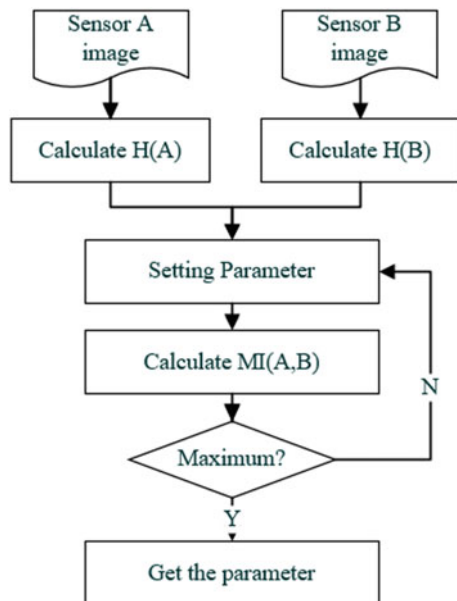
$$G(x, y) = dx(i, j) + dy(i, j) \tag{5}$$

where $G(x, y)$ is the image gradient of the x multiply y pixels, $dx(i, j)$ is the differential coefficient of the x direction and $dy(i, j)$ is the differential coefficient of the y direction. Mutual information represents the statistical dependence between two images and the mutual information of the corresponding map is the largest when the two images are matching. Figure 1 is the flowchart of image registration in this paper and in Fig. 1 the parameter includes angel of rotation, scaling factor and pixel offset. Figures 2a, b are both the view of airport. And we set Fig. 2a as the reference image and Fig. 2b as the float image. In order to find the point that can make the maximum of mutual information entropy of the two images, we hold the reference image and move the float image to match the reference image and calculate their mutual information. In this paper, we set the space of x -directions and y -directions are both from -20 to 20 pixels and each step is 1 pixel. Figure 2c shows the result of matching, it denotes that $(0, 0)$ is the peak point of two images. This proves we can find out the matching point from two multi-sensor images in an accuracy way for further step of selecting corresponding areas to perform image fusion.

2.2 Image Fusion

After image registration, the corresponding relationship between two images from different sources can be calculated. And if the images are simply superimposed on

Fig. 1 Flowchart of our algorithm



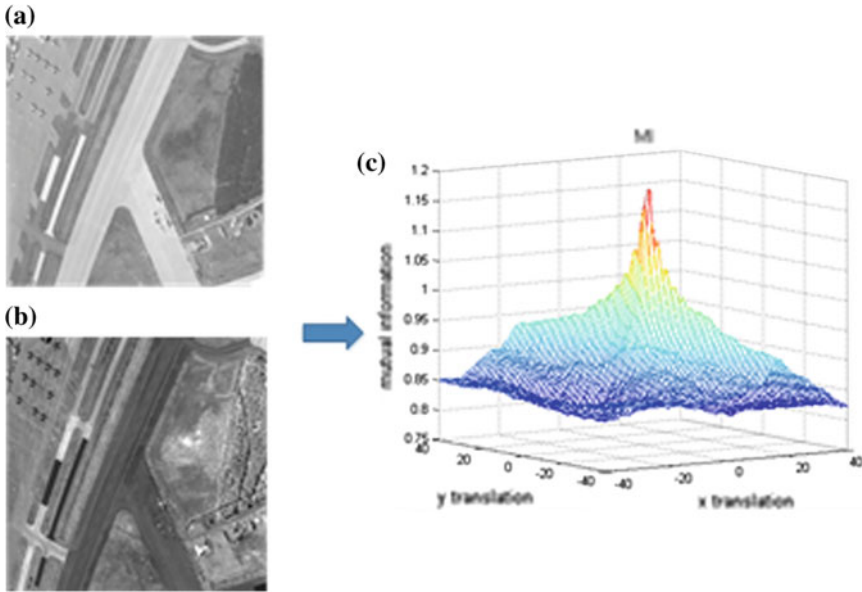


Fig. 2 a Infrared band image; b Visible band image; c Mutual information from (a) and (b)

this relationship, the structural information of two images cannot be show completely in composite image. It will lead to the lack of information and it is necessary to performance image fusion to keep this information. Wavelet transform can effectively decompose the signal and extract more information and it inherits and develops the thought of short time Fourier transform, it uses multi-scale analysis of signals by means of flexing and translating operation to achieve high frequency division by time, low frequency subdivision by frequency.

Firstly, if assumed function $\psi(t) \in L^2(R)$ satisfies $\int_{-\infty}^{\infty} \psi(t)dt = 0$, then the function $\psi(t)$ can be called basic wavelet. And the function cluster can be defined

$$\psi_{a,b}(t) = |a|^{-1/2} \psi\left(\frac{t-b}{a}\right) \tag{6}$$

Here, a and b are scale factor and shift factor respectively. And when signal $f(t)$ belongs to $L^2(R)$, Continuous Wavelet Transform (CWT) can be defined below:

$$W_f(a,b) = \int_{-\infty}^{\infty} f(t)\psi_{a,b}(t)dt = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t)\psi\left(\frac{t-a}{b}\right)dt \tag{7}$$

where a is a scale factor, b is a translation factor for controlling the time position. Parameter a controls the oscillation of wavelet. In general, CWT cannot be used

cause its limitations in reality. Hence, Discrete Wavelet and Discrete Wavelet Transform (DWT) are defined:

$$\Psi_{m,n} = a_0^{-\frac{m}{2}} \Psi(a_0^{-m}x - nb_0) \tag{8}$$

$$W_f(m,n) = \int_{-\infty}^{\infty} f(t)\Psi_{m,n}(t)dt \tag{9}$$

where $a = a_0^m, b = b_0^m$. According to the wavelet transform, the image is decomposed into four sub images which contain three high frequency bands and one low frequency band, while the three high frequency bands contain the details of the image in the horizontal, vertical and diagonal directions. If continue to decompose the low frequency components into next layer, we will get another four bands of three high and one low. Figure 3a is the original image. After the wavelet transform at a time, four images derive from original image. From Fig. 3b, top left corner image is the low-frequency part of original image that includes the most original image information and the other parts include the structure information. Level-2 wavelet decomposition result is generated from the Level-1 wavelet decomposition's low-frequency part and by this analogy we performance the Level-3 wavelet decomposition.

In this paper, we use Haar wavelet transform to decompose images. After wavelet decomposition of the image, fusing the high-frequency and low-frequency

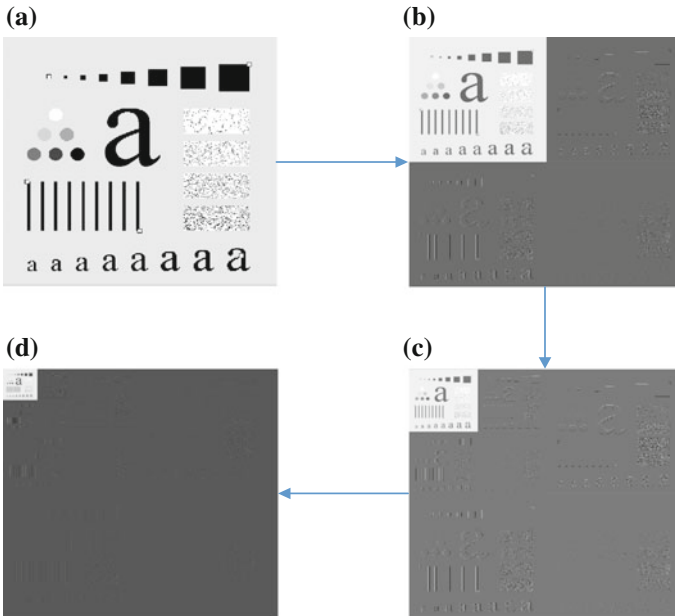


Fig. 3 a Original image, b–d are Level-1, Level-2 and Level-3 wavelet decomposition

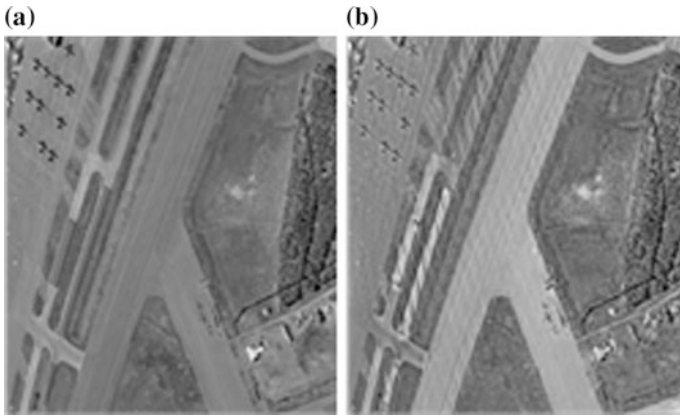
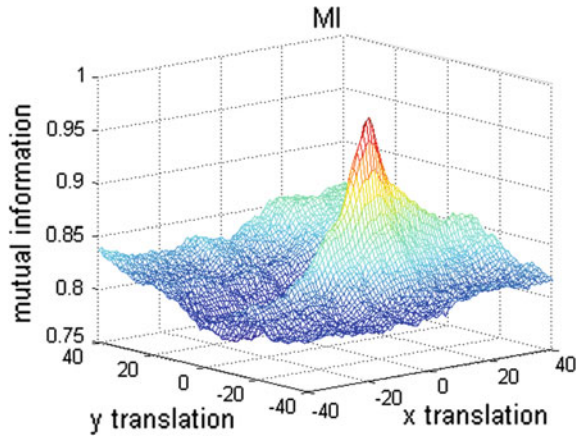


Fig. 4 a Superposition result and b Wavelet fusion result from Fig. 2

Fig. 5 Mutual information of Fig. 6a, d



components with weighted mean respectively. Eventually, we obtain the fusion image through the wavelet inverse transform of the high frequency and low frequency. Figure 4a assumed as image after registration and (b) is image after fusion. Compared with Fig. 5a, b, it is convenient to estimate the road in Fig. 4b, which is more clearly and image contrast is better.

3 Results and Discussion

In order to verify the performance of the algorithm, two images from different sensors were used to test our method. Figure 6a, b are images of the farm in visible image and infrared image respectively. The composition and intensity of two

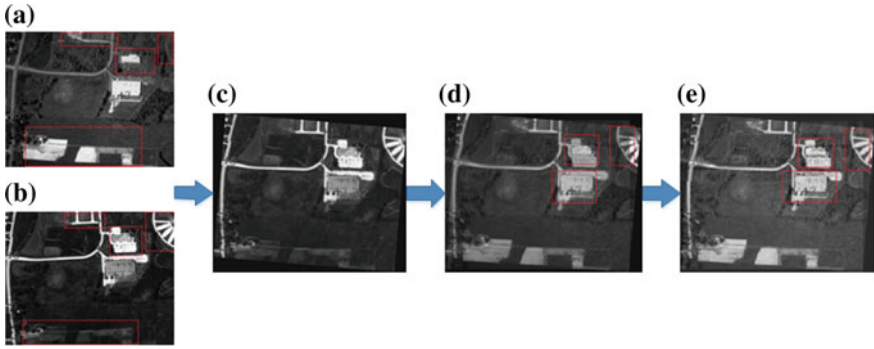


Fig. 6 **a** Visible image, **b** Infrared image, **c** Rotation and enlarging of **(b)**, **d** Image after registration, **e** Final image

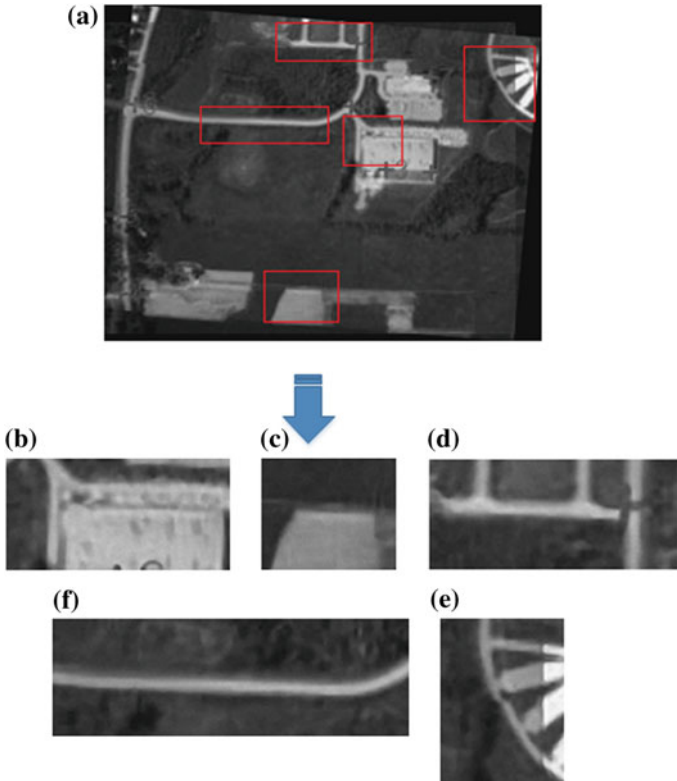


Fig. 7 **a** Registered image, and **b–e** are parts of **(a)**

images is different due to the different reflectance. This cause the objects in (a) are high value, and (b) has more structure information and better contrast. We exploit a red outline to mark features respectively before registration experiment. Finish the first step of registration; we get the scale factor and angle of rotation. Figure 6c is the result of (b). In Fig. 5, we can conclude the pixel offset of Fig. 6a, c and the result is (2, -11). To make sure the accuracy of registration, we cut some pieces from Fig. 7a and enlarge those areas. Figures 7b, c, e are the roof of buildings and (d), (f) are roads. It's obvious that the edges of the roof and roads are almost coincident. The result shows that Fig. 6a, b are well matching. After registration, we select the corresponding area from Fig. 6a, c to perform image fusion and Fig. 6d is the result of this experiment. According to the compare of Figs. 6d, e, the image under wavelet fusion has better contrast, especially in the areas marked by red box shows that wavelet fusion image has better structure information and continuity. Table 1 shows the comparison result, which means the performance costing is lower than original mutual information method. And the accuracy rate is higher according the pixel offset. In this experiment, it does not matter whether the images are from different resolution, cause in registration part we use bilinear interpolation to make sure two images from different sensors are under the same resolution. And to verify feasibility of this method, we did one more experiment. Figure 8a, b are visible image and infrared image and they have their own features responsibility like the part marked by the green outline. Figure 8c is the rotating and scaling after (b) and Fig. 8d is image weighting fusion. And compare to image (c), Fig. 8d has better structure information and contrast especially in comments section by green outline.

Table 1 Parameter comparison results of experiment

Parameter	Pixel offset	Overlapping (%)	Time
Our method result	(2, -11)	98	12S
Original mutual result	(1, -5)	97	20S

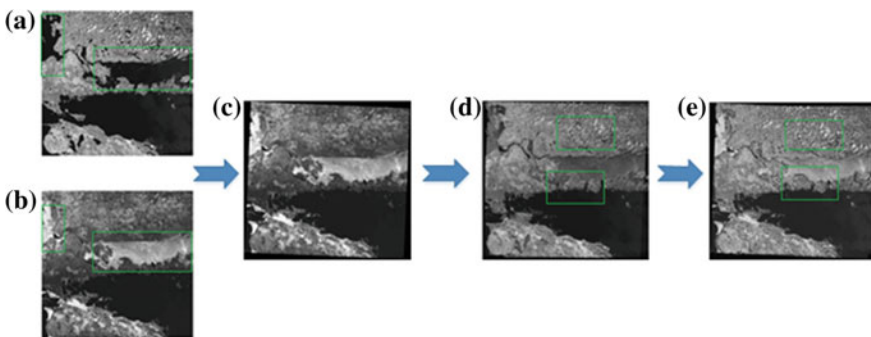


Fig. 8 a Visible image, b Infrared image, c Rotating and enlarging of (b), d Image after registration, e Final image

4 Conclusions

In this paper, we propose a feature-based mutual information method to image registration and fusion from two different sensors. The proposed method generates the model by conducting the maximal gradient direction of image to feature extraction for coarse registration, using the mutual information to precise matching. Eventually, an image fusion procedure presented based on wavelet transform. The research work aims to combine information of multiple relative images from different sensors. Experiments demonstrate that the proposed algorithm improves both time efficiencies and robustness of image fusion with overall accuracy of better than 95.1 %, especially the registration rate is 98 %.

References

1. Maes, F., Collignon, A., Vandemeulen, D., Marchal, G., Suetens, P.: Multimodality image registration by maximization of mutual information. *IEEE Trans. Med. Imaging* **16**(2), 187–198 (1997)
2. Yao, J.C., Goh, K.L.: A refined algorithm for multi-sensor image registration based on pixel migration. *IEEE Trans. Image Process.* **15**(7), 1839–1847 (2006)
3. Keller, Y., Averbuch, A.: Multi-sensor image registration via implicit similarity. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(5), 794–801 (2006)
4. Shekhar, C., Govindu, V., Chellappa, R.: Multi-sensor image registration by feature consensus. *Pattern Recogn.* **32**(98), 39–52 (1999)
5. Li, H., Manjunath, B.S., Mitra, S.K.: Contour-based multi-sensor image registration. *IEEE Trans. Image Process.* **4**(3), 320–334 (1995)
6. Suri, S., Reinartz, P.: Mutual-information-based registration of TerraSAR-X and Ikonos imagery in urban areas (2010)
7. Guo, Y.J., Lu, C.C.: Multi-modality image registration using mutual information based on gradient vector flow. In: *ICPR2006*, Hong Kong, 20–24 Aug, vol. 3, pp. 697–700 (2013)
8. Gong, M., Zhao, S., Jiao, L., Tian, D., Wang, S.: A novel coarse-to-fine scheme for automatic image registration based on SIFT and Mutual information. *IEEE Trans. Geosci. Remote Sens.* **52**(7), 4328–4338 (2014)
9. Wang, Z., Ya, Y., Gu, J.: Multi-focus image fusion using PCNN. *Pattern Recogn.* **43**(6), 2003–2016 (2010)
10. Yang, B., Li, S.T.: Pixel-level image fusion with simultaneous orthogonal matching pursuit. *Inf. Fusion* **13**(1), 10–19 (2012)
11. Miao, Q.G., Shi, C., Xu, P.F., Yang, M., Shi, Y.B.: A novel algorithm of image fusion using shearlets. *Optics Commun.* **284**(6), 1540–1547 (2011)
12. Ehlers, M., Klonus, S., Astrand, P.J., Rosso, P.: Multi-sensor image fusion for pansharpening in remote sensing. *Int. J. Image Data Fusion* **1**(1), 25–45 (2010)
13. Durga, R.V., Kumari, O., Prakash, M.S., Kumar, P.D., Tirupathi, Y.: Region-based image fusion using complex wavelets. *IOSR J. Electron. Commun. Eng.* **9**(1), 23–26 (2014)
14. Goshtasby, A.A., Nikolov, S.: Image fusion: advances in the state of the art. *Inf. Fusion* **8**(2), 114–118 (2007)
15. Li, H., Manjunath, B.S., Mitra, S.K.: Multisensor image fusion using the wavelet transform. In: *IEEE International Conference on Image Processing*, Austin, TX, 13–16 Nov, vol. 1, pp. 51–55 (1994)

16. John, J., Wilscy, M.: Enhancement of weather degraded video sequences using wavelet fusion. In: 7th IEEE International Conference on Cybernetic Intelligent Systems, London, UK, 9–10 Sept, pp. 1–6 (2008)
17. Ma, J.J., Gong, M., Zhou, Z.: Wavelet fusion on ratio images for change detection in SAR. *IEEE Geosci. Remote Sens. Lett.* **9**(6), 1122–1126 (2012)
18. El-Khamy, S.E., Hadhoud, M.M., Dessouky, M.I., Salam, B.M., El-Samie, F.E.A.: Wavelet fusion: a tool to break the limits on LMMSE image super-resolution. *Int. J. Wavelets Multi-resolution Inf. Process.* **4**(1), 105–118 (2011)

Part VIII
Measurement

An Intelligent RGB-D Video System for Bus Passenger Counting

Daniele Liciotti, Annalisa Cenci, Emanuele Frontoni,
Adriano Mancini and Primo Zingaretti

Abstract The information of the number of passengers getting in/off a vehicle is very important for public bus transport companies. In fact, the operators need to estimate the number of travellers using their vehicles for marketing purposes, for evaluating transit service capacities and allocating the proper number of buses for each connection-line. The goal of this work is to provide a system for counting and monitoring passengers, both adults and children, at the entrance of bus. This system is mainly based on an RGB-D sensor, located over each bus door, and image processing and understanding software. The RGB image could be affected by a high luminescence sensibility, whereas depth data allow a greater reliability and accuracy in people counting. The correctness and effectiveness of our method has been confirmed by experiments conducted in a real scenario. Furthermore, this approach has the advantage of being computationally inexpensive and flexible enough to obtain, in real time, statistical measures on the amount of people present in the bus, with the use of an Analytical Processing System (a separate process) that accesses the data stored in the database and extracts statistical data and knowledge about the bus passengers.

Keywords People counting · RGB-D · Intelligent transportation systems

D. Liciotti (✉) · A. Cenci · E. Frontoni · A. Mancini · P. Zingaretti
Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche,
Via Brecce Bianche, 60131 Ancona, Italy
e-mail: d.liciotti@pm.univpm.it

A. Cenci
e-mail: a.cenci@pm.univpm.it

E. Frontoni
e-mail: e.frontoni@univpm.it

A. Mancini
e-mail: a.mancini@univpm.it

P. Zingaretti
e-mail: p.zingaretti@univpm.it

1 Introduction

People detection and tracking from image sequences generated by cameras has been a research topic since several years. Creating an automatic system able to monitor and count people from image sequences of a dynamic environment is a very important task for several applications, where it is important to measure the flow of people who enter and exit a pre-defined area.

People counting, which can easily bring to the instantaneous estimation of the number of people present in a scene, has large importance in many advanced vision applications, such as intelligent security surveillance [1], virtual reality interfaces and human activity and behaviour understanding, e.g., assisted living [2], healthcare assistants, shopper analytics [3, 4], asset management and entertainment.

In commercial areas (e.g., shopping centres, supermarkets), knowing the number of people over different periods of time can be useful to optimize resource management during the most crowded time frame. Considering very crowded locations, such as stadiums, knowing the number of visitors at the entrance gates is highly important for security. In urban transport systems (e.g., buses, railways, airports), counting the effective passing of people in correspondence of the entrance doors is fundamental to improve the quality and the planning of service and also to monitor security. A careful knowledge of the flow of passengers is very useful to avoid accidents due to overcrowding, but also to avoid empty routes. In this way the public transport companies can plan rationally the routes according to people affluence, given the importance that nowadays the public transport system has in traffic optimization [5].

During the time, several automated methods for people counting have been developed as the traditional manual techniques not only have high labour costs, but also cannot work for counting people for long periods. Mechanical counting devices are hard to set up and also not affordable to use.

In literature there are several approaches to automatically count people and according to [6] this issue can be classified into two broad categories: Detection-based methods, which identify person-regions in an image, and Feature-based methods, which use the relationships between the number of people and the features extracted from images.

Early works on this issue are based on the use of conventional video cameras that capture appearance image sequences but do not include depth information. These works often exploit existing video surveillance systems and are simple in terms of implementation, for example, without the need to calibrate cameras [7]. Research carried out over the years has allowed the creation of a wide range of methods that give good results when applied to some benchmark videos, even though the task of identifying and tracking people using the study of the conventional image sequences is very challenging. Some of these methods use statistical learning with local features and boosting, such as HOG [8], and some others use the extraction of spatial global features [9]. As already said, these methods seem to work well on benchmark videos, but their performance is much worse when applied to real-world cases, especially when the background is dynamic and complex. Moreover, the use of RGB data

from conventional cameras is not robust to possible changes of illumination. Chen et al. [10] proposed a cost-effective bi-directional people-flow counter by analysing areas and colours of people who pass through a door or a gate. However, also this method cannot solve the problem of variable ambient light.

To overcome these limits, several approaches using depth data generated from stereoscopic cameras are proposed [11]. These methods have advantages compared to those which use traditional cameras that do not exploit depth data. However, the quality of depth data obtained using this technology is limited since data is generated by stereo matching algorithms that encounter difficulties in depth discontinuities. Moreover, this kind of approach requires synchronized and calibrated cameras determining a higher computational load.

Also infrared sensors and ultrasonic sensors have been used in many applications to determine the number of people in a scene. To go beyond the limit of conventional technologies that frequently fail because of irregular illumination changes and crowded environments, people are detected on the base of the interference of two or more sensors [12].

Recently, depth cameras like Xtion, Kinect and TOF cameras have become available at low prices, but with improved depth maps compared with that of stereo cameras. Studies using these sensors [13–15] have shown the great value of depth cameras, both in terms of accuracy and efficiency in addressing issues like occlusions and complex backgrounds. Some problems occur when many people pass the entrance at the same time, thus affecting the accuracy of the evaluation. If the person is isolated in the scene, identifying its position is quite simple, because its shape and colour distributions remain unchanged even if the person is moving. Occlusion problems increase with the number of people, becoming important in situations of crowded environments, where pixels of the foreground image could belong to several contiguous people and not to a single person. Many approaches, involving different methodologies, have been proposed to solve the problem of occlusions in crowded environments. A solution could be the use of multi-view points of the same scene, since different images of the same scene can discover parts that can appear hidden by considering a single image [16]. Other approaches solve the problem of occlusions by using image segmentation and point feature tracking, where the points of interest are some components rather than the whole image. Chan [17] applies a Gaussian process regressor on segmented blobs to estimate the people count, but this hybrid approach needs training data (i.e., annotated people counts for a number of frames). Mounting the camera on the ceiling facing downward, obtaining top view images, is another expedient used to solve the occlusion problem [18]. In this way, passing people are not overlapped each other on the obtained images, also in a crowded scene.

In this paper the main topic concerns the flow of people in a bus and in particular the count of people that come in or out a bus. In this context the count of passengers is a very difficult task [19] for many reasons. For example, it depends from the physical aspect and look (pose, body shape, clothing) of passengers and also from the fact that each bus stop presents different backgrounds that influence the quality of signal, as well as illumination, shadows and solar position [20].

We propose a method to count people at the entrance of bus using an RGB-D sensor positioned over each door of the bus and a system for image processing and understanding. The RGB image could be affected by a high luminescence sensibility, so the depth data allows a greater reliability and accuracy for the counting procedure.

The paper is organized as follows: Sect. 2 describes the system architecture, Sect. 3 introduces the general concept of the system proposed and describes the particular methodology used. The experimental results are exposed in Sect. 4 and finally, Sect. 5 presents the conclusions.

2 System Architecture

The system architecture proposed in this paper is directed to the integration of RGB-D cameras, used for our people counting algorithm, with the automated central system already present on the bus.

Buses are equipped with an Automatic Vehicle Monitoring (AVM) system, including the sub-system Automatic Vehicle Location (AVL) that takes care of automatic vehicles localization typically done by GPS. It permits the monitoring of many parameters related with vehicles in movement, such as location, speed, diagnostic of mechanical components. However, the AVM system is not limited to the monitoring of the physical parameters of the moving vehicle: in the context of local public transport it performs the monitoring of the service actually carried out by the bus, both from the mechanical point of view and from that of routes and timing. This permits to avoid inconveniences to citizens and to better manage public transport, according to the real observed needs.

The AVM system, thanks to the onboard computer that continuously processes data recorded by the GPS satellite reception service, allows to compare the position of the bus in relation to the planned service to be performed, to record the actual transit time to each stop and to carry out estimated times of arrival at the next stops of the route. All of these newly detected and estimated information may be disclosed by the onboard computer to the AVM central unit, which can in turn send updated information in real time about minutes of waiting at transit stops (hypothesizing the presence of an electronic display at the bus stop).

In addition to the benefits for the user, electrical and electronic interfacing procedures with other technological equipment already present on the bus are designed. The AVM computer is usually connected with other devices on board: electronic monitors to display and playback next scheduled stops; ticket punch machines and ticket vending machines to check the level of attendance of routes; the video surveillance system to improve the safety of the bus driver and of passengers (Fig. 1).

The goal we want to achieve with this work is primarily to increase the information that can be obtained from sensors placed in the bus to allocate the right resources, i.e. the right number of buses per route, by counting the people who enter and leave the bus. Currently, the information about the number of passengers on board is derived from the number of tickets purchased and especially from the num-

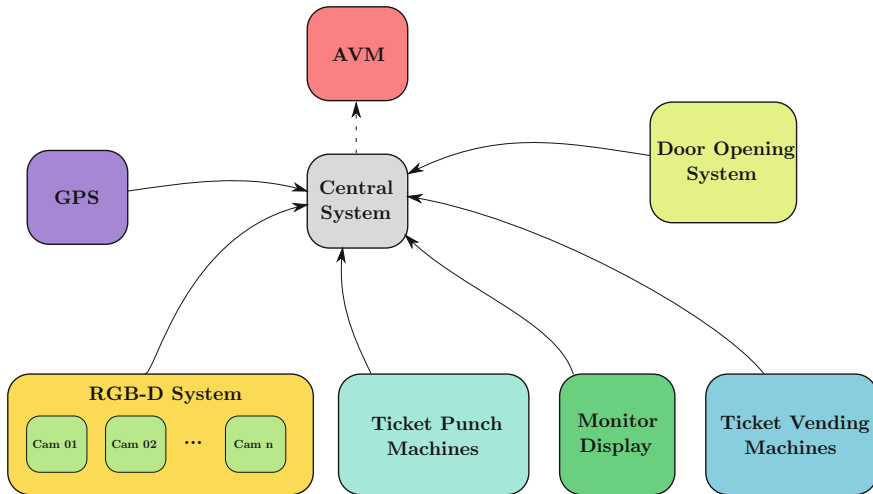


Fig. 1 The general architecture

ber of tickets validated on board, but this does not include passengers who do not have to punch, because, for example, they have a bus pass. This happens very often for the scholastic bus rides, where the majority of students who take the bus have a bus pass. Then, to obtain a more accurate estimate of people on board in this work we use the information coming from the RGB-D sensors placed in correspondence of each door to count the number of people getting in and off the bus.

Of course, to get an idea of how many people are on the bus while it is moving in a given route and at a precise time it is important to relate the information from the RGB-D cameras to the details on the bus location and time from AVM unit. So, the RGB-D cameras are connected to the central computer and to the AVM system via USB.

In existing works about people counting, depth cameras are often placed either in a top-view configuration or laterally at the same height of people. In the first case the problem of occlusion is reduced, but in this way it is difficult to capture human body details. On the contrary, in the second configuration human body details can be observed more completely, while occlusion issues can occur frequently. The oblique-view configuration could be a good compromise solution between the occlusion problem and the visibility of details of the human silhouette. The top-view configuration would be the simplest way of exploiting depth for counting people, but this layout may strongly reduce the area covered by sensors when the ceiling is not high enough as in our case, where the bus roof is usually low. For all these reasons, in the proposed system the cameras are placed in an oblique-view configuration, above each of the bus doors, as illustrated in Fig. 2: the camera is placed on the ceiling in front of the bus door facing downward with an angle of 15° with the horizontal axis in order to completely frame the door area.

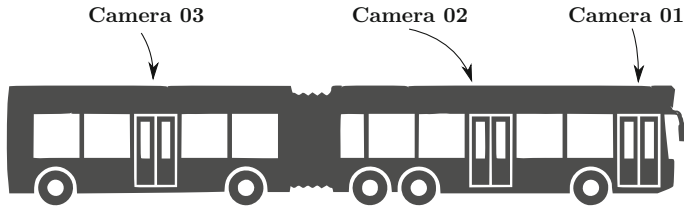


Fig. 2 Representation of the bus environment in which the system is installed

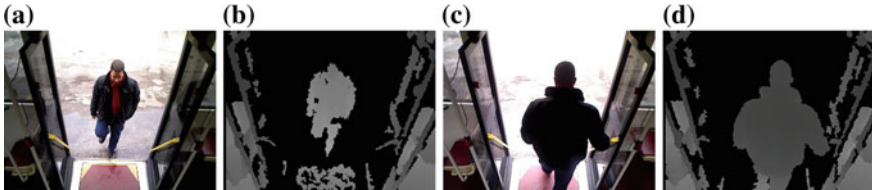


Fig. 3 RGB and depth image frames of a person who gets in and off the bus

The RGB-D sensor we have chosen in this work is the Asus Xtion Pro Live.¹ It is based on a sensor design developed by PrimeSense Ltd and it is based on the principle of structured light that allows the construction of 3D depth maps of a scene in real-time. Structured near-infrared light is directed towards a region of space and a standard CMOS image sensor is used to receive the reflected light. This sensor does not require special or powerful computer hardware. The central computer of the bus can handle the sensor just fine. Moreover, the Asus Xtion Pro Live is cheaper and smaller compared to other depth sensors and it is powered directly by the USB port of the computer, i.e., it does not need an external power supply (Fig. 3).

3 Proposed Method

The software architecture is organized as illustrated in Fig. 4.

The sensor captures depth images, with a resolution of 320×240 , at a rate up to 30 frames per second.

To elaborate sensors data we used OpenNI (Open Natural Interaction), a middleware delivered from a collaboration of PrimeSense to develop natural interaction software and devices. This software library includes skeleton tracking and gesture recognition algorithms. The function of user tracking, in which an algorithm processes the depth image to determine the position of all the joints of any user within the camera range, is provided by an OpenNI compliant module called NiTE (Natural Interface Technology for End-User).

¹https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/.

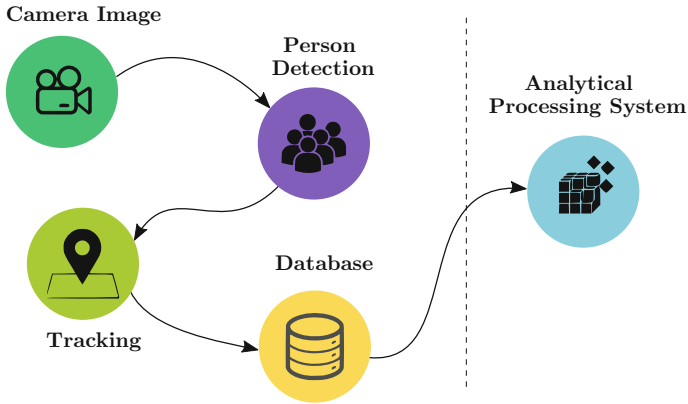


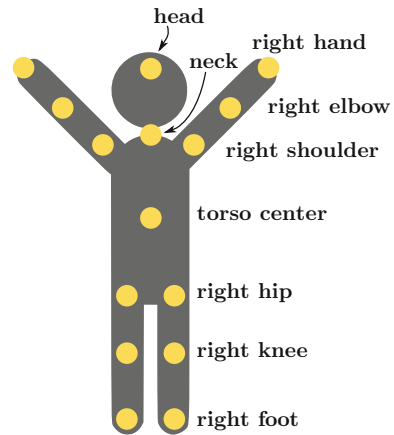
Fig. 4 The main steps of RGB-D system

More specifically, the sensor returns the stream of raw depth images in which each pixel contains a value that represents the distance between the sensor and the person. After the background subtraction, a common algorithm for detection of moving objects within a sequence of images, NiTE middleware uses computer vision algorithms to perform user segmentation, a process in which individual users and objects are separated from the background and tagged accordingly. The purpose of user segmentation is to identify and track users in the scene. A unique and persistent ID is given to each user in the scene and the main output of the user segmentation process is a label map giving user ID for each pixel.

The segmentation of the body through labelling is used to track users' bodies. The output is a set of locations of body joints. The NiTE tracking module can recognize up to 15 joints in the 3D space (Head, Neck, Torso center, Left/Right shoulder, Left/Right elbow, Left/Right wrist, Left/Right hip, Left/Right knee, Left/Right foot) as illustrated in Fig. 5 and for each joint returns the 3D coordinates and the orientation and the tracking confidence: tracked (the joint is recognized with the utmost confidence), inferred (the joint is not recognized because occluded, but its location is obtained from the position of other joints) or not tracked (the joint is not recognized by the tracker). The position and the orientation of the joint is provided in the real world coordinate system. The origin of the coordinate system is the center of the sensor, the x axis is the horizontal axis, the y axis points upward and z axis points toward the direction of increasing depth.

To avoid false alarms during the route, the RGB-D system is activated, for allowing people counting, only when the bus arrives at the bus stop and opens the doors and it is automatically deactivated when the driver closes the doors and restarts the autobus. Thanks to the skeleton-tracking algorithm provided by NiTE the system can identify people in the scene at the entrance of the bus. Then it monitors the position and the orientation of the 3D coordinates of some joints of the skeleton tracked, so that it can understand if the person is getting in or off the bus and it can automatically increase the number of people who got in or off the bus at the bus stop, respectively.

Fig. 5 Fifteen skeleton joints provided by the NiTE skeleton tracker



The experimental evaluation showed that the employed skeleton-tracking algorithm is relatively robust for our application. In particular, the position of the joints is usually detected accurately. Sometimes inaccurate detection of the joint positions occurred in correspondence of very sudden and intense movements or when occlusions are present, but we are facilitated because we need only the information about the torso joint to track a person in the scene. In particular, we use the orientation of the torso 3D coordinates to understand if the person is getting in or off the bus when he crosses the bus door. Thus, we do not need the exact reconstruction of the entire skeleton: it is not a problem if some joints, such as hip, knee or foot joints, are not visible due to occlusions for the presence of other persons.

The video timestamp information, derived from the depth sensor, together with the timing information from the AVM central unit allows us to connect the number of people on the bus with a particular bus route and stop.

All data are recorded in a local database and then pass through an Analytical Processing System, i.e., a separate process that accesses the data published on the database and extracts statistics and knowledge about the passengers.

The main indicators we extrapolate from the database and that are useful to evaluate the passengers' flows are:

- N_i^p and N_o^p , the number of people who, respectively, get in and off the bus during a bus ride;
- $\overline{N_i^p}$ and $\overline{N_o^p}$, the average number of people who, respectively, get in and off the bus during a bus ride;
- $N^s = N_i^p - N_o^p$, the number of people who stay in the bus during a bus ride;
- $\overline{N^s} = \overline{N_i^p} - \overline{N_o^p}$, the average number of people who stay on the bus during a bus ride;
- $N_{day_i}^p$ and $N_{day_o}^p$, the number of people who, respectively, get in and of the bus during all the day (to know the difference between working days and holidays);

- $\overline{N_{day_i}^p}$ and $\overline{N_{day_o}^p}$, the average number of people who, respectively, get in and of the bus during all the day;
- $N_{stop_i}^j$ and $N_{stop_o}^j$, the number of people who, respectively, get in and off the bus for each bus stop;
- T_{stop}^j , the rise and descent time of people on/from the bus for each bus stop.
- $\overline{T_{tot}} = \left(\sum_j T_{stop}^j \right) / N_{stop}$, the average rise and descent time of people on/from the bus;

We use these parameters to make statistics: using these data in relation to time bands and bus routes we can facilitate the work for resource allocation and so the decision of the number of buses to be arranged for a given route and in a certain time band. With the availability of the information derived from the ticket punch machines and bus passes, it would be possible to infer the presence of passengers without tickets (gatecrashers), especially when there is a big difference between people counted by the system and those who validated tickets.

4 Experimental Results

This section presents some experimental results aimed at highlighting the performances of the system with real tests.

We first did some proof tests when the bus was at a standstill. To test the performance of our application we used 100 passages of people getting in the bus and 100 of people getting off the bus. The system was able to recognize 95 people out of 100 who entered, and 97 people out of 100 that came out. From data analysis we detected that all the errors occurred in situations where a group of people entered together one close to the other, i.e. due to severe occlusions.

Then we tested the system during a real bus ride of about 10 km during a working day. Acquisitions have been performed within half an hour in the morning with sunlight, as we can see from the images depicted in Fig. 3. We did not test system robustness at different illumination conditions in this preliminary work.

We have installed three cameras, one for each door of the bus. In the bus under examination it is possible to use each door either for entering or leaving the bus (see Fig. 2). The total recording time for each camera was about 10 min, i.e. 45 s of registration time at each bus stop for the 13 bus stops during the route. We totally monitored 108 people and the system was able to correctly track their skeletons.

The system extracts a high number of parameters and the most significant are shown in Fig. 6.

Figure 6a shows the number of people getting in the bus counted by the system at each bus stop (yellow bars) compared with the ground truth (blue bars). The x axis reports the bus stop number. Unexpected errors occurred at bus stop number four, ten and twelve, where the system counts one more person, probably due to a quick about-turn of the monitored person.

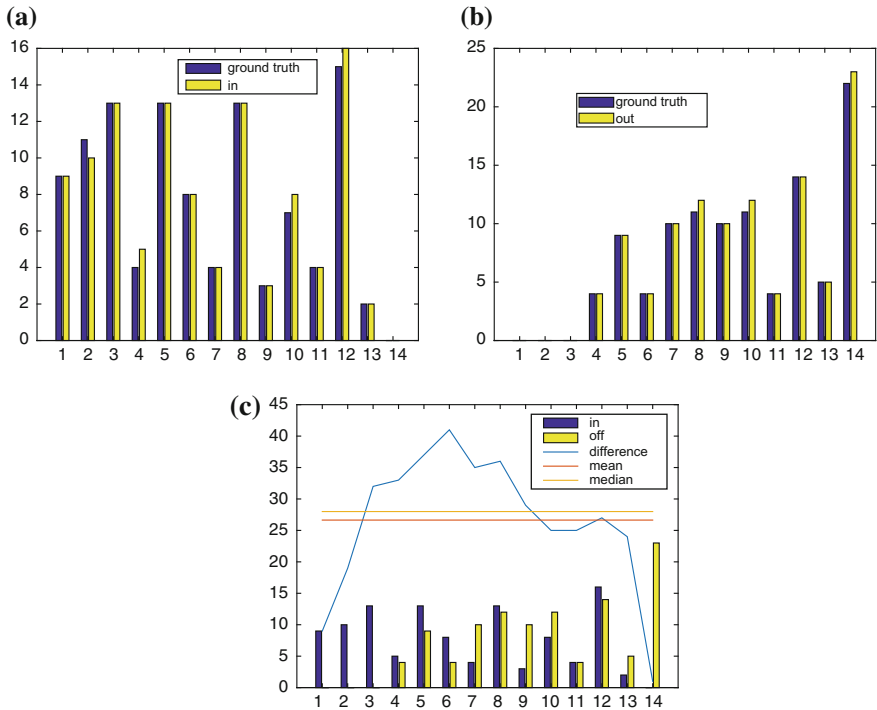


Fig. 6 Some graphics used to evaluate passengers' flows

Figure 6b shows the number of people getting off the bus counted by the system at each bus stop (yellow bars) compared with the ground truth (blue bars). By observing this graph, we can see that in the first few stops, as expected, there is no person getting off the bus, whereas most of the people got off at the last stop.

The combination of the information reported in these two graphs (Fig. 6c) allows a better understanding of the number of passengers getting in (yellow bars) and off (blue bars) at each bus stop. The light blue line indicates the number of passengers actually in the bus after each bus stop, i.e., the cumulative difference between people who get off and on the bus. The red and the orange lines indicate, respectively, the mean and the median of the number of passengers on board during the bus ride. By combining this information with the bus route and the specific time bands, public transport companies can extract important information to better allocate resources: knowing bus seating capacity they may, for example, decide whether to require the use of a greater number of buses for a specific ride.

5 Conclusion and Future Works

In this work, we presented an application for the automatic counting of the number of people getting in or off a bus, by using a people detection and tracking algorithm applied to depth images from a single RGB-D camera above each of the bus doors.

The system has received many interest from public transport companies, since they can derive useful information about passengers' flows from the extracted data. With this technology, it is possible to find some important indicators about: the amount of people who take the bus every day, the busiest bus rides in terms of route locations and of time bands, the busiest days of the week. In fact, even if a ticket punch machine is present, usually it cannot take into account passengers with bus passes, as in Italy. From these indicators, extracted from the system, the public transport companies can profitably allocate human resources and means of transport.

We placed an RGB-D camera in front of each bus door to count the people who enter or exit each door. We chose the best possible configuration. The camera is placed on the ceiling in front of the bus door facing downward with an angle of 15° with the horizontal axis. We avoid a lateral configuration with the camera placed at about the same height as people in order to reduce occlusions. We also avoid the top configuration viewing downward because the bus ceiling is not high enough so that this layout strongly reduces the registered area.

In the experimental phase, the test of the system in a real environment has provided very interesting results. We have preliminary results, since we have monitored only a bus ride, but we have observed a good accuracy. In the future, we want to test our system also under different light conditions during different hours of the day, with different weather conditions, during the night with the lights of the bus on and with the illumination of bus shelters.

The system could be improved also with person re-identification, to determine if a given individual has been previously observed by the camera and to use this information to better count the amount of people who stay in the bus. The accuracy of the system could be also improved by using more precise depth sensors like Kinect v2 and by increasing the image processing algorithm performances. Finally, let us highlight how the proposed architecture is easy to install and especially affordable due to the low cost of the components.

References

1. Zhou, J., Hoang, J.: Real time robust human detection and tracking system. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE, pp. 149–149 (2005)
2. Liciotti, D., Massi, G., Frontoni, E., Mancini, A., Zingaretti, P.: Human activity analysis for in-home fall risk assessment. In: 2015 IEEE International Conference on Communication Workshop (ICCW). IEEE, pp. 284–289 (2015)

3. Liciotti, D., Contigiani, M., Frontoni, E., Mancini, A., Zingaretti, P., Placidi, V.: Shopper analytics: a customer activity recognition system using a distributed rgb-d camera network. In: *Video Analytics for Audience Measurement*, pp. 146–157. Springer (2014)
4. Mancini, A., Frontoni, E., Zingaretti, P., Placidi, V.: Smart vision system for shelf analysis in intelligent retail environments. In: *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, pp. V004T08A045–V004T08A045 (2013)
5. Catani, L., Frontoni, E., Zingaretti, P., Di Pasquale, G.: Efficient traffic simulation using busses as active sensor network. In: *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, pp. 889–894 (2011)
6. Bondi, E., Seidenari, L., Bagdanov, A.D., Del Bimbo, A.: Real-time people counting from depth imagery of crowded environments. In: *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, pp. 337–342 (2014)
7. Conte, D., Foggia, P., Percannella, G., Tufano, F., Vento, M.: A method for counting moving people in video surveillance videos. *EURASIP J. Adv. Signal Process.* **2010**, 5 (2010)
8. Zhu, Q., Yeh, M.C., Cheng, K.T., Avidan, S.: Fast human detection using a cascade of histograms of oriented gradients. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1491–1498. IEEE (2006)
9. Felzenszwalb, P.F.: Learning models for object recognition. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001. CVPR 2001, vol. 1, pp. 1–1056. IEEE (2001)
10. Chen, T.H., Chen, T.Y., Chen, Z.X.: An intelligent people-flow counting method for passing through a gate. In: *2006 IEEE Conference on Robotics, Automation and Mechatronics*, IEEE, pp. 1–6 (2006)
11. Van Oosterhout, T., Bakkes, S., Kröse, B.J.: Head detection in stereo data for people counting and segmentation. In: *VISAPP*, pp. 620–625 (2011)
12. Shbib, R., Zhou, S., Ndzi, D., Al-Kadhimi, K.: Distributed monitoring system based on weighted data fusing model. *Am. J. Social Issues Humanit.* **3**(2) (2013)
13. Han, J., Shao, L., Xu, D., Shotton, J.: Enhanced computer vision with microsoft kinect sensor: a review. *IEEE Trans. Cybern.* **43**(5), 1318–1334 (2013)
14. Hsieh, C.T., Wang, H.C., Wu, Y.K., Chang, L.C., Kuo, T.K.: A kinect-based people-flow counting system. In: *2012 International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS)*, IEEE, pp. 146–150 (2012)
15. Liu, J., Liu, Y., Zhang, G., Zhu, P., Chen, Y.Q.: Detecting and tracking people in real time with rgb-d camera. *Pattern Recognit. Lett.* **53**, 16–23 (2015)
16. Mohedano, R., Del-Blanco, C.R., Jaureguizar, F., Salgado, L., García, N.: Robust 3d people tracking and positioning system in a semi-overlapped multi-camera environment. In: *15th IEEE International Conference on Image Processing*, 2008. ICIP 2008. IEEE, pp. 2656–2659 (2008)
17. Chan, A.B., Liang, Z.S.J., Vasconcelos, N.: Privacy preserving crowd monitoring: counting people without people models or tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. CVPR 2008. IEEE, pp. 1–7 (2008)
18. Zhang, X., Yan, J., Feng, S., Lei, Z., Yi, D., Li, S.Z.: Water filling: unsupervised people counting via vertical kinect sensor. In: *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, IEEE, pp. 215–220 (2012)
19. Lengvenis, P., Simutis, R., Vaitkus, V., Maskeliunas, R.: Application of computer vision systems for passenger counting in public transport. *Elektronika ir Elektrotechnika* **19**(3), 69–72 (2012)
20. Yahiaoui, T., Khoudour, L., Meurie, C.: Real-time passenger counting in buses using dense stereovision. *J. Electron. Imaging* **19**(3), 031202–031202 (2010)

A Powerful and Cost-Efficient Human Perception System for Camera Networks and Mobile Robotics

Marco Carraro, Matteo Munaro and Emanuele Menegatti

Abstract In this work, we present a software library which enables the efficient use of the Kinect One, a time-of-flight RGB-D sensor, with the nVidia Jetson TK1, an ARM-based embedded system, for the purpose of people detection. Our software exploits nVidia CUDA to process all data necessary for robust people detection algorithm and other perception algorithms by parallelizing the generation of the 3D point cloud and many pixel-wise operations on both the raw depth and the infrared images coming from the Kinect One sensor. The library developed has been released as open-source and the whole system has been tested as a people detection node in an open source multi-node RGB-D tracking framework (OpenPTrack). The results gathered show that the proposed system can be effectively used as a people detection node, outperforming the state-of-the-art in terms of people detection frame rate not only with the nVidia Jetson, but also with non-embedded computers.

Keywords People detection and tracking · Mobile robotics · Kinect one · nVidia Jetson · OpenPTrack

1 Introduction

In the last years, human detection and tracking algorithms proved to be useful not only in surveillance, but also in different applications like service robotics, ambient assisted living (AAL), culture and arts installations. People perception was boosted

M. Carraro (✉) · M. Munaro (✉) · E. Menegatti (✉)
Department of Information Engineering, University of Padova,
Via Gradenigo 6/A, 35131 Padova, Italy
e-mail: marco.carraro@dei.unipd.it; carraro1@dei.unipd.it
URL: <http://robotics.dei.unipd.it>

M. Munaro
e-mail: matteo.munaro@dei.unipd.it; munaro@dei.unipd.it

E. Menegatti
e-mail: emg@dei.unipd.it

by the introduction of new low cost RGB-D sensors such as the Microsoft Kinect. The second generation of the Microsoft Kinect sensor, called *Kinect One*, come with an increased image resolution and a better depth estimation in terms of accuracy and noise [1]. These improved performances are due to a change in the sensing technology implemented in the Kinect One, no longer IR triangulation as for Kinect 360, but Time-of-flight (ToF) technology [2]. Unfortunately, Kinect One is even more computationally demanding of its predecessor, because of the large amount of data of each frame (tens of megabytes per frame at 30 frames per second). To keep the pace with the sensor maximum frame rate, a dedicated graphic card is required by the Linux driver. As a result, this sensor is not likely to be used on mobile robots fitted with embedded computers and requires powerful laptop (or industrial) computers. In this work, we demonstrate the use of an nVidia Jetson TK1 embedded computer as an acquisition and processing unit for the Kinect One sensor. This is enabled by the new software library we propose in this paper. The complete system is tested and demonstrated in the application of implementing a perception node of ROS (Robot Operating System) [3] for people detection and tracking from RGB-D data. We show that the integrated hardware and software system is well suited for application in service robots and also in distributed perception networks composed by dozens of nodes. The main advantage of the proposed solutions are high data processing throughput and the low power consumption of the Jetson embedded platform. The proposed system is tested and validated in distributed RGB-D camera network which uses OpenPTrack [4] (the open-source people tracking library). In summary, the contribution of this work is two-fold:

- We propose a new library which permits the usage of the Kinect One with CUDA-capable embedded systems and we demonstrate the validity of this work obtaining suitable frame-rates for real-world applications as people detection and tracking.
- We release this library as open-source¹ together with a ROS bridge² to make it work out-of-the-box with the most popular framework for the robotics community.

The remainder of the paper is organized as follows. Section 2 reviews the state-of-the-art of RGB-D sensors. Section 3 explains the features of the new library we developed. In Sect. 4, experiments are reported and results are shown in terms of acquisition and people tracking frequency. Finally, Sect. 5 draws the conclusions of this work.

¹<https://github.com/OpenPTrack/libfreenect2>

²https://github.com/OpenPTrack/kinect2_bridge

2 State-of-the-Art

2.1 RGB-D Sensors

When dealing with mobile robots or complex surveillance scenarios, two dimensional information is not always sufficient to obtain reliable results of detection and tracking in real time. Furthermore, passive 3D solutions such as stereo cameras require additional processing for computing depth information and they are not able to estimate depth for lowly textured areas. For these reasons, the advent of active and low-cost 3D sensors, such as Microsoft Kinect, significantly improved the research on autonomous mobile robotics and computer vision. The first-generation Kinect is an RGB-D sensor that provides both color and depth data at VGA resolution. Depth is estimated by means of an active triangulation process [5] between an infrared pattern projector and an infrared camera, i.e., the position of each 3D point is the intersection of the optical rays corresponding to a dot of the projector and the one of the considered pixel in the infrared camera. The Kinect is widely used in computer vision and robotics for Simultaneous Localization and Mapping (SLAM) [6], people detection and tracking [4, 7, 8], short-term and long-term people re-identification [9–11], ambient assisted living [12–14] and many other applications. Besides the wide usage of this sensor, it has the drawback of not being able to estimate depth information outdoors because the infrared component of the sunlight interferes with the pattern projected by Kinect. Furthermore, the depth estimation error increases quadratically with the distance [15]. To overcome these problems, in late 2013, Microsoft released the second generation of the Kinect sensor. This new RGB-D camera relies on the continuous wave time-of-flight [2] technology to infer depth, that is an array of emitters sends out a modulated signal that travels to the measured point, gets reflected and is received by the CCD of the sensor. The sensor acquires a 512×424 depth map and a 1920×1080 color image at 15 to 30 fps depending on the lighting condition, since the sensor exploits an auto-exposure algorithm. Kinect One outperforms its predecessor on several aspects. In particular, it works outdoors up to four meters and depth accuracy remains constant while increasing the distance [1]. However, since the data resolution is higher than for Kinect v1, processing Kinect One data is computationally more demanding and turns out to be unsuitable for embedded systems. In this work, in order to overcome this problem, we modified the Linux driver and ROS wrapper for the Kinect One so that they allowed to obtain color and point cloud data at more than 20 fps with the nVidia Jetson, a CUDA-capable embedded system. While Kinect One is directly supported in Microsoft Windows with a free driver and SDK provided by Microsoft, the only driver available in Linux is unofficial and open

source and is called *libfreenect2*.³ In this work, we also used nVidia CUDA [16], a scalable library for exploiting the General Purpose GPU (GP-GPU) computing on nVidia GPUs. Our work improved a first working library [17] developed in CUDA. The numerical comparisons between the different versions of these libraries are presented in Sect. 4.

2.2 People Detection and Tracking in Camera Networks

The ability to autonomously detect and track humans in camera networks is one of the most important issues in robotics and computer vision applications. The problem can be split into two different sub-problems: (1) perform people detection and tracking within a view of a single camera and (2) maintain the same ID for the same person seen by different cameras [18]. For solving (1), a wide set of works in literature relies on RGB data alone [19, 20], while, recently, new methods were developed for using RGB-D data to perform this task [4, 21, 22]. The problem of associating the correct ID among different cameras (2) is often solved based on the knowledge of camera poses and by exploiting features extracted from the person motion and appearance. OpenPTrack is an open source software for multi-camera calibration and people tracking in RGB-D camera networks. It allows scalable, robust and real-time person tracking using affordable off-the-shelf components, such as Kinect One, and an open source codebase. It constitutes a powerful tool for enabling interactive experiences for education, arts and culture, but it is also exploited for guaranteeing people safety in industrial environments [23]. The OpenPTrack nodes which use the Kinect One are usually equipped with a powerful computer with a dedicated GPU because the sensor is eager of performance. The use of these computers causes space problems, high costs and high power consumption. Moreover, the OpenPTrack network can potentially be made of dozens of nodes, thus amplifying these problems. Therefore, the use of embedded systems as the nVidia Jetson can fix these issues, allowing the building of large networks.

3 Methodology

Our objective is to acquire data from the Kinect One sensor with the nVidia Jetson TK1 at high frame rate and integrate the camera into an OpenPTrack network to perform people detection and tracking. The state-of-the-art Linux driver for the Kinect One, *libfreenect2*, is not able to perform the operation needed by OpenPTrack at a frame rate suitable for people tracking. For this reason, in order to improve the performance, we use nVidia CUDA, shifting computational burden from the ARM CPU to the GPU of the embedded system. In Fig. 1a, the overall state-of-the-art system

³<https://github.com/OpenKinect/libfreenect2>

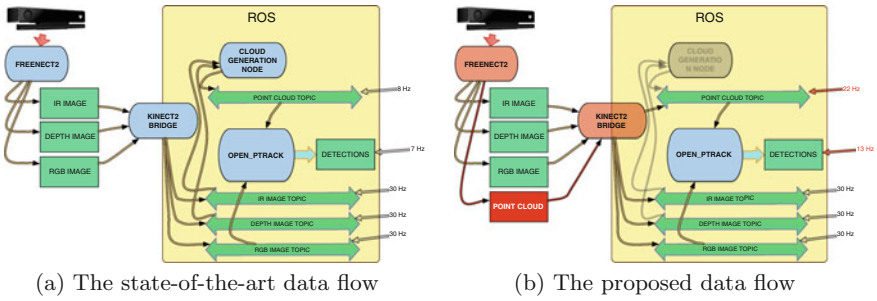


Fig. 1 High level representations of the two data flows. On the *left*, the state-of-the-art data flow of the OpenPTrack system using the Kinect One and the Jetson embedded system. The 3D point cloud is computed outside the *libfreenect2* library by the ROS nodelet *cloud_generation_node*. On the *right*, the proposed data flow of the same system. The 3D point cloud is now directly computed within the new version of *libfreenect2* and streamed by the new version of *kinect2_bridge*, thus the external nodelet is no longer required

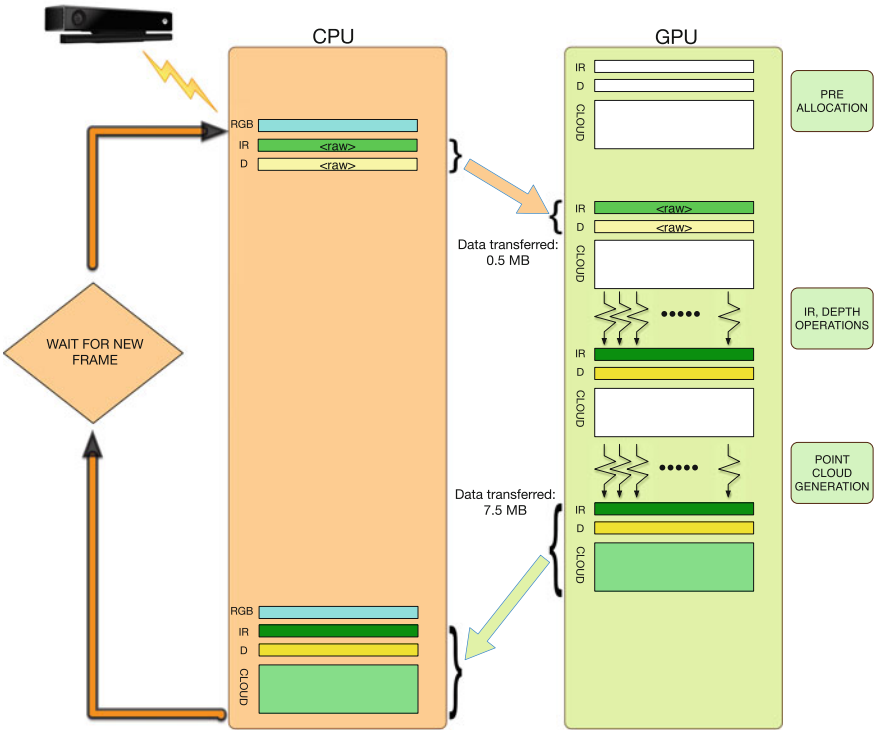


Fig. 2 The processing flow performed by our library for each single frame. At start-up, the space for the needed data is allocated once in the GPU memory, then, for each frame, the data coming from the CPU memory are processed to achieve the final data. The data transferred from CPU to GPU consist of the raw IR and depth data, while the data transferred from GPU to CPU consists of the final IR, depth and point cloud obtained after the parallel computations have been made

needed to perform people detection with Kinect One is shown. At first, the Linux driver for Kinect One, *libfreenect2*, is used to obtain the raw data from the sensor. OpenPTrack is based on the ROS middleware, thus, in order to interface the Kinect camera with this people tracking library, we need a ROS wrapper of the *libfreenect2* driver. This wrapper is implemented in the *kinect2_bridge* ROS package, that reads sensory data obtained from the driver and streams them to ROS topics whenever a ROS node requests them. In this work, we adapt also this wrapper to be compatible with the proposed version of *libfreenect2*. The algorithm exploited by the standard version of the driver for computing depth image, infrared (IR) image and point cloud is reported in Algorithm 1, while the one we propose in this paper is reported in Algorithm 2 and illustrated in Fig. 2. The RGB information that comes from the sensor does not need any additional computation, so we store it directly without passing it to the GPU. The operations needed to transform the raw depth and infrared data in the final data needed by the tracking library are all pixel-wise or consist of operations on the neighborhood of each pixel. These types of functions are implementable in CUDA, thus lowering the final computational complexity from $O(N)$ to $O(1)$, with N number of pixels. Furthermore, in our approach, the generation of the point cloud from IR and depth data is not computed any more by the ROS wrapper, but directly within the driver, thus shifting computational burden from the ARM CPU to the GPU of the embedded system. Each parallel function we implemented requires as input the number of CUDA threads that will concurrently operate. Since the dimension of the depth and infrared images is 512×424 , we designed a grid of 512 threads per block with $\left\lfloor \frac{512 \cdot 424 + (512 - 1)}{512} \right\rfloor = 424$ blocks [24]. OpenPTrack, the library we use for performing people detection, requires as input from the Kinect One a point cloud filled with 3D points *colored* with the corresponding intensity obtained from the infrared image. Here, the infrared is preferred to the RGB because the former is constant also in the dark. However, to help the people detection module, an intensity rescaling operation has to be performed on the intensity image in order to improve its contrast, thus helping people detection. Also this computation is performed in our version of the *libfreenect2* driver by exploiting the parallelization achievable with CUDA. In the next sections, we detail the important steps developed in this work.

Algorithm 1: Standard algorithm of the *libfreenect2* driver for each frame

input : a frame $F = (I, D, M)$ where I is the raw infrared image, D is the raw depth image and M is the camera calibration matrix

output: The final infrared image \hat{I} , the final depth image \hat{D} and the point cloud P

```

1 foreach pixel  $p$  of  $D$  do
2   | computeDepth( $p, \hat{p}$ )
3 foreach pixel  $p$  of  $I$  do
4   | computeIR( $p, \hat{p}$ )
5 foreach pixel  $p_d$  of  $\hat{D}$  and the correspondent pixel  $p_i$  of  $\hat{I}$  do
6   | computePoint( $p_d, p_i, p_p$ )

```

Algorithm 2: Our version of the `libfreenect2` driver for each frame

input : a frame $F = (I, D, M)$ where I is the raw infrared image, D is the raw depth image and M is the camera calibration matrix

output: The final infrared image \hat{I} , the final depth image \hat{D} and the point cloud P

```

1 Image_size  $\leftarrow$  512 * 424;
2 Block_size  $\leftarrow$  512;
3 Grid_size  $\leftarrow$   $\lfloor \frac{Image\_size + (Block\_size - 1)}{Block\_size} \rfloor$ ;
4 memoryCopyFromCPUToGPU( $D, I$ );
5 computeDepth( $\langle\langle\langle$ Grid_size, Block_size $\rangle\rangle\rangle$ )( $D, \hat{D}$ );
   ; // pixel-wise depth computation
6 computeIR( $\langle\langle\langle$ Grid_size, Block_size $\rangle\rangle\rangle$ )( $I, \hat{I}$ );
   ; // pixel-wise IR computation
7 computePointCloud( $\langle\langle\langle$ Grid_size, Block_size $\rangle\rangle\rangle$ )( $\hat{D}, \hat{I}, M, P$ );
   ; /* pixel-wise point cloud generation (every pixel
   corresponds to a point) */
8 memoryCopyfromGPToCPU( $\hat{D}, \hat{I}, P$ );
```

3.1 Memory Management

When exploiting GPU processing, the typical bottleneck is the overhead due to the data transfer between the central memory and the GPU memory [25]. To prevent these passages to affect the overall performance of our algorithm, we pre-allocate in the GPU memory the exact space needed by the infrared and depth images and the point cloud. This way, we avoid new allocations whenever a new frame is acquired by the sensor, thus making the GPU only overwrite the previous frame information. The memory passages performed by our algorithm consist of the copy of the input data from CPU to GPU (only IR and depth raw images) and of those needed to transfer the output data from GPU to CPU central memory at the end of GPU processing. Figure 2 highlights the memory transfers performed by our algorithm. In particular, the memory transfers are less than 500 KB for the first transfer (from CPU to GPU) and the same quantity plus 7 MB (the point cloud) for the transfer-back. The maximum data transfer required by the application is then $7.5 * 30 = 225\text{MB/s}$ which is about the 1.51 % of the total Jetson GPU bandwidth and the 0.067 % of a nVidia Geforce GTX Titan Black total GPU bandwidth.

3.2 Point Cloud Generation

A 3D point cloud is the typical input of several algorithms in 3D computer and robot vision [26, 27]. This data structure is needed also by OpenPTrack to perform people detection while being robust to light changes. This data type is built from three pieces of information: the depth map, the infrared image and the intrinsic parameters

of the sensor. The point cloud is computed at each new frame after that the depth data have been processed and become available. Given the equations of 3D perspective projection:

$$\begin{bmatrix} x \\ y \\ d \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \Rightarrow \begin{cases} X = \frac{(x-c_x)d}{f_x} \\ Y = \frac{(y-c_y)d}{f_y} \\ Z = d \end{cases} \quad (1)$$

where (x, y) are the coordinates of a pixel in the depth image, d is the measured depth, f_x, f_y, c_x, c_y are the intrinsic parameters of the sensor and represent the focal lengths and the optical centers of the camera, we can obtain (X, Y, Z) , the 3D coordinates of the correspondent point in the point cloud. For what concerns color information, in order to maintain the same structure as for point cloud colored with RGB information, we consider all the three R, G and B channels and fill them with i , that is the infrared intensity of the pixel (x, y) , in the IR image. Indeed, people detection performed on infrared information is more robust to changes in visible light. We fill the three R, G and B channels with the same i values in order to have the same algorithms working both on real RGB-colored point clouds and intensity-colored clouds. It is worth noting that, in this work, the additional computational burden due to the filling and use of three identical intensity channels is actually negligible. The space in memory allocated for the cloud is then filled in with $P = (X, Y, Z, R, G, B)$ points. Once the point cloud is filled, it is transferred back to the CPU (bottom part of Fig. 2). An illustration of this process and some examples of the resulting point cloud are shown in Fig. 3.

3.3 The kinect2_bridge Wrapper

The `kinect2_bridge` wrapper is an executable which streams data from the `libfreenect2` driver library to the ROS topics whenever these data are requested. We had to adapt this wrapper because our version of the driver directly generates the point cloud that was computed by an external ROS node with the standard versions of the driver and the wrapper (see Fig. 1). This choice allowed to save time avoiding to allocate, transfer and fill each point cloud.

4 Experiments

To show the improvements of the proposed system with respect to the state of the art, we compared the outputs of two implementations of our system with a people detection and tracking system presented in scientific literature [17]. Table 1 compares the output of the software library proposed in [17] with two implementations

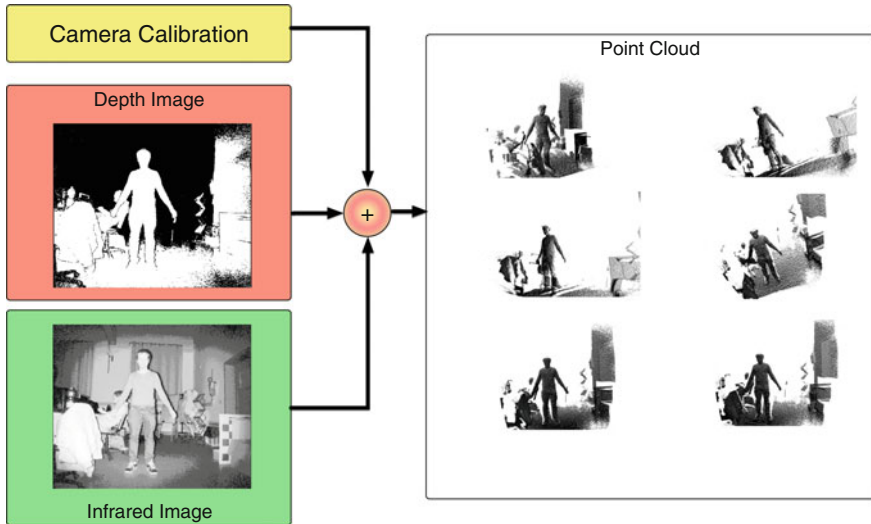


Fig. 3 The infrared point clouds are obtained from the depth and infrared images and by exploiting the intrinsic parameters of the sensor. Example of six views of the resulting cloud are reported on the *right*

Table 1 Frame rate comparison with a Jetson TK1 embedded system and a Jetson TX1 embedded system

Libraries used	Point cloud topic frame rate (Hz)	OpenPTrack detection frame rate (Hz)
Ours with the Jetson TX1	30	23
Ours with the Jetson TK1	22	13.7
[17]	8	7

of our system which used as processing unit respectively: the nVidia Jetson TK1 and the more recent (and much more powerful) nVidia Jetson TX1. The analysis is performed in terms of the following parameters: numbers of processed point clouds per second and people detection frequency. As reported in Table 1, the publishing rate of the point cloud gets to 22 point clouds streamed per second with an improvement of a factor 3. The people detection frame rate doubles with our approach reaching 13.7 frames per second, that enables real-time people tracking with continuous tracks. The same test has also been performed by substituting the Jetson embedded system with an high-end laptop, with an Intel i5-4210M CPU and a nVidia Quadro K1100M GPU. The frame rates reported in Table 2 prove that the proposed system increases the overall performance also on a machine with CUDA-enabled graphics card. A qualitative comparison of the output of the people detection module of OpenPTrack run by the embedded system and by the laptop computer has been performed by comparing the tracks of two moving people (see Figure Fig. 4 in which the two people are

Table 2 Frame rate comparison with a high-end laptop

Libraries used	Point cloud topic frame rate (Hz)	OpenPTrack detection frame rate (Hz)
Ours	30	25.4
[17]	30	22

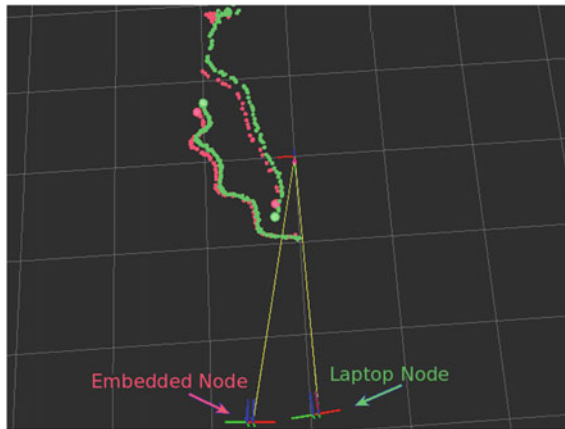


Fig. 4 Output of OpenPTrack people tracking algorithm. Two people are moving in the environment and are simultaneously observed by the two tracking systems: a Kinect One attached to a high-end laptop (*green* tracks) and a Kinect One attached to the embedded system nVidia Jetson TK1 (*red* tracks). The data of the two tracks are perfectly overlapped, a small offset of the two tracks is introduced afterwards in the picture only for visualization purposes

correctly detected by both nodes). The track generated by the embedded node (in red) is slightly less dense than the one generated by the laptop (in green) because of the differences in frame rate outlined in Tables 1 and 2. The tracking output data from the two systems are perfectly overlapped, a small offset of the two tracks is introduced afterwards in the picture only for visualization purposes. These tests confirm that the hardware and software system proposed in this work creates a compact, powerful and cost-efficient perception system (here tested as a people tracking system) based on a Kinect One attached to an embedded computer with CUDA processing capability. The hardware and software system we proposed in this work is of major interest also for the mobile robotics community, because on board of a mobile robot weight and power consumption are an issue. Indeed, processing data on board of a mobile robot with an embedded computer (which can be powered with 12 V) instead of a laptop (which needs to be powered with a 220 V) while keeping real-time performance is a great advantage. Moreover, the software library we developed outputs all the information needed by state of the art robust navigation algorithms and it is compatible with ROS, the de-facto standard used in the robotics community.

5 Conclusion

In this work, we presented a hardware and software perception system which is powerful and cost-efficient. It has been designed for use in camera networks and mobile robotics. In this paper, it has been tested on people tracking algorithms. The system is composed of an embedded computer (i.e. the nVidia Jetson TK1), an RGB-D camera (i.e. the Kinect One), and a multi-camera people tracking system (i.e. OpenPTrack, an open-source ROS-based software library). The main contribution of this work is a novel software library to process at high frame-rate the heavy stream of data coming from a Kinect One exploiting the GPU capabilities of the Jetson TK1. A second major contribution is the wrapper which makes the *libfreenect2* library compatible with the ROS middleware. The proposed system was tested performing people detection in a RGB-D camera network using OpenPTrack. The results shows that our system allowed to triple the rate of generation of point clouds and to double the people detection frame rate with respect to the state-of-the-art library [17]. This work is not only fundamental for distributed perception systems composed of many perception units, where each unit has to be as powerful and cheaper as possible, but also as perception unit for mobile robots, given its compatibility with ROS. To provide the best benefit for the computer vision and robotics community, but also for developers of human-computer interaction applications, we released all this work as open source within the OpenPTrack repository⁴ [28].

Acknowledgements Portions of this work have been supported by NVidia, OpenPerception and the REMAP center at UCLA. The authors would like to thank Randy Illum and Jeff Burke at UCLA for the extensive testing of the developed software.

References

1. Zennaro, S., Munaro, M., Milani, S., Zanuttigh, P., Bernardi, A., Ghidoni, S., Menegatti, E.: Performance evaluation of the 1st and 2nd generation kinect for multimedia applications. In: 2015 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2015)
2. Gokturk, S.B., Yalcin, H., Bamji, C.: A time-of-flight depth sensor-system description, issues and solutions. In: Conference on Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04, pp. 35–35. IEEE (2004)
3. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA Workshop on Open Source Software, vol. 3, p. 5 (2009)
4. Munaro, M., Basso, F., Menegatti, E.: Openptrack: open source multi-camera calibration and people tracking for RGB-D camera networks. Robot. Auton. Syst. (2015)
5. Khoshelham, K., Elberink, S.O.: Accuracy and resolution of kinect depth data for indoor mapping applications. Sensors **12**(2), 1437–1454 (2012)

⁴<https://github.com/OpenPTrack/libfreenect2/tree/jetson-dev>
https://github.com/OpenPTrack/kinect2_bridge/tree/jetson-dev

6. Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., Burgard, W.: An evaluation of the RGB-D slam system. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 1691–1696. IEEE (2012)
7. Munaro, M., Menegatti, E.: Fast RGB-D people tracking for service robots. *Auton. Robots* **37**(3), 227–242 (2014)
8. Munaro, M., Basso, F., Michieletto, S., Pagello, E., Menegatti, E.: A software architecture for RGB-D people tracking based on ROS framework for a mobile robot. In: *Frontiers of Intelligent Autonomous Systems*, pp. 53–68. Springer (2013)
9. Fleuret, F., Shitrit, H.B., Fua, P.: Re-identification for improved people tracking. In: *Person Re-identification*, pp. 309–330. Springer (2014)
10. Munaro, M., Ghidoni, S., Dizmen, D.T., Menegatti, E.: A feature-based approach to people re-identification using skeleton keypoints. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 5644–5651. IEEE (2014)
11. Nanni, L., Munaro, M., Ghidoni, S., Menegatti, E., Brahmam, S.: Ensemble of different approaches for a reliable person re-identification system. *Appl. Comput. Inf.* (2015)
12. Carraro, M., Antonello, M., Tonin, L., Menegatti, E.: An open source robotic platform for ambient assisted living. In: *Artificial Intelligence and Robotics (AIRO)* (2015)
13. Fischinger, D., Einramhof, P., Papoutsakis, K., Wohlkinger, W., Mayer, P., Panek, P., Hofmann, S., Koertner, T., Weiss, A., Argyros, A., et al.: Hobbit, a care robot supporting independent living at home: first prototype and lessons learned. *Robot. Auton. Syst.* (2014)
14. Ghidoni, S., Anzalone, S.M., Munaro, M., Michieletto, S., Menegatti, E.: A distributed perception infrastructure for robot assisted living. *Robot. Auton. Syst.* **62**(9), 1316–1328 (2014)
15. Basso, F., Pretto, A., Menegatti, E.: Unsupervised intrinsic and extrinsic calibration of a camera-depth sensor couple. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 6244–6249. IEEE (2014)
16. Nickolls, J., Buck, I., Garland, M., Skadron, K.: Scalable parallel programming with CUDA. *Queue* **6**(2), 40–53 (2008)
17. Xiang, L.: libfreenect2 CUDA library. <https://github.com/xlzl/libfreenect2> (2015). [Online; accessed 2016-02-03]
18. Wang, X.: Intelligent multi-camera video surveillance: a review. *Pattern Recognit. Lett.* **34**(1), 3–19 (2013)
19. Vezzani, R., Baltieri, D., Cucchiara, R.: Pathnodes integration of standalone particle filters for people tracking on distributed surveillance systems. In: *Image Analysis and Processing—ICIAP 2009*, pp. 404–413. Springer (2009)
20. Kandhalu, A., Rowe, A., Rajkumar, R., Huang, C., Yeh, C.-C.: Real-time video surveillance over IEEE 802.11 mesh networks. In: *Real-Time and Embedded Technology and Applications Symposium, 2009. RTAS 2009. 15th IEEE*, pp. 205–214. IEEE (2009)
21. Jafari, O.H., Mitzel, D., Leibe, B.: Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 5636–5643. IEEE (2014)
22. Munaro, M., Basso, F., Menegatti, E.: Tracking people within groups with RGB-D data. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2101–2107. IEEE (2012)
23. Munaro, M., Lewis, C., Chambers, D., Hvass, P., Menegatti, E.: RGB-D human detection and tracking for industrial environments. In: *Intelligent Autonomous Systems*, vol. 13, pp. 1655–1668. Springer (2016)
24. Sanders, J., Kandrot, E.: *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional (2010)
25. Ryoo, S., Rodrigues, C.I., Bagnsorkhi, S.S., Stone, S.S., Kirk, D.B., Hwu, W.-m.W.: Optimization principles and application performance evaluation of a multithreaded GPU using CUDA. In: *Proceedings of the 13th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pp. 73–82. ACM (2008)
26. Aldoma, A., Tombari, F., Prankl, J., Richtsfeld, A., Di Stefano, L., Vincze, M.: Multimodal cue integration through hypotheses verification for RGB-D object recognition and 6DOF pose

- estimation. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 2104–2111. IEEE (2013)
27. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: using kinect-style depth cameras for dense 3D modeling of indoor environments. *Int. J. Robot. Res.* **31**(5), 647–663 (2012)
 28. Carraro, M., Munaro, M., Menegatti, E.: Cost-efficient RGB-D smart camera for people detection and tracking. *J. Electron. Imaging* **25**(4), 041007–041007 (2016)

Influence of Stimulus Color on Steady State Visual Evoked Potentials

Leeyee Chu, Jacobo Fernández-Vargas, Kahori Kita and Wenwei Yu

Abstract Due to the low training time and high time resolution, steady state visual evoked potential (SSVEP)-based brain computer interfaces (BCIs) have been largely studied in recent years. The stimulus properties such as frequency, color and shape can greatly affect the performance, comfort and safety of the brain computer interfaces. Despite this fact, stimulation properties have received fairly little attention. This study aims to investigate the influence of stimulus color on SSVEPs. We tested 10 colors and did evaluation by using three different methods: the power amplitude, multivariate synchronization index, and canonical correlation analysis. The results showed that violet color had the least influence to SSVEPs while red color tended to have stronger impact.

Keywords SSVEP · BCI · EEG · Stimulus properties

1 Introduction

Brain computer interface (BCI) is a technology that allows communication between machines and the human brain. This technology allows people with neurodegenerative diseases or spinal cord injury to control robots such as wheelchairs and prosthetic devices by brain signals.

L. Chu (✉) · J. Fernández-Vargas
Department of Medical System Engineering, Chiba University, Chiba, Japan
e-mail: leeyee@chiba-u.jp; leeyeeee@gmail.com

J. Fernández-Vargas
e-mail: jacobofv@chiba-u.jp

K. Kita (✉) · W. Yu
Center for Frontier Medical Engineering, Chiba University, Chiba, Japan
e-mail: kkita@chiba-u.jp

W. Yu
e-mail: yuwill@faculty.chiba-u.jp

Studies about BCIs have shown promising results in recent years. Patients with complete hand paralysis were able to close and open a virtual hand by focusing their attention [1]. Performance of a mental spelling system, was enhanced by implementing a dual eye tracking/BCI system [2]. A wheelchair prototype was able to be controlled with high accuracies by brain signals [3]. Although the performance of BCIs is enhancing, more effort is needed to assure the stability and accuracy of BCIs so that they are applicable in real life. In addition, the performance of BCIs also rely on the physical conditions and physiological traits such as, concentration or surroundings between others [4]. One type of response that is relatively easy to detect is the Steady State Visual Evoked Potential (SSVEP) in electroencephalogram (EEG). SSVEP is a response that can be detected at visual field of cerebral cortex when a flickering light stimulates the retina. This response occurs at the same frequency as the flickering stimulus and, with lower intensity, in the harmonic and sub-harmonic frequencies [5]. SSVEP-based BCIs provide promising result because of their short training time, easy detection and accessibility for most people [6].

Although SSVEP-based BCIs have been studied in recent years, most studies focus on the signal processing techniques to improve the performance of BCIs. Despite the fact that parameters of stimulus properties such as shape, color and size have significant effects on the performance and safety of SSVEP-based BCIs [4], stimulus properties are given fairly little attention. It is important to study which parameters provide the most efficient BCI systems to optimize the performance of SSVEP-based BCIs. The main objective of this study is to study the influence of stimulus colors to SSVEP.

Few studies focus their attention on the effect of the color to the SSVEP response. Most studies were done only with a few colors (e.g. yellow, green, red, white, violet) [3, 4, 7] which we consider insufficient to conclude which color is more effective to SSVEPs. In addition, most studies about performance of stimuli colors on SSVEPs were done by displaying a multiple flickers simultaneously [3, 7]. Stimuli with larger size and closer to one another has detrimental effect on SSVEP performance of the target stimulus [4]. The studies of [8] showed that inter-stimulus distance is significantly different with SSVEP performance. This might due to interference of neighboring flickers on the same visual field and this may cause users to lose their attention. To tackle these problems, we investigated the influence of wavelength of colors and their respective color luminance to SSVEPs with nine colors from the visible spectrum (380–735 nm) and white color, which in total ten colors. Our study focuses on the sheer SSVEP response to each color, therefore, we displayed only one visual stimulus during the stimulation of each color.

2 Materials and Methods

2.1 Participants

Fifteen healthy subjects (nine males, six females) participated in our experiment. All subjects have normal or corrected-to-normal vision and did not suffer from epilepsy. The mean age is 23.9 years old in the range of 19–25 years old. Permission of the ethics committees of Chiba University Graduate School and Faculty of Engineering was obtained. All subjects participated voluntarily with informed consent and were informed that they could leave the experiment if they experience discomfort or epilepsy during the experiment.

2.2 Stimulation

We used a 10 Hz stimulation frequency. The flickering stimulus was generated using a laptop through a control program written in C++ running on Ubuntu 14.04.3 LTS. A 22 in. monitor screen (LG W2242TQ-BF), a 60 Hz refresh rate and a resolution of 1680 × 1050 pixels adjusted to its brightest state (300 cd/m²) was used to present the stimulus. Colors were chosen within the range of visible spectrum from 390–735 nm as shown in Table 1. Within the range, we intended to select nine colors which have equal difference in wavelength from one another, which is 40 nm. However, some colors appear to have no difference due to the limitations of our monitor display. In that case, we selected colors which have difference within 25 nm from 40 nm and appear differently from the previous color. Note that the colors we obtained are of approximation. Luminance of the color was

Table 1 Color information

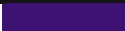
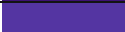







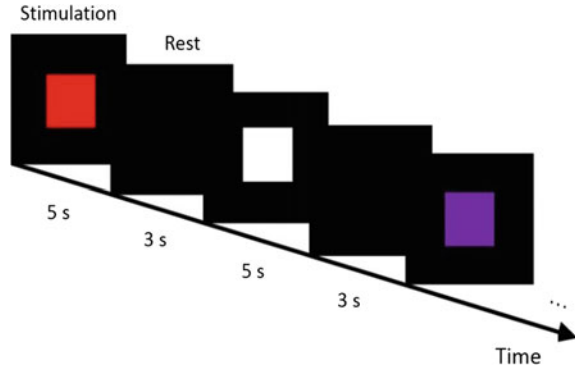
	Color	Wavelength (nm)	RGB	Luminance
1		390	(101,0,121)	30.21
2		430	(43,0,255)	27.55
3		470	(0,153,255)	127.84
4		510	(0,255,0)	182.38
5		550	(146,255,0)	213.42
6		600	(255,177,0)	180.80
7		630	(255,59,0)	96.41
8		670	(255,0,0)	54.21
9		735	(177,0,0)	37.63
10		-	(255,255,255)	255.00

Fig. 1 Stimulation sequences in experiment



calculated respectively according to Recommendation ITU-R BT.709-6 [9], under consideration that our monitor display uses a sRGB color space.

$$\text{Luminance} = 0.2126R + 0.7152G + 0.0722B \quad (1)$$

The experiment was carried out in a dark room, with the subjects seated approximately 70 cm from the center of the monitor screen. Subjects were asked to ocular or muscular movements during visual stimulation. A stimulus with a size of 500×500 pixels was presented at the center of monitor display during stimulation. Every color was presented for 5 s with a 3 s resting time in between without any stimulus (Fig. 1). The process was repeated for the 10 colors in random order. Stimulating the 10 colors take 80 s. This process was repeated 10 times for every subject.

2.3 EEG Measurement

We acquired EEG by Biosemi ActiveTwo system (Biosemi, Netherlands) with 16 active channels. The location of the electrodes was chosen to primarily cover the parietal and the occipital area. The electrode positions were: T7, T8, C3, C4, Cz, P1, P2, P3, P4, Pz, PO3, PO4, POz, O1, O2 and Oz according to the international 10–20 system. The signal acquisition frequency of the system was 1024 Hz.

2.4 Signal Processing

Previous study found that detection of SSVEP is better by using two bipolar channels (e.g. PO3-O1) than by using monopolar channels (e.g. O1, O2 etc.) [10]. Therefore, although 16 EEG channels were recorded we used only channels POz and Oz (POz-Oz) for the signal processing. They were filtered with by a 4th order

Butterworth high pass filter with cutoff frequency of 1 Hz. Then they were divided into 1 s segment with 87.5 % overlap. For every segment a detrending and a hamming window was applied. Next, we evaluated SSVEP with SSVEP power amplitude, multivariate synchronization index (MSI), and canonical correlation analysis (CCA). All the analysis were performed offline.

SSVEP Power Amplitude

For every segment of the pre-processed data the corresponding FFT was calculated. The highest power amplitude for the 10 Hz frequency, during stimulation of every color for every repetition was selected. In total we had 10 values for each one of the 10 colors for each subject, i.e. 100 values/subject.

MSI

MSI is a novel classification method for frequency recognition in SSVEP. MSI is used to estimate the synchronization between the EEG signals and the reference signal [11], which in our case, generated from the sine and cosine components of the 10 Hz stimulus. Assume that the EEG signals are denoted by a matrix X of size $N \times M$, and the reference signal by a matrix Y of size $2N_h \times M$. N is the number of channels, M is the number of samples and N_h is the number of harmonics for the sine and cosine components. In our case, we used only channel POz and Oz, therefore $N = 2$. We also divided the data into 1 s segments, corresponding to our sampling frequency which is 1024 Hz, $M = 1024$. We used only one harmonic frequency therefore $N_h = 1$. Next, a correlation matrix is calculated as

$$c = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \tag{2}$$

where

$$C_{11} = \frac{1}{M} XX^T \tag{3}$$

$$C_{22} = \frac{1}{M} YY^T \tag{4}$$

$$C_{12} = C_{21}^T = \frac{1}{M} XY^T \tag{5}$$

To remove the internal correlation structure of X and Y which is irrelevant to the stimulus frequency, a linear transformation is implemented [12].

$$U = \begin{bmatrix} C_{11}^{-\frac{1}{2}} & 0 \\ 0 & C_{22}^{-\frac{1}{2}} \end{bmatrix} \tag{6}$$

Then, the transformed correlation matrix R is calculated.

$$R = UCU^T = \begin{bmatrix} I_{N \times N} & C_{11}^{-\frac{1}{2}} C_{12} C_{22}^{-\frac{1}{2}} \\ C_{22}^{-\frac{1}{2}} C_{21} C_{11}^{-\frac{1}{2}} & I_{2N_h \times 2N_h} \end{bmatrix} \quad (7)$$

where $I_{N \times N}$ is the identity matrix of dimension N , and $I_{2N_h \times 2N_h}$ is the identity matrix of dimension $2N_h$. Then, normalized eigenvalues are calculated from the eigenvalues, $\lambda_1, \lambda_2, \dots, \lambda_P$ of matrix R .

$$\lambda'_i = \frac{\lambda_i}{\sum_{i=1}^P \lambda_i} = \frac{\lambda_i}{tr(R)} \quad (8)$$

where $P = N + 2N_h$. Then the synchronization index between the EEG signals and the reference signal can be calculated.

$$\text{Synchronization index} = 1 + \frac{\sum_{i=1}^P \lambda'_i \log(\lambda'_i)}{\log(P)} \quad (9)$$

Consider that there is only one stimulus frequency, which is 10 Hz in our experiment, the reference signal is calculated as follows:

$$Y_i = \begin{bmatrix} \sin(2\pi \cdot f \cdot t_i) \\ \cos(2\pi \cdot f \cdot t_i) \\ \vdots \\ \sin(2\pi \cdot N_h \cdot f \cdot t_i) \\ \cos(2\pi \cdot N_h \cdot f \cdot t_i) \end{bmatrix} = \begin{bmatrix} \sin(2\pi \cdot 10 \cdot t_i) \\ \cos(2\pi \cdot 10 \cdot t_i) \end{bmatrix}, t_i = \frac{i}{F_s} \quad (10)$$

$$Y = [Y_1 Y_2 \dots Y_i], i = 1, 2, \dots, M \quad (11)$$

where F_s is the sampling frequency, which is 1024 Hz. Note that $N_h = 1$ as aforementioned. Synchronization index is used as an indication of how well the EEG signals synchronized with the stimulus and we named it as MSI value in the following sections. In each repetition, maximum MSI value was selected for every color.

CCA

CCA is a method for SSVEP frequency recognition, which is a multivariable method to compute the correlation between two sets of data. It is widely used in studies to extract frequency features for multichannel SSVEP-based BCI [13–15]. First, assume that we have x_1, x_2, \dots, x_i which are signals from i number of EEG channels, and y_1, y_2, \dots, y_j which are signals from j number of reference signals as calculated in Eqs. (10) and (11). Two data sets, X and Y with linear combinations $w_{x_1} x_1, \dots, w_{x_i} x_i$ and $w_{y_1} y_1, \dots, w_{y_j} y_j$ were calculated respectively as follows:

$$X = w_{x_1}x_1 + \dots + w_{x_i}x_i = w_x^T x \quad (12)$$

$$Y = w_{y_1}y_1 + \dots + w_{y_j}y_j = w_y^T y \quad (13)$$

where w_{x_1}, \dots, w_{x_i} and w_{y_1}, \dots, w_{y_j} are the linear combination coefficients. The CCA finds the linear combination coefficients which maximize the correlation between X and Y. The correlation coefficient is then calculated as follows:

$$\rho = \frac{w_x^T x y^T w_y}{\sqrt{w_x^T x x^T w_x w_y^T y y^T w_y}} \quad (14)$$

CCA is discussed more in detail by [13, 14]. Similar as in SSVEP power amplitude and MSI, the maximum value of CCA coefficient for every color was selected in every repetition.

Statistics

After extracting the maximum value of each color for every repetition by all three methods, we examined correlation of SSVEP and color luminance, and correlation of SSVEP and colors (wavelength). To examine the correlation of SSVEP and color luminance, first we calculated the mean values of all the repetitions for every color and subject. As we have fifteen subjects, there are 15 mean values for each method. Finally, the mean of these 15 values was calculated for every color. Then, we computed the Pearson's correlation coefficient to examine the relationship between SSVEP and luminance. The significance threshold for the analysis was assigned as $p < 0.05$. About correlation of SSVEP and colors (wavelength), we compared the influence of colors on SSVEP by Friedman test followed by Tukey-Kramer post hoc test. The significance threshold for Friedman test was assigned as $\chi^2 < 0.05$.

3 Results

3.1 Correlation of SSVEP and Luminance

Figure 2 shows correlation between SSVEP and color luminance by (a) SSVEP power amplitude, (b) MSI, and (c) CCA. The colors of the dots represent the stimulation colors. By SSVEP power amplitude, the correlation coefficient, $r = 0.2921$, $p = 0.4128$. By MSI, $r = 0.5698$, $p = 0.0855$. By CCA, $r = 0.5577$, $p = 0.0939$. From the results, p by all methods are higher than the significance level of 0.05, showing that SSVEP is not significantly related to luminance.

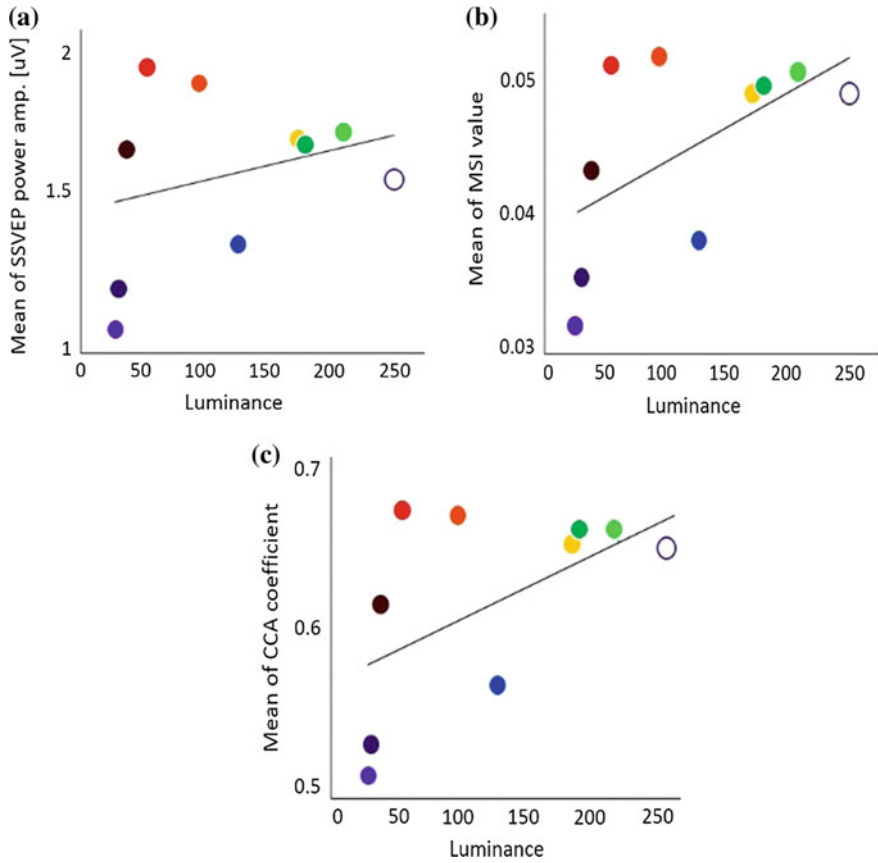


Fig. 2 Correlation of SSVEP and luminance, **a** SSVEP power amplitude, **b** MSI, **c** CCA

3.2 Correlation of SSVEP and Colors (Wavelength)

Figure 3 shows the results of such comparison by (a) SSVEP power amplitude, (b) MSI, (c) CCA. The colors of the bars represent the stimulation colors. With all the methods we used, color 2 is significantly different with other 5 colors (color 4, 5, 6, 7 and 8). The means of color 1 and color 3 are also less than the other colors although they are not statistically significant different except for SSVEP power amplitude that color 1 is significantly different with color 4, 5, 6, 7 and 8. Although all the results appear to be similar, result with SSVEP power amplitude is slightly different with the other two methods. With SSVEP power amplitude, color 7 (orange) and color 8 (red) appear to have higher mean compared to other colors.

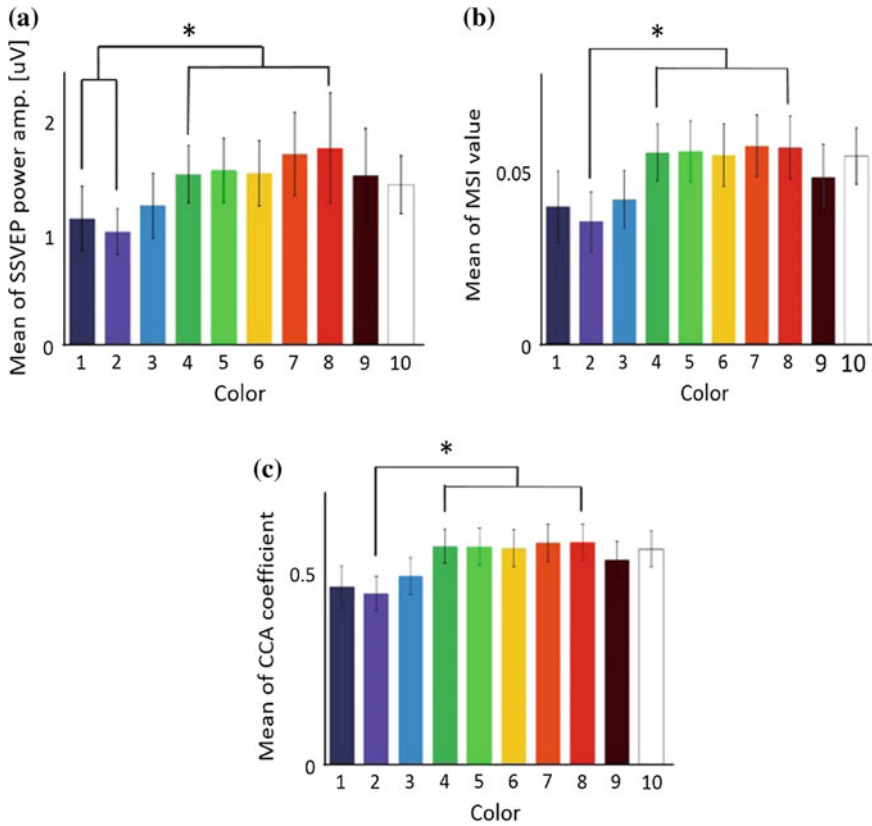


Fig. 3 Error bars with means and 95 % confidence intervals (* $\chi^2 < 0.05$ Friedman test followed by a Tukey-Kramer post hoc test), **a** SSVEP power amplitude, **b** MSI, **c** CCA

4 Discussion

We investigated the relation between SSVEP and color and found that violet color has the worst SSVEP response. Our results contradicted with [3], who showed that violet color has higher accuracy than that with green, red and blue color. They displayed four stimuli simultaneously with four different frequencies and tested the accuracy to control a wheelchair prototype, while our experiment simply tested SSVEP response by displaying only one stimulus each time. The difference on the results may arise from the use of multiple colors at the same time, as well as the use of different frequencies.

Colors with longer wavelength such as color 7 (orange color) and color 8 (red color) appear to have higher mean compared to other colors, although the differences are not significant. Since looking at a flickering stimulus and generating SSVEPs requires high visual attention, it is recommended to use colors that may

guide attention. The studies of [16] are coherent with our results that warm colors such as red and orange are more likely to guide attention. It is no doubt that red color grabs attention and stimulates visual sense considering its usage on symbolic sign like “dangerous” and “attention” to draw people attention. In this study the results also show that red color might have a great effect on SSVEPs, which are considered as a type of attention-related brain signals. The result is supported by the studies of [17] about response of red color in human eyes, by showing that red color had the prominent amplitude peak compared to blue and yellow color. Red color also provides the highest rates of accuracy for detection of SSVEP as showed by [7]. Our similar results indicate that by displaying only one stimulus each time could be helpful in preventing interference by neighboring flickers.

In our study, it can be concluded that colors with shorter wavelength like violet and blue have relatively worse SSVEP response compared to other colors. On the other hand, colors with longer wavelength like orange and red tend to have better SSVEP response. However, our study investigated the sheer SSVEP response to colors and only offline experiment was done. As SSVEP-based BCIs are involved in real life application, online experiment is necessary to give further insights about our results. As our experiment displayed only one stimulus each stimulation, displaying multiple stimuli with different colors simultaneously might be the future work to further investigate if it supports our current results. Besides, different stimulation frequencies might be tested to study the correlation between frequency and color, and their effect on SSVEPs. In addition, questionnaire will be carried out at the end of the experiment to study the comfort level of each color in each subject.

The future study is expected to provide us more understanding and convincing result to the influence of colors to SSVEPs.

References

1. Foldes, S., Weber, D., Collinger, J.: MEG-based neurofeedback for hand rehabilitation. *J. Neuroeng. Rehabil.* **12**(1), 1–9 (2015)
2. Lim, J., Lee, J., Hwang, H., Kim, D., Im, C.: Development of a hybrid mental spelling system combining SSVEP-based brain–computer interface and webcam-based eye tracking. *Biomed. Signal Process. Control* **21**, 99–104 (2015)
3. Singla, R., Khosla, A., Jha, R.: Influence of stimuli color on steady-state visual evoked potentials based BCI wheelchair control. *J. Biomed. Sci. Eng.* **06**(11), 1050–1055 (2013)
4. Bieger, J., Molina, G.G., Zhu, D.: Effects of stimulation properties in steady-state visual evoked potential based brain–computer interfaces. In: Buenos Aires, Argentina: 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2010)
5. Herrmann, C.: Human EEG responses to 1–100 Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. *Exp. Brain Res.* **137**(3–4), 346–353 (2001)
6. Jasper, H.H.: Report of the committee on methods of clinical examination in electroencephalography. *Electroencephalogr. Clin. Neurophysiol.* **10**(2), 370–375 (1958)
7. Tello, R., Müller, S., Ferreira, A., Bastos, T.: Comparison of the influence of stimuli color on steady-state visual evoked potentials. *Res. Biomed. Eng.* **31**(3), 218–231 (2015)

8. Ng, K., Bradley, A., Cunnington, R.: Stimulus specificity of a steady-state visual-evoked potential-based brain–computer interface. *J. Neural Eng.* **9**(3), 036008 (2012)
9. Parameter values for the HDTV standards for production and international programme exchange (2015). Geneva: International Telecommunication Union, p. 6
10. Müller, S., Bastos-Filho, T., Sarcinelli-Filho, M.: Monopolar and bipolar electrode settings for SSVEP-based brain-computer interface. *J. Med. Biol. Eng.* **35**(4), 482–491 (2015)
11. Zhang, Y., Xu, P., Cheng, K., Yao, D.: Multivariate synchronization index for frequency recognition of SSVEP-based brain–computer interface. *J. Neurosci. Methods* **221**, 32–40 (2014)
12. Carmeli, C., Knyazeva, M., Innocenti, G., De Feo, O.: Assessment of EEG synchronization based on state-space analysis. *NeuroImage* **25**(2), 339–354 (2005)
13. Lin, Z., Zhang, C., Wu, W., Gao, X.: Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs. *IEEE Trans. Biomed. Eng.* **54**(6), 1172–1176 (2007)
14. Friman, O., Cedefamn, J., Lundberg, P., Borga, M., Knutsson, H.: Detection of neural activity in functional MRI using canonical correlation analysis. *Magn. Reson. Med.* **45**(2), 323–330 (2001)
15. Bin, G., Gao, X., Yan, Z., Hong, B., Gao, S.: An online multi-channel SSVEP-based brain–computer interface using a canonical correlation analysis method. *J. Neural Eng.* **6**(4), 046002 (2009)
16. Pal, R., Mukherjee, J., Mitra, P.: How Do Warm Colors Affect Visual Attention? Mumbai, India: The Eighth Indian Conference on Computer Vision, Graphics and Image Processing (2012)
17. Regan, D.: An effect of stimulus colour on average steady-state potentials evoked in man. *Nature* **210**(5040), 1056–1057 (1966)

Accelerated Adaptive Local Scanning of Complicated Micro Objects for the PSD Scanning Microscopy: Methods and Implementation

Mehdi Rahimi and Yantao Shen

Abstract A PSD (Position Sensitive Detector) -based microscopy was introduced previously by same authors to scan sophisticated objects using the PSD sensor and a laser beam connected to an X-Y table (Rahimi et al., 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp 1685–1690 (2014) [1]; Rahimi et al., 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 4955–4960 (2015) [2]). The system has numerous capabilities including but not limited to tracing an unknown object on the PSD, finding the dimensions of the object and adaptive local scanning for objects with intersections and bifurcations. Although promising results were obtained from that system, the speed of the scanning was still dependent on the speed of the movement of the micromanipulator. This work presents an extension to the tracing method by scanning tree-shaped objects. The simulations show a very successful scanning that can map any tree-shaped object.

Keywords PSD · Adaptive scanning · Local scanning · Micro objects

1 Introduction

In the previous works [1, 2], we developed a comprehensive and adaptive local scanning algorithm that is capable of rapidly scanning indiscrete, one-piece objects with various shapes such as open or closed curves, loops, intersections or bifurcations and so on. That proposed method does not require any prior information about the scanned objects nor does it need any preparation step. Instead, all the points

M. Rahimi · Y. Shen (✉)

Department of Electrical and Biomedical Engineering, University of Nevada, Reno,
1664 N. Virginia St., Reno, NV 89557, USA
e-mail: ytshen@unr.edu

M. Rahimi

e-mail: mrahimi@unr.edu

are scanned on-line during the scanning. It uses an effective initial search pattern, Start-6, until one point of the object is found and then maps a sinusoidal scanning pattern to scan the object and then the frequency and amplitude of this sine wave would be adaptively adjusted based on predicting the curvature of the object. An Archimedean spiral search pattern is also used for missed scanning. In addition, the developed algorithms can find intersections and bifurcations on the object, and then resume the scanning from these points. To minimize the object scanning error, a Fourier curve fitting was employed to improve the scanning accuracy. Extensive implementation results demonstrated that the fitting was able to reduce the scanning error to almost half. It also validated that our comprehensive and adaptive local scanning method can shorten the scanning time in order of hundreds of times in comparison to the traditional raster scanning without losing any important information about the object.

In this work, we focused on objects with more complicated bifurcations. Namely, objects that have many branches and do not necessarily have any closed loop. These tree-shaped objects can be seen in branches of vein anatomy, dendrites and axon terminals of neurons and many other similar shapes.

Scanning these tree-shaped objects requires an intelligent algorithm that differentiates between the base and the branches from each other and be able to find and follow the correct path to scan the whole object. Considering the limitations of the PSD [3–5]. This was achieved by using a stack-based algorithm that maps the objects as it goes through it. Although promising results were obtained from this, but there would still be some cases that would have a missed branch due to the high angle of the path between a base and a branch. To be sure that these problems are addressed, a Logarithmic spiral search is initiated after the algorithm finishes its normal scanning. This final scan would start from the center of the scanned object and would eventually find any missed branches.

In the last part of this work, we use a parametric Fourier curve fitting to do one more processing step on the acquired scanned points. This would help in eliminating the shadow effect that might influence the scanning pattern.

This paper is structured into the following sections: In Sect. 2, the scanning method for the tree-shaped object is introduced and the simulation results of the scanning would be presented. Section 3 shows the validation methods and results that we used to examine the scanning algorithm. In Sect. 4 we conclude the work.

2 Scanning Tree-Shaped Objects

The scanning of the tree-shaped objects needs a special algorithm since the scanning pattern can get so complicated that might miss a considerable portion of the object. An example of a complicated tree-shaped object is shown in Fig. 1. Note that we are not necessarily going to start the scanning from the base. The scanning might start from any point of the object depending on how the object is positioned on the PSD. In this particular situation, since the object is near the starting point of the initial

Fig. 1 A tree-shaped object is shown. The object has many branches. Note that we are not necessarily going to start the scanning from the base

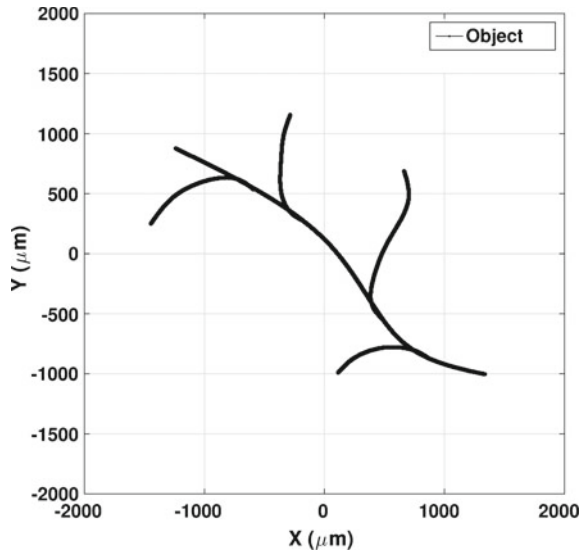
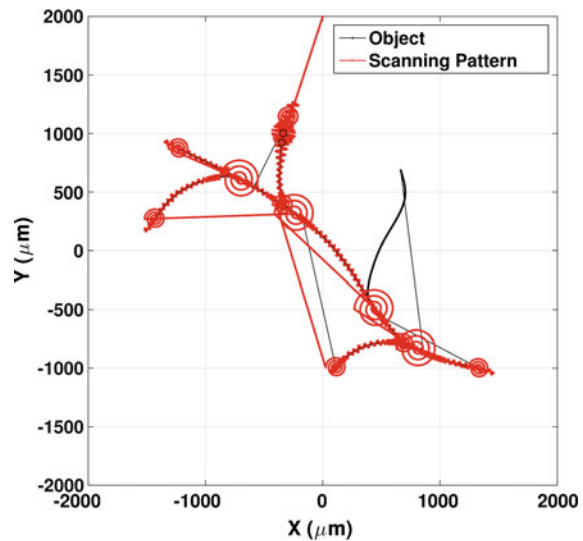


Fig. 2 An unsuccessful scan is shown here. Color *black* represents the object and *red* is the scanning pattern. (Total passes over the object were 337 times and the total time is 338.9 times faster than traditional scanning time)



scanning pattern, one of the branches is the first contact of the scanning pattern and the object. After going through the scanning algorithm the final result can be seen in Fig. 2. At it can be seen, one of the branches is missed in the scanning.

To get to a comprehensive scanning method for tree-shaped objects, at first, we applies a few modifications to the section of the algorithm that detects and marks the bifurcations. By this, there would be much less chances of missing a branch but it still does not guarantee a definite fail-proof solution.

The challenge here is that since there is no prior information of the object, there might be cases that the branch is exactly perpendicular to the base. In such a scenario, a sinusoidal scanning pattern has a fifty percent chance of catching the branch depending upon which side of the sine wave is toward the branch at the exact moment that it gets to that location. If the scanning pattern does not meet the branch, there would be absolutely no solution to extract that information later from the data.

A couple of methods was examined to prevent this problem. One approach might be to scan the whole object with a very low frequency sine wave after finishing the normal scanning. This can be very quick and also can reduce the possibility of missing a branch significantly but it is still not a definite solution.

Another solution would be to do a complete go-around search after finishing the scanning. In this approach, a scanning line would simply go around the object trying to constrain it in the smallest area that is possible. Since a not-scanned branch would be out of this border, it guarantees to not miss any branches. The problem with this method is that it is very hard to implement and verify.

The final approach is to add a final spiral search and continue the scanning from the newly found points. This method is very fast and also easy to implement but it can introduce one serious challenge where the final spiral scan might find a branch that includes bifurcations itself. In another word, the new branch might be connected to a whole new group of branches that were not scanned before and therefore a missed branch would be a possibility after going through that group.

To address this problem, the spiral search would continue until it exceeds the furthest recognized and scanned point of the object.

To define a spiral search, two options were available. One option was to use an Archimedean spiral and the other to use a Logarithmic spiral. Figure 3 shows the final Archimedean spiral scanning that is used with the same object as in Fig. 2.

A Logarithmic spiral can be defined in polar coordinates (r, θ) as:

$$\theta = \frac{1}{b} \ln\left(\frac{r}{a}\right) \quad (1)$$

With e being the base of natural logarithm, a and b being arbitrary positive real constants. Logarithmic spirals have been studied extensively by Miura [6].

In parametric form, the curve is defined as:

$$x(t) = r(t) \cos(t) = ae^{bt} \cos(t) \quad (2)$$

$$y(t) = r(t) \sin(t) = ae^{bt} \sin(t) \quad (3)$$

With real numbers a and b which would define the shape of the spiral.

To evaluate which spiral is more suitable for this specific situation, a simulation program was designed in MATLAB[®] and ran with many different objects. The results show some improvements in some cases and almost the same results in other cases. A small sample of these results are presented in Table 1.

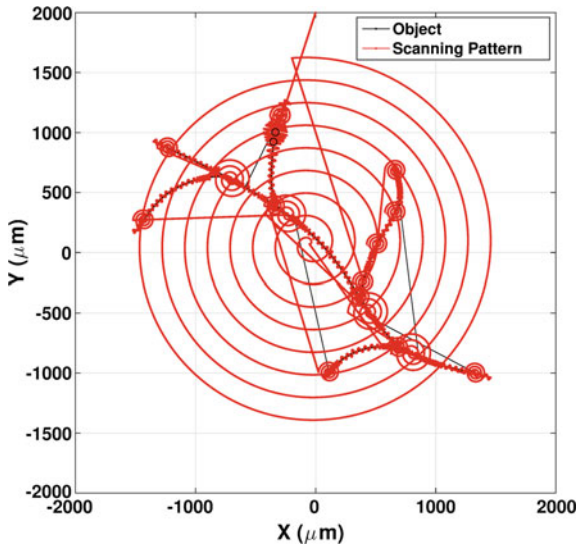


Fig. 3 The same object as Fig. 2 but after all the modifications in the program and applying the final spiral search. (Total passes over the object were 422 times and the total time is 226.0 times faster than traditional scanning time)

Table 1 Comparison between different spirals scanning time. (“Archimedean time” is the “Scanning time with Archimedean spiral” and “Logarithmic time” is the Scanning time with Logarithmic spiral”)

Sample	Archimedean time (s)	Logarithmic time (s)	Improvement (%)
First sample	142	131	7.7
Second sample	177	176	0.5
Third sample	208	196	5.8

It should be noted that the simulation of others samples yielded almost same results as what is presented here.

It can be concluded that using a Logarithmic spiral would improve the scanning time for an average of almost 5% but its other advantages and disadvantages are not clear and to be discussed. It should be noted that the a and b parameters of the Logarithmic spiral have a significant impact on the shape of it. In these experiments $a = 50$ and $b = 0.07$ where selected by trial and error. Figure 4 shows the Logarithmic spirals scanning of the same object that was used in Fig. 2. The result of this scanning is shown in Fig. 5.

Another example of a very complicated object is presented in Fig. 6. The different between this object and the previous one is that it has nested branches. In another word, there are branches that they include other branches themselves. Although it seems that this is too complicated to scan but by applying the suggested algorithm

Fig. 4 The same object as Fig. 2 is scanned with logarithmic spiral pattern. The scanning time is improved by 7.7%. (Total Passes over the object were 419 times and the total time is 243.7 times faster than traditional scanning time)

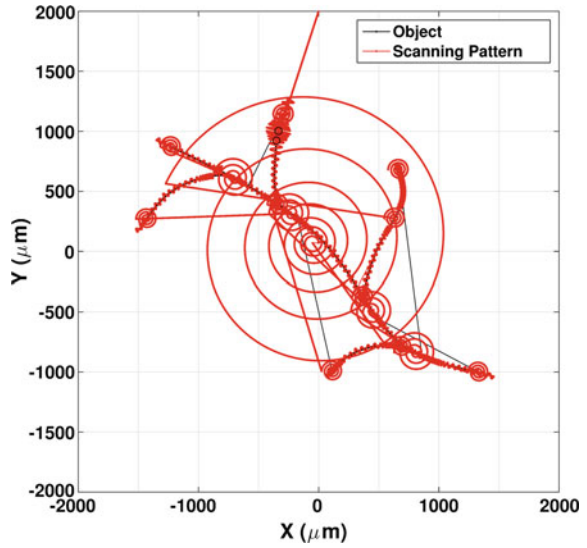
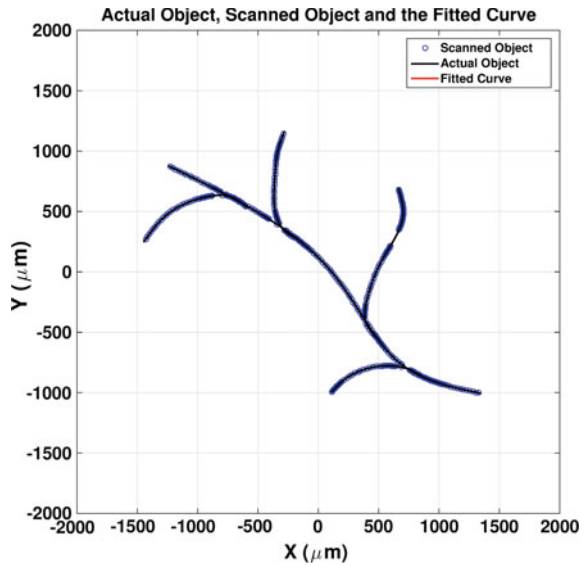


Fig. 5 This is the acquired scanned points from the logarithmic scanning that was shown in Fig. 4



we can scan an object even as complicate as this one. The scanning result is presented in Fig. 7. The acquired scanned points of is shown in Fig. 8. As it can be seen, all the branches are scanned perfectly.

Fig. 6 A very complicated sample is presented here. The result of the scanning is shown in Fig. 7

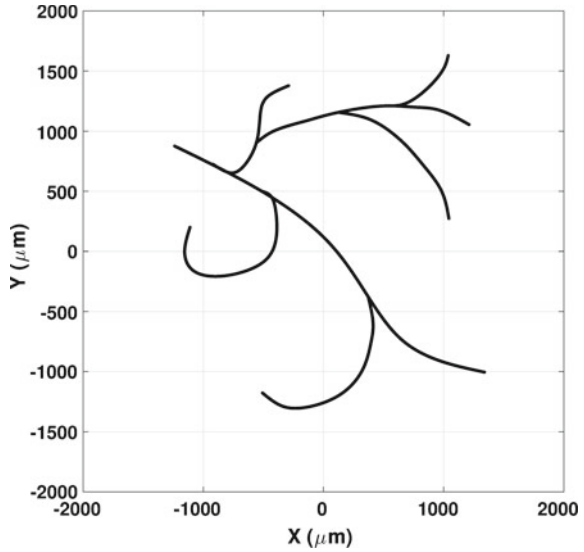
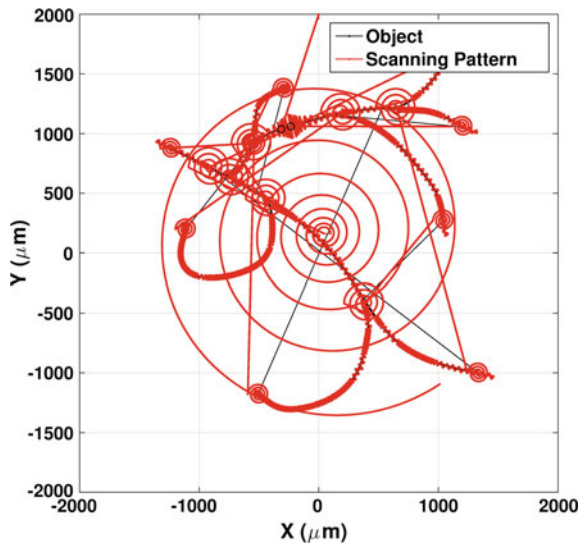


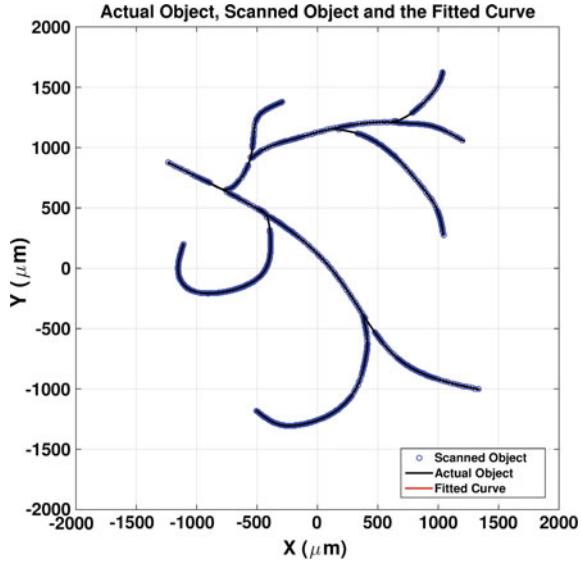
Fig. 7 The complicated object of the Fig. 6 is scanned here. The result of the scanning can be seen with red lines. (Total Passes over the object were 851 times and the total time is 163.2 times faster than traditional scanning time)



3 Verification of the Results

As the scanning pattern moves along the object, it would find points from the object that are on the outer edges of the object. To get the actual object which is in between these scanned points, a parametric Fourier curve fitting with 6 terms was used as described in previous works [2]. The Fourier equation was used as shown in the following equation:

Fig. 8 This is the acquired scanned points from the Logarithmic scanning that was shown in Fig. 7



$$y = a_0 + \sum_{i=1}^6 a_i \cos(n\omega x) + b_i \sin(n\omega x) \tag{4}$$

where these coefficients a_0 , a_i , and b_i in the equation are determined by the fitting function. Any number of terms more than this would result in over-fitting [7].

In addition, as the whole scanned points are obtained by a lot of sudden jumps to different positions -like scanning each side of the object or resuming the scanning for intersections and bifurcations- it is not possible and reliable for all the scanned points to be fitted at once. To solve this, the scanned points are divided into small parts by considering the jumps and sudden changes in directions. Each part is fitted separately and the final result is obtained by simply putting these pieces together.

An example of the fitting for the object in Fig. 1 is shown in Fig. 9. In this particular situation the mean of the differences between the scanned points and the actual object is $0.307 \mu\text{m}$. This is shown with red circles in the Fig. 9. After using the fitting, the mean of the differences between the fitted points and the actual object is $0.229 \mu\text{m}$.

We can see an almost same results for the other scanings. For example the differences and the fitting results for the object in Fig. 6 can be seen in Fig. 10. In this example, the mean of the differences between the scanned points and the actual object is $0.317 \mu\text{m}$. After using the fitting, the mean of the differences between the fitted points and the actual object is $0.179 \mu\text{m}$. This shows that the accuracy has been almost doubled.

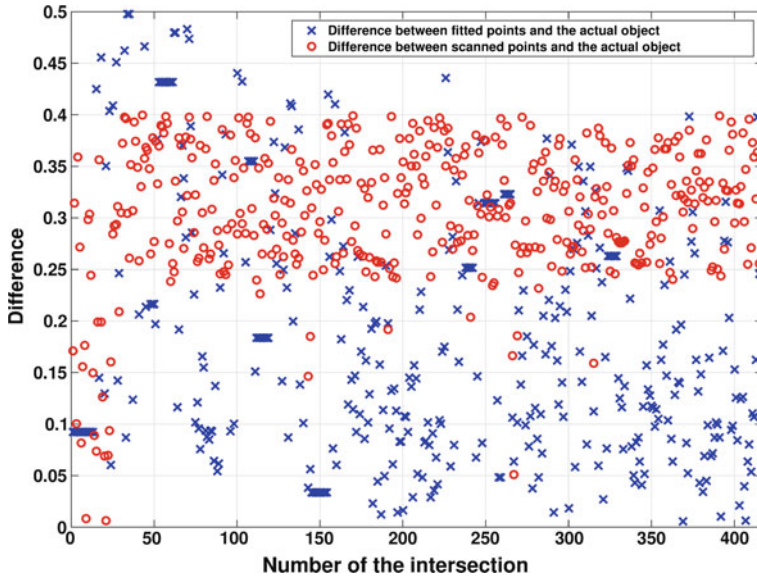


Fig. 9 This is the fitting result for the object in Fig. 1. The differences given here are in scale of micrometers. The mean of the differences between the scanned points and the actual object is $0.307 \mu\text{m}$. The mean of the differences between the fitted points and the actual object is $0.229 \mu\text{m}$

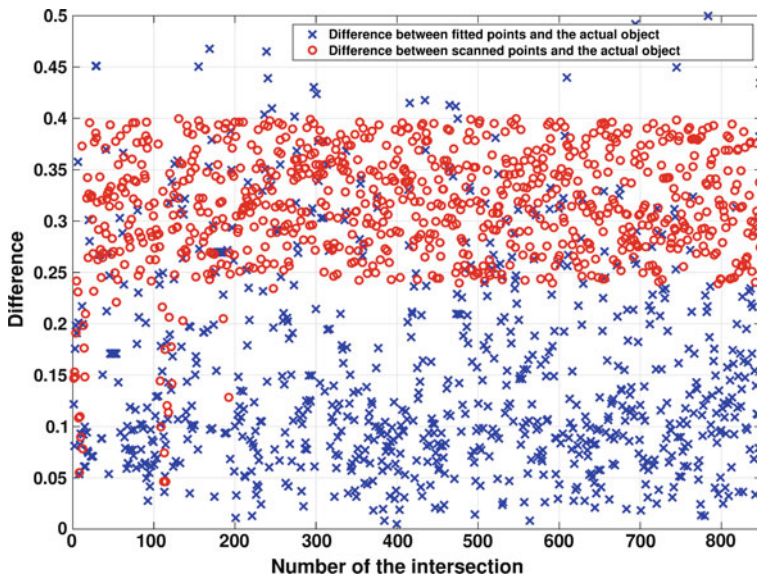


Fig. 10 This is the fitting result for the object in Fig. 6. The differences given here are in scale of micrometers. The mean of the differences between the scanned points and the actual object is $0.317 \mu\text{m}$. The mean of the differences between the fitted points and the actual object is $0.179 \mu\text{m}$

This fitting has a coefficient of determination of and only 16 points couldn't be fitted. Also, the object is defined in a square scanning area of $4000 \times 4000 \mu\text{m}$. The mean difference of $0.179 \mu\text{m}$ between final results and the actual object shows the very high accuracy of the scanning algorithm and the fitting.

4 Conclusion

This work presents our recent improvements of the previously proposed PSD Microscopy.

In the first part of the work, we extended the research to scan tree-shaped objects that have many branches. To achieve this purpose, we improved the required algorithm and also introduced a final spiral search to find any missed branches after finishing the main scanning. To decide on which spiral shape we need to use, a comparison between Archimedean and Logarithmic spiral was done to make sure which one is more suitable for this particular problem. At the end and after the simulations, we chose the Logarithmic spiral because of the slight improvement in the scanning time that it presents. The algorithm can now scan any tree-shaped object that has many branches. This was shown with a very complicated object and as a result the algorithm could scan the whole object flawlessly.

In the verification section of the research, the differences between the actual object and the scanned object was shown. Finally to minimize the object scanning error, a Fourier curve fitting is employed to improve the scanning accuracy. Extensive implementation results demonstrate that the fitting can reduce the scanning error to an almost half value. Two examples of this reduction in scanning error were shown in the verification section.

Future works include extensive experiments to verify the performance of this system on different objects and also using other kinds of PSDs with larger areas to scan bigger objects and finally finding any problems that might be left with the microscopy system.

References

1. Rahimi, M., Luo, Y., Harris, F.C., Dascalu, S.M., Shen, Y.: Improving measurement accuracy of Position Sensitive Detector (PSD) for a new scanning PSD microscopy system. In: 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1685–1690. IEEE (2014)
2. Rahimi, M., Shen, Y.: Adaptive local scanning: a comprehensive and intelligent method for fast scanning of indiscrete objects. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4955–4960. IEEE (2015)
3. Hamamatsu Selection Guide.: Position Sensitive Detectors. Solid State Division. Hamamatsu. (2008)
4. OSI Optoelectronics. Duo-Lateral PSD, Super Linear PSDs manual. www.osioptoelectronics.com, p. 48
5. OSI Optoelectronics. PSD Characteristics. www.osioptoelectronics.com. Technical Statement

6. Miura, K.T.: A general equation of aesthetic curves and its self-affinity. *Comput.-Aided Design Appl.* **3**(1–4), 457–464 (2006)
7. Faber, N.M., Rajko, R.: How to avoid over-fitting in multivariate calibration. The conventional validation approach and an alternative. *Anal. Chim. Acta* **595**(1), 98–106 (2007)

Part IX
Medical Engineering

Development of a Robotic Thumb Rehabilitation System Using a Soft Pneumatic Actuator and a Pneumatic Artificial Muscles-Based Parallel Link Mechanism

Kouki Shiota, Tapio V.J. Tarvainen, Masashi Sekine, Kahori Kita and Wenwei Yu

Abstract The main function of a hand is to grasp and manipulate objects, and the thumb contributes most to this function. However, thumb rehabilitation devices, especially soft robotic gloves, have not been widely investigated. Soft pneumatic actuators are lighter, more flexible, and easier maintenance than other actuators. This makes them safer and more cost-effective. In this paper, we present a design for a soft robotic thumb rehabilitation system. We used a soft pneumatic actuator for bending (SPAB) for flexion-extension motion. For the carpometacarpal (CMC) joint, we used a parallel link mechanism by placing pneumatic artificial muscles (PAM) around it in three directions. We evaluated whether this setup of SPAB and PAM could enable the required three-dimensional thumb motions by using a prototype system on a dummy thumb. The results showed that the proposed mechanism enables opposition, abduction, and adduction motions for the thumb.

Keywords Hand rehabilitation · Soft pneumatic robotics · Power assist

K. Shiota (✉) · T.V.J. Tarvainen (✉)
Graduate School of Engineering, Chiba University, Chiba, Japan
e-mail: acaa2343@chiba-u.jp

T.V.J. Tarvainen
e-mail: tapio.tarvainen@chiba-u.jp

M. Sekine (✉) · K. Kita (✉) · W. Yu (✉)
Center for Frontier Medical Engineering, Chiba University, Chiba, Japan
e-mail: sekine@office.chiba-u.jp

K. Kita
e-mail: kkita@chiba-u.jp

W. Yu
e-mail: yuwill@faculty.chiba-u.jp

1 Introduction

In Japan, there are over 1.17 million stroke patients. The most common impairments after stroke are motor deficits, such as hemiparesis, which is experienced by approximately 80 % of stroke survivors [1]. These patients experience either partial or total absence of hand motor function, which can considerably reduce quality of life, because of major restrictions on activities of daily living (ADL) and ability to work.

Rehabilitation is provided under the direction and assistance of an occupational therapists or a physiotherapists, in order to have the patients return to normal life, and to improve their ability to perform ADL adequately. Especially repetitive task practice has been shown to be effective in improving motor function after stroke [2, 3]. However, hand rehabilitation is labor intensive and costly due to the required long hours and terms of training with a therapist. In order to deal with this problem, many robotic hand rehabilitation systems have been investigated recently to automate some of the therapy [2, 4, 5].

In recent years, robotic rehabilitation devices that use soft fluidic actuators with high level of elasticity have been studied [4, 6, 7]. Soft fluidic actuators are light, flexible, and easy maintenance. These properties make them inherently safer, and more cost-effective than many other actuation systems that are based on rigid mechanics. Thus, they would be a good choice for rehabilitation, which requires high level of safety, and relatively low cost, as the goal is to replace human labor.

The main function of a hand is to grasp and manipulate objects, and the thumb contributes most to this function. Losing the thumb causes a 40 % loss of overall hand function [8]. Therefore, the thumb could be considered the most important target in hand rehabilitation. However, because the joints of the thumb make very complicated motions, their control is difficult. Thumb rehabilitation, especially with soft robotic gloves, has not been widely investigated.

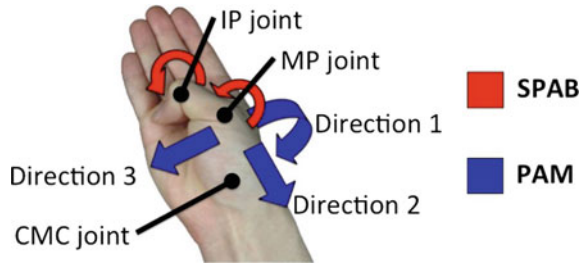
In this paper, we present a design for a soft robotic thumb rehabilitation system, using a soft pneumatic actuator and a pneumatic artificial muscle-based parallel link mechanism, with the goal of facilitating research in hand rehabilitation therapy.

2 Methods

2.1 Mechanism Design

The main function of the interphalangeal (IP) and metacarpophalangeal (MP) joint is flexion-extension. Therefore we use a soft pneumatic actuator for bending (SPAB), which is placed on the dorsal side of the thumb. Thus, the palmar side is left free. Furthermore, there is no need for fine adjustment of the joint pivot point, unlike in systems that are based on rigid mechanics, because the actuator shape conforms to the shape of the assisted digit. For the carpometacarpal (CMC) joint,

Fig. 1 Functional concept of the mechanism



we used a parallel link mechanism by coupling pneumatic artificial muscles (PAM) in three directions around the proximal side of the MP joint (Fig. 1). This simple mechanism enables, ideally, a full stable control and a wide range of motion for the first metacarpal bone. In consideration of comparing the amount of actuator contraction and muscle contraction, we use two PAMs for each direction. We defined traction directions 1: dorsal side, 2: radial side, 3: palmar side (Fig. 1).

2.2 System Concept

The proposed system layout is shown in Fig. 2. The PAMs are placed on an orthotic structure on the forearm to take their size and contraction motion range into consideration. The motion is transmitted to the thumb through wires that are guided by hard tubing attached to the orthosis.

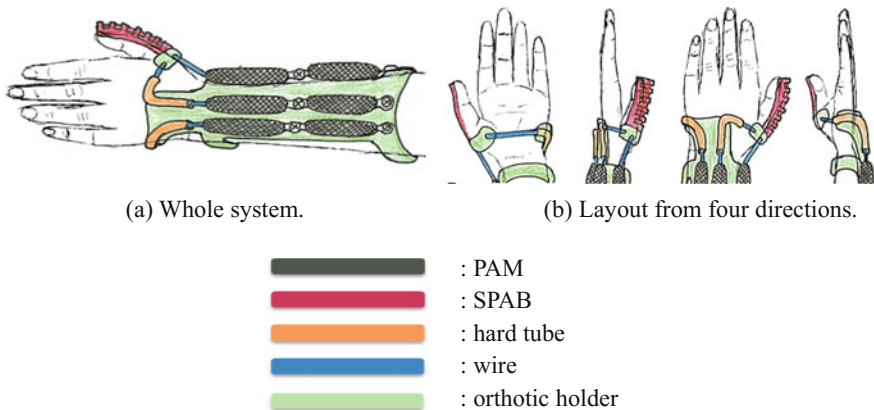


Fig. 2 System concept

2.3 *Pneumatic Artificial Muscle (PAM)*

We used PM-10P (SQUSE Co.) as PAM (Fig. 3). Its specification is shown in Table 1.

2.4 *Soft Pneumatic Actuator for Bending (SPAB)*

For the thumb flexion we used a 2-pocket SPAB (Fig. 4) [10], which is a modified version of similar actuators by Deimel and Brock [11]. The actuator's specification is shown in Table 2.

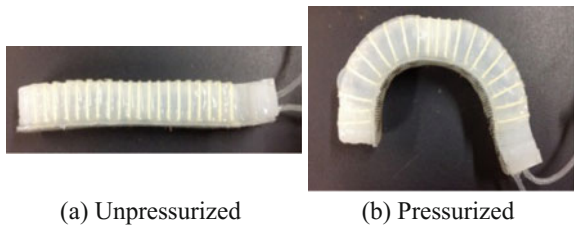
Fig. 3 External appearance of PAM



Table 1 Specification of PM-10P [9]

Drive system	McKibben type
Average maximum contraction	30 %
Maximum traction (at rated pressure)	100 N
Specification fluid	Air
Rated pressure	200 kPa
Mass	3 g
Maximum pressure (range of use)	250 kPa
Durability (inflation cycles at rated pressure)	50,000
External dimensions (no pressure) (full length × outer diameter)	75 × φ 10 mm
External dimensions (pressurized)	52.5 × φ 24 mm

Fig. 4 External appearance of SPAB



(a) Unpressurized

(b) Pressurized

Table 2 Specification of SPAB

Rated pressure		150 kPa
Specification fluid		Air
Material	Main body	Silicone
	Reinforcement fiber	Cotton thread
	Inextensible bottom layer	Polypropylene net
Mass		27 g
External dimensions		W18 × H17 × L105 mm

2.5 Prototype

We evaluated the proposed mechanism using a prototype (Fig. 5a, b). A supporting structure for the system was made from wood, and fixed to a desk. A first metacarpal bone holder was made of polymer clay and a PAM holder was made from wood (Fig. 5c). We used a Velcro straps for the fixation of PAM and the first metacarpal bone holder. Total weight of the prototype system was approximately 100 g, without an orthotic holder, and excluding the dummy hand structure and its supports.

2.6 Thumb Dummy Hand

For the measurement of the prototype, we made a thumb dummy hand in imitation of a human thumb (Fig. 6). For the CMC joint we used a plastic ball joint, which enabled flexion-extension and adduction-abduction. The part equal to the first metacarpal and phalanx bones were made of 5 mm diameter wood sticks. The sticks were covered with hot melt glue to form the general shape of the bones.

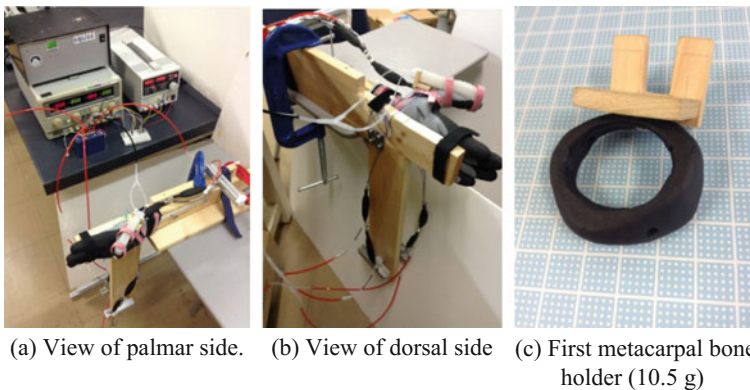
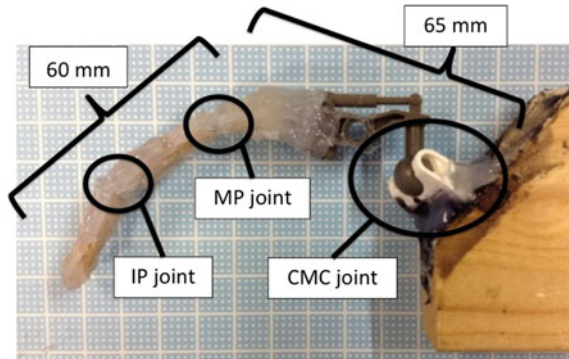
**Fig. 5** Prototype and first metacarpal bone holder

Fig. 6 Thumb structure for dummy hand



The IP and MP joints were made as flexural joints from the same hot melt glue to enable only flexion-extension. The length of the metacarpal part was 65 mm, and the combined length of phalanges was 60 mm. These measures were based on the data of Kawauchi [12].

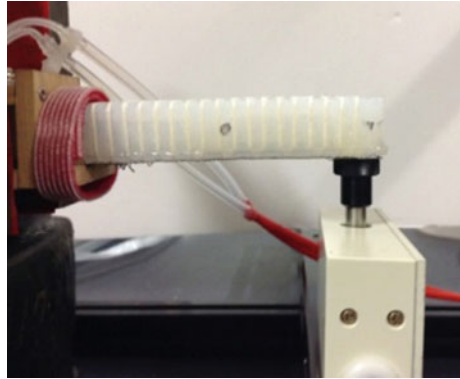
2.7 Measurement of SPAB

The following experiments were carried out to confirm whether the modified SPAB for thumb satisfied the requirements for using it for motion assist.

To estimate the trajectory of the thumb tip, the SPAB was pressurized with air three times while the camera captured the motion from the side. For this, the SPAB was clamped in a vertical position from its proximal end, and a black marker was used to mark the tip, the middle point, and the proximal end. The pressure inside the SPAB was gradually increased from 0 to 150 kPa, in increments of 25 kPa. The SPAB tip angle was measured from the images as the angle between the vector formed by the proximal end and tip markers at 0 kPa, and the tip normal vector at different pressures.

A force gauge (ZTS-i, IMADA Co.) was used to measure the force output. It was brought in contact with the tip of the actuator as shown in Fig. 7. The pressure inside the SPAB was gradually increased from 0 to 125 kPa, in increments of 25 kPa, and the force exerted by its tip was recorded. The rated pressure was 150 kPa, but it was not considered, because the tip of the SPAB did not stay on the pedestal of the force gauge due to its large deformation at 150 kPa. The experiment was repeated three times.

Fig. 7 View of force measurement setup



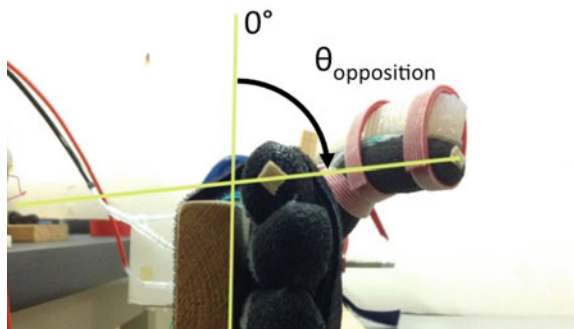
2.8 *Thumb Opposition Movement*

Support of the thumb opposition movement is one of the most important functions for a hand rehabilitation device. We evaluated whether the proposed mechanism could really assist opposition. At first, 200 kPa air pressures was applied to dorsal and radial side PAMs. Next, the air pressure was released from them, and 200 kPa air pressure was applied to the palmar side PAM. In the end, 150 kPa air pressure was applied to the SPAB.

2.9 *Angle of Thumb Opposition*

The overall motion path of the thumb during opposition is combination of abduction, medial rotation, and flexion. However, in this study we defined the angle of thumb opposition as the angle between a vector from the index finger tip to the thumb tip, and the back board, which was parallel to the back of the dummy hand (Fig. 8).

Fig. 8 Angle of thumb opposition



PAM2 (radial side) pressure was kept constant, while PAM1 (dorsal side) and PAM3 (palmar side) pressures were changed in 50 kPa increments from 200 to 0 kPa, and from 0 to 200 kPa, respectively. This was repeated five times, increasing PAM2 pressure from 0 to 200 kPa, in 50 kPa increments. From the results we calculated the relationship between pressure and angle of thumb opposition. We considered the target value to be 60° , based on the results of Li and Tang [13].

2.10 Angle of Adduction-Abduction

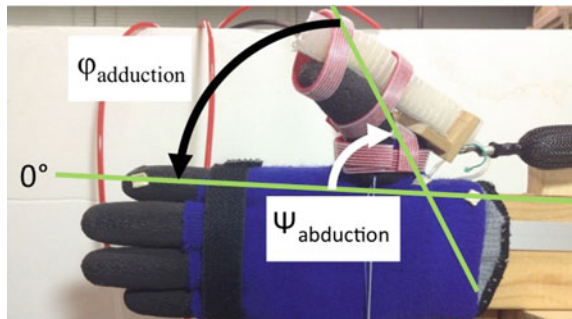
The wide range of motion of the CMC joint enables the thumb opposition and adduction-abduction motions [14]. Many activities of everyday life require these two complex motions. Yonemoto et al. defined the angle of thumb adduction-abduction as 60° [15]. We measured the angle using this value as a reference. We put two markers on the side of the first metacarpal bone, one on the index finger tip, and one approximately where the end of the radial bone would be. We defined the angle of thumb abduction ($\Psi_{\text{abduction}}$) and adduction ($\varphi_{\text{adduction}}$) as the angle between these markers (Fig. 9). The pressure of each PAM was gradually changed from 0 to 200 kPa, in increments of 50 kPa. We calculated the relationship between the pressure and the angle of thumb adduction and abduction. The experiment was repeated three times.

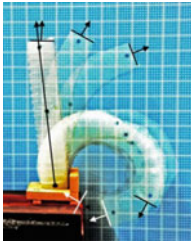
3 Results

3.1 Measurement of SPAB

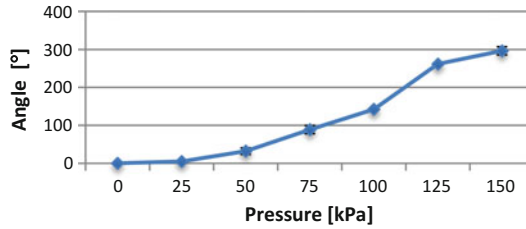
Results of trajectory measurements are shown in (Fig. 10a, b. Error bars show the standard deviations. The mean maximum bending angle was 296° at 150 kPa pressure.

Fig. 9 Angle of adduction-abduction





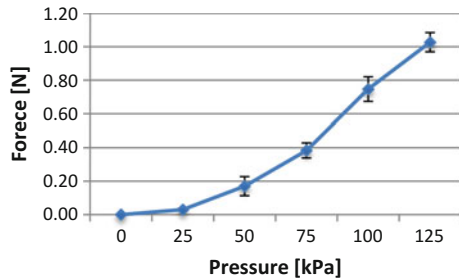
(a) Trajectory of the SPAB tip.



(b) Mean bending angle.

Fig. 10 Results of the trajectory and bending angle

Fig. 11 The mean tip force data of the SPAB with standard deviations



The mean force result is shown in Fig. 11. The mean maximum tip force at 125 kPa was 1.03 N.

3.2 *Thumb Opposition Movement*

Figure 12 shows the states of movements. The proposed mechanism was able to achieve the needed motions to assist opposition.

3.3 *Angle of Thumb Opposition*

The result is shown in Fig. 13, Table 3. We labeled the results from *a* to *e* based on PAM2 pressure increments. On the whole, the mean maximum angle difference was 69.7° between *c-i* and *a-v*.

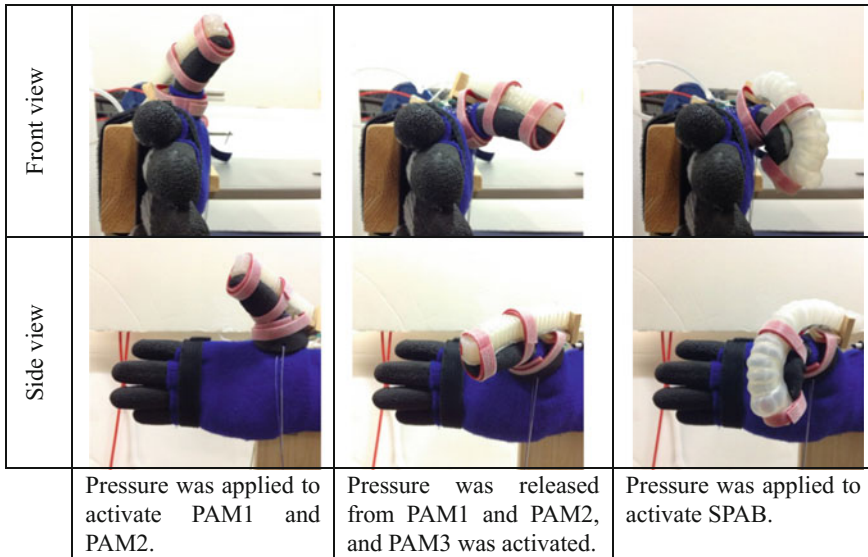


Fig. 12 Thumb opposition movement

3.4 Angle of Adduction and Abduction

The result is shown in Fig. 14. In adduction, the mean maximum angle was 62.7° at f-i, the mean minimum angle was 3.3° at f-v, and their difference was 59.3° . Similarly, in abduction, the mean maximum angle was 57.3° at f-i, the mean minimum angle was 3.3° at f-v, and their difference was 54.0° .

4 Discussion

4.1 Measurement of SPAB

Regarding the bending angle, the combined range of motion of IP and MP joint is 160° as described by Yonemoto et al. [15]. Experiment results from tracking the tip of the SPAB demonstrate its ability to curl approximately 300° . The result of the SPAB tip force is almost the same as in a previous study by Polygerinos et al. [7]. The mean maximum bending angle and force of their actuator were 320° and 1.21 N. Based on this, we can say that the SPAB has good enough performance for assisting the flexion-extension movement.

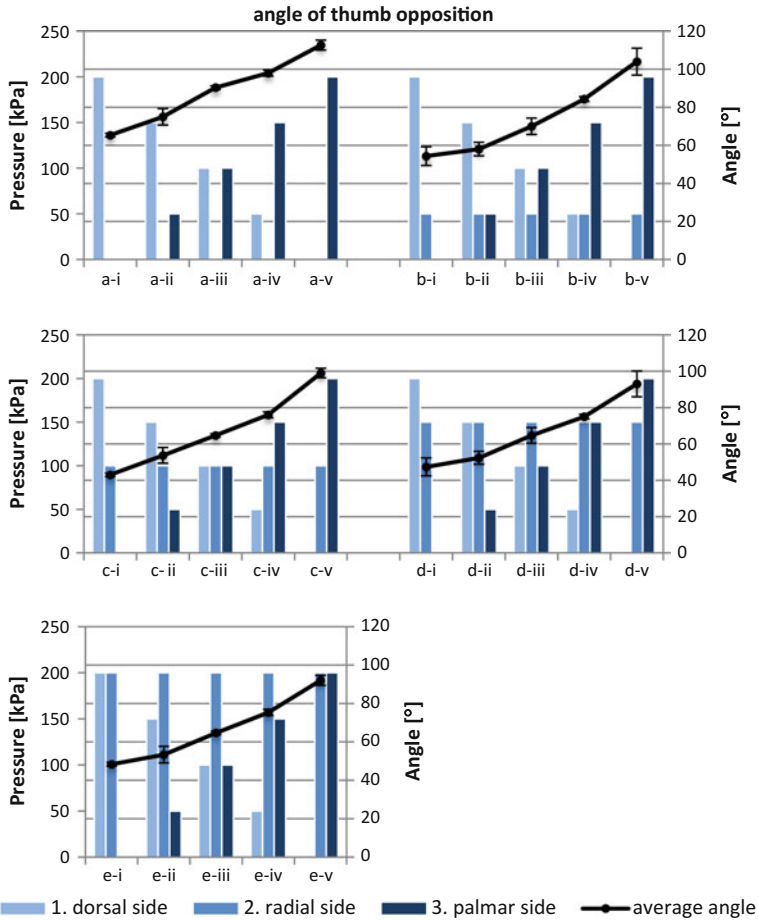


Fig. 13 The results of angle of thumb opposition. The vertical axes show the pressure and angle and the horizontal axis shows alteration of each PAM air pressure. Bar graphs show each PAM air pressure and the line graph shows average angles. Error bars show the standard deviations

Table 3 Mean minimum and maximum angle of each PAM2 pressure

PAM2 (radial side) pressure (kPa)	Minimum (°)	Maximum (°)	Difference (°)
0 (a)	65.3	112.7	47.4
50 (b)	54.3	104.0	49.7
100 (c)	43.0	99.0	56.0
150 (d)	47.3	93.0	45.7
200 (e)	48.3	92.0	43.7

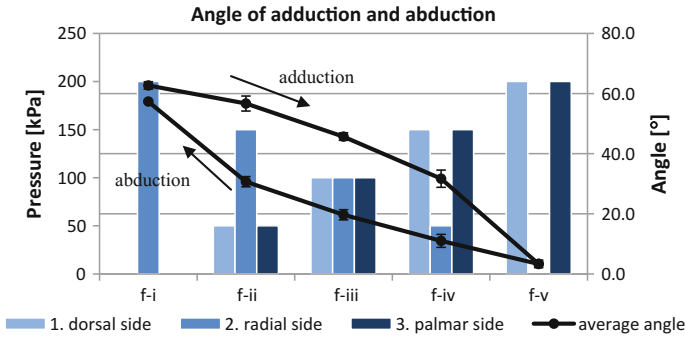


Fig. 14 The results of angle of adduction and abduction. The *vertical axis* shows the pressure and angle and the *horizontal axis* shows alteration of each PAM air pressure. *Bar graphs* show each PAM air pressure and the *line graph* shows average angles. *Error bars* show the standard deviations

4.2 Thumb Opposition Movement

The proposed mechanism enables opposition that is the most important motion of the thumb, and it could assist CMC, IP, and MP joints independently.

4.3 Angle of Thumb Opposition, Adduction and Abduction

The adduction, abduction and opposition are possible with the proposed mechanism, and the adjustment of the angle is possible by changing pressure independently for each actuator. However, there were standard deviations of more than 7° at *b-v* and *c-v*, as seen in Fig. 13. Also, hysteresis in the actuator response between adduction and abduction, as shown in Fig. 14, caused the intermediate state angle to be different. This could cause unwanted stress on the user's joints, and should be considered in the future to ensure stable and safe movement assist.

Our future prospects are making of a geometric model and a controller based on it, examining the use of the system on a real human hand, and improving its portability. After this, we will develop a device that assists all digits, and verify its efficacy for rehabilitation.

5 Conclusion

In this paper, we presented a design for a soft robotic thumb rehabilitation system, using a parallel link mechanism with pneumatic actuators. We evaluated the proposed mechanism using a prototype. The results showed that the proposed mechanism enables opposition, radial abduction, and ulnar adduction motions for the thumb.

References

1. The Royal College of Physicians Intercollegiate Stroke Working Party: National clinical guideline for stroke. 4th edn. London: Royal College of Physicians (2008)
2. Kutner, N.G., Zhang, R., Butler, A.J., Wolf, S.L., Alberts, J.L.: Quality-of-life change associated with robotic-assisted therapy to improve hand motor function in patients with subacute stroke: a randomized clinical trial. *Phys. Ther.* **90**(4), 807–824 (2012)
3. Wolf, S.L., Blanton, S., Baer, H., Breshears, J., Butler, A.J.: Repetitive task practice: a critical review of constraint-induced movement therapy in stroke. *Neurol* **8**(6), 325–338 (2002). PMC. Web. 1 Mar. 2016 <http://doi.org/10.1097/01.nrl.00000031014.85777.76>
4. Heo, P., Gu, G.M., Lee, S., Rhee, K., Kim, J.: Current hand exoskeleton technologies for rehabilitation and assistive engineering. *Int. J. Precis. Eng. Manuf.* **13**(5), 807–824 (2012)
5. Takahashi, C.D., Der-Yeghiaian, L., Le, V., Motiwala, R.R., Cramer, S.C.: Robot-based hand motor therapy after stroke. *Brain* **131**(2), 425–437 (2008)
6. Noritsugu, T., Yamamoto, H., Sasaki, D., Takaiwa, M.: Wearable power assist device for hand grasping using pneumatic artificial rubber muscle. In: SICE Annual Conference (Hokkaido Institute of Technology, Japan), pp. 420–425. SICE 1999, Aug 2004
7. Polygerinos, P., Wang, Z., Galloway, K.C., Wood, R.J., Walsh, C.J.: Soft robotic glove for combined assistance and at-home rehabilitation. *Robot Auton Syst* **73**, 135–143 (2015). doi:10.1016/j.robot.2014.08.014
8. Jones, L.A., Lederman, S.J.: *Human Hand Function*. Oxford University Press, New York (2006)
9. SQUSE, PM-10P, (<http://www.squse.co.jp/product/detail.php?id=9>) (in Japanese), Accessed 7 June 2016
10. Tarvainen, T.V.J., Yu, W.: Preliminary results on multi-pocket pneumatic elastomer actuators for human-robot interface in hand rehabilitation. In: IEEE-ROBIO, 6–9 Dec 2015
11. Deimel, R., Brock, O.: A compliant hand based on a novel pneumatic actuator. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA) (2013)
12. Kawauchi, M.: AIST dimensions data of the Japanese hand (2012). (<https://www.dh.aist.go.jp/database/hand/index.html>) (in Japanese), Accessed 7 June 2016
13. Li, Z.M., Tang, J.: Coordination of thumb joints during opposition. *J. Biomech.* **40**(3), 502–510
14. Yasuo, U.: *The hand—Its Function and Anatomy*, 4th edn. Kinpodo, Kyoto (2006). (in Japanese)
15. Kyoza, Y., Shigenobu, I., Toru, K.: Joint range of motion display and measurement method. *Jpn. J. Rehabil. Med.* **32**(4), 207–217 (1995) (in Japanese) (<http://doi.org/10.2490/jjrm1963.32.207>)

An fMRI Study on Vibration Stimulation Synchronized Mirror Therapy

Kazuya Imai, Kahori Kita and Wenwei Yu

Abstract Hemiplegia is one of the major deficits caused by strokes. Mirror therapy (MT), which causes visual illusion to severely impaired side through presenting mirror images of the normal side in exercises, has been used in rehabilitation practice. However, due to individual differences, it is difficult for some patients to experience visual illusion. We had a hypothesis: vibration simulation for kinesthetic illusion, synchronized with visual stimulation with mirrored images could improve the elicitation of visual illusion. In this research, functional magnetic resonance imaging (fMRI) was employed to test the hypothesis. The brain activities of the subjects performing finger motor tasks with MT and vibration stimulation with the parameters identified in another experiment were compared. As a result, it was found that the parietal lobe of both sides of most of subjects showed higher activities. These two facts suggested MT synchronized tactile stimulation could achieve a higher possibility of visual illusion.

Keywords Mirror therapy · Vision illusion · Functional magnetic resonance imaging · Kinesthetic illusion · Tendon vibration

K. Imai (✉)
Department of Medical System Engineering, Chiba University,
Chiba, Japan
e-mail: kiacya2326@gmail.com

K. Kita (✉) · W. Yu (✉)
Center for Frontier Medical Engineering, Chiba University,
Chiba, Japan
e-mail: kkita@chiba-u.jp

W. Yu
e-mail: yuwill@faculty.chiba-u.jp

1 Introduction

The cerebral stroke causes aftereffects in most cases such as hemiplegia. Owing to the neuroplasticity, it is possible to recover part of lost functions through rehabilitation. One of the rehabilitation methods is called mirror therapy. Mirror therapy was originally proposed as a treatment for phantom limb pain [1]. When a mirror is placed at a sagittal plane between the paralyzed limb and the phantom limb, the mirror reflection of the paralyzed limb is superimposed on the phantom limb, creating the illusion that the paralyzed extremity is still workable. This mirror reflection caused the sensation in patients with phantom limb pain that they could move and relax the often cramped phantom limb and experienced pain relief [2]. This mirror therapy could be also applied to rehabilitation for hemiplegic patients [3]. However, some patients don't experience the illusion due to individual differences.

On the other hand, giving the tendon of muscles of a limb vibration at 100 Hz excites its muscle spindle afferents [4, 5, 6], and the brain receives and processes their inputs so that people experience a sensation of slow movement as if the vibrated limb was moving (kinesthetic illusory limb movement) in the absence of actual movement, intention to move, or sense of effort [7].

Therefore, we hypothesized that combining visual and contralateral kinesthetic illusions can enhance the rehabilitation effect given by only the mirror. In our study, we measured the brain activity by using fMRI when subjects were given stimulation in the form of vibration together with mirror therapy.

2 Experiment 1: Evaluation of Kinesthetic Illusion

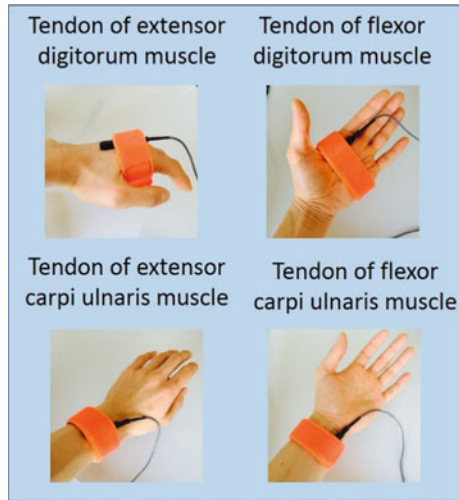
2.1 Subjects

Six healthy right-handed participants (four males and two females) aged 22–24 with no history of neurological or other diseases participated in this study.

2.2 Experiment Setup

Before the fMRI experiment, we evaluated the vibration induced illusory hand movement. We assumed the right hand to be the non-paralyzed side, and left hand to be the paralyzed side. We stimulated different positions of the left hand (Fig. 1)

Fig. 1 Different sites of stimulation



with 100 Hz for 9 s and repeated for three times. Subjects evaluated the illusion degree with the numbers 1–5, here 1 means the lowest illusion degree, and 5 means the highest one.

2.3 Results

The mean of the illusion degree was shown in Table 1. From the result, stimulation at the tendon of extensor digitorum muscle showed the highest mean of illusion degree, meaning that all subjects experienced the highest degree of illusion from the stimulation in average. Therefore, we decided to stimulate the tendon of extensor digitorum muscle in our next experiment.

Table 1 The illusion degree

Places of stimulation	Motion of illusion	Mean of illusion degree	Standard deviation
Tendon of extensor digitorum muscle	Slowly straightening of the second joint	3.83	0.896
Tendon of flexor digitorum muscle	Slowly bending of the second joint	3.17	1.34
Tendon of extensor carpi ulnaris muscle	Slowly straightening of the wrist	2.00	1.00
Tendon of flexor carpi ulnaris muscle	Slowly bending of the wrist	1.50	0.500

3 Experiment 2: fMRI Study on MT and Vibration Stimulation

3.1 Experiment Setup

In experiment 2 we carried out mirror therapy (Fig. 2) while giving vibration to the tendon to give both visual and kinesthetic illusions. 1.5T fMRI system Echelon Vega (Hitachi Medico, Japan) was used in our experiment. The photography setting was concluded in Table 2. A video camera (TOA, MR C-CV150) was used to capture hand motion and the recorded motion was shown to the subject as the reflection of the non-paralyzed side. For stimulating vibration in the fMRI device, we used a non-magnetic body vibration stimulation device (Uchida Denshi Co., Ltd., Japan). All participants gave their informed consent, and the experiment was approved by Ethical Committee of Tokyo Denki University, where the experiment was performed.

The experiment was carried out as shown in block diagram (Fig. 3) with intervals of 9 s of task and 15 s of rest time. During the task, we also stimulated the tendon of extensor digitorum muscle with 100 Hz. It is known that slower movement is easier to cause illusion [5]. However, other criteria (speed, frequency, power) are needed to be taken into consideration when we measure brain activity during the

Fig. 2 Presentation course of the mirror picture

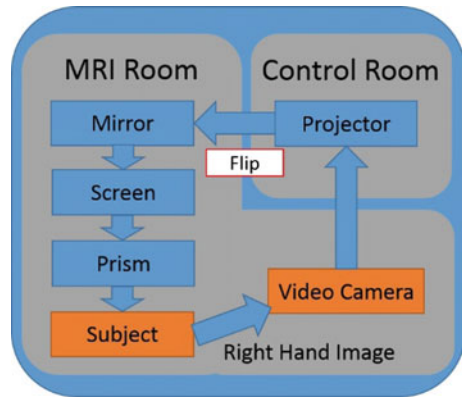


Table 2 fMRI scan parameters

Sequence	GE EPI
FOV	240 mm
TR	2400 ms
TE	45 ms
FA	90
Thickness	5 mm
Multi slice	24

Fig. 3 Block design

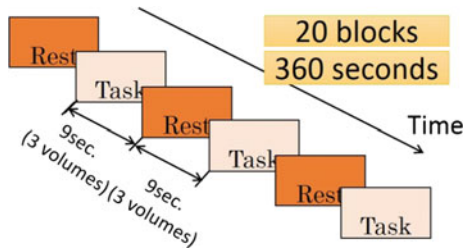


Table 3 Task description

		Right hand	Left hand
Fast task	Sub-Task 1	Bending and straightening (2 Hz)	Vibration stimulation
	Sub-Task 2	Bending and straightening (2 Hz)	No stimulation
	Rest	Relax	No stimulation
Slow task	Sub-Task 1	Bending (9 s)	Vibration stimulation
	Sub-Task 2	Straightening (9 s)	No stimulation
	Rest	Relax	No stimulation

movement [6]. Therefore we prepared two types of tasks: fast task and slow task. In addition, in each task there are two sub-tasks: sub-task 1 and sub-task 2, as shown in Table 3.

3.2 Results

Two brain areas, parietal lobe, and primary motor cortex of frontal lobe, were focused on. The parietal lobe works to integrate visual and somatic sensation [7]. In addition, there is a mirror neuron system in the parietal lobe. Therefore, activation of both sides of parietal lobe would denote the occurrence of an illusion. The activity in the ipsilateral (severely impaired side) primary motor cortex denotes rehabilitation effects of mirror therapy [3]. In our study, since we assume the left hand to be the paralyzed side, the activation of the right side primary motor cortex would be analyzed.

We acquired the coordinates of activated brain parts using the analysis software of SPM8. Thereafter, we demonstrated these activated brain regions by using Talairach Client (Ver.2.4.3).

Table 4 The activation number of people of the primary motor cortex of the fast task

	Left frontal lobe	Right frontal lobe
Sub-Task 1	4	2
Sub-Task 2	4	1

Fast Task (Subjects: 4)

All subjects showed activation in the left primary motor cortex in both Sub-Task 1 and Sub-Task 2. However, only two subjects showed activation in the right primary motor cortex in Sub-Task 1 and one subject in Sub-Task 2 (Table 4). The brain activation image of one of four subjects is shown in Fig. 4.

Only two subjects showed activation in both sides of parietal lobe in both Sub-Task 1 and Sub-Task 2 (Table 5).

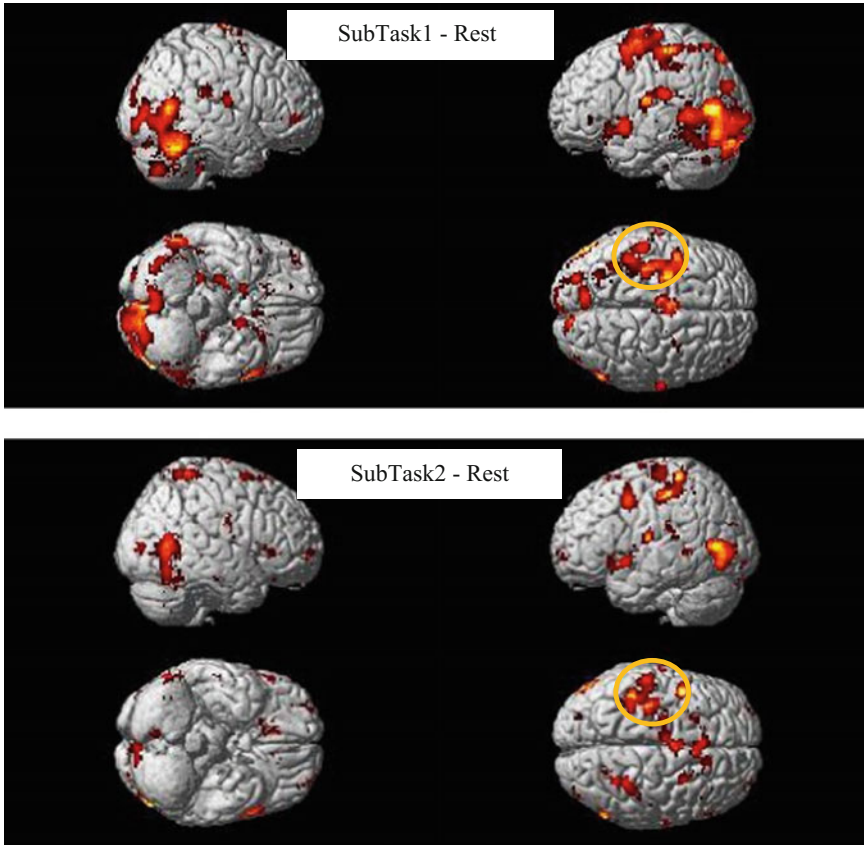


Fig. 4 Vibration and mirrored image induced brain activation during the fast task. (Processed with SPM 8, $P < 0.001$ (uncond))

Table 5 The activation number of people of the parietal lobe of the fast task

	Both sides of parietal lobe
Sub-Task 1	2
Sub-Task 2	2

Slow Task (Subjects: 5)

In the slow task, only one subject showed activation in both sides of left primary motor cortex in Sub-Task 1. In Sub-Task 2, only one showed activation in left primary cortex and none of the subjects showed activation in the right primary cortex (Table 6). The brain activation image of one of four people was shown in Fig. 5.

Four subjects showed activation in both sides of parietal lobe in both Sub-Task 1 and only two subjects in Sub-Task 2 (Table 7).

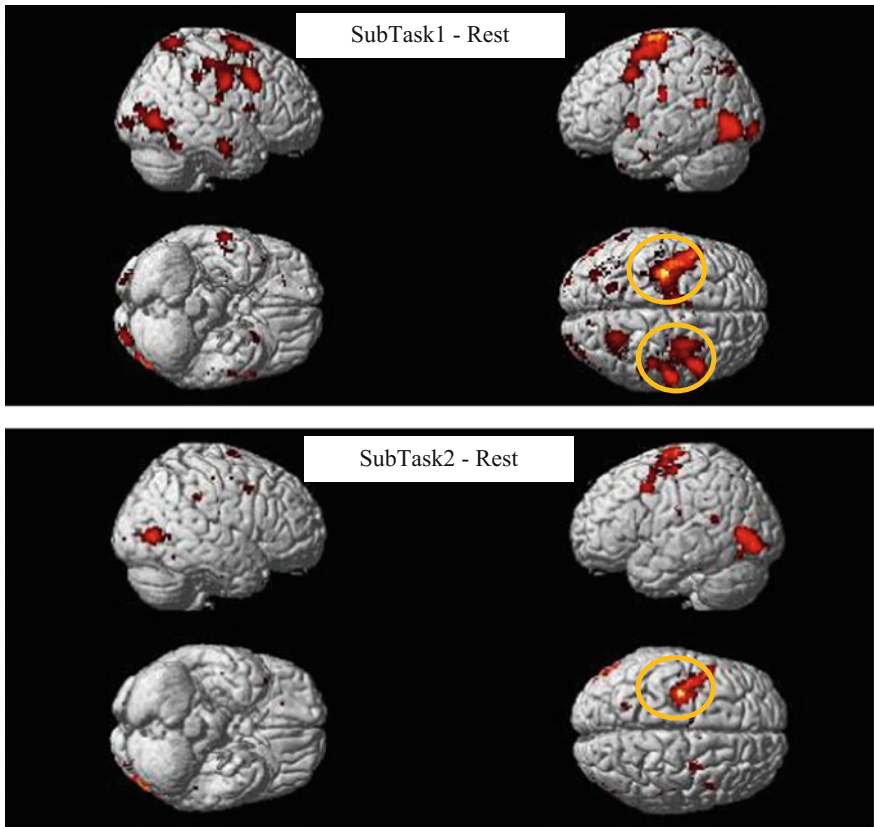


Fig. 5 Vibration and mirrored image induced brain activation during the slow task (Processed with SPM 8, $P < 0.001(\text{uncond})$)

Table 6 The activation number of people of the primary motor cortex of the slow task

	Left frontal lobe	Right frontal lobe
Sub-Task 1	1	1
Sub-Task 2	1	0

Table 7 The activation number of people of the parietal lobe of the slow task

	Both sides of parietal lobe
Task 1	4
Task 2	2

4 Discussions

Basically, the number of subjects was insufficient, especially for the experiment 2. With the current dataset, the discussion made following is quite qualitative.

In the fast task, all the subjects showed activity in their left primary motor cortex, while in Sub-Task 1, 2 of them showed activity in right primary motor cortex, corresponding to their inactive limb, more than the number of right primary motor cortex activated in the Sub-Task 2. Half of the subjects showed clear activity in both sides of parietal lobe.

On the other hand, in the slow task, the clear activity in left primary motor cortex was only found in 1 out of 5 subjects. However, 4 out of 5 subjects showed clear activity in parietal lobe in the Sub-Task 1, more than that of in the Sub-Task 2.

Certainly it is not clear that, the illusion experience in slow task is mainly visual or kinesthetic or both. Since, the sub task with only vibration stimulation was not put into the block diagram due to the scanning time limitation. This needs further investigation. With this in mind, we can assume that, the slow task could elicit illusion for more subjects, than the fast task.

Nevertheless, few subject showed activation of the right primary motor cortex. This might be two possibilities: (1) only visual illusion could induce further the activation of right primary motor cortex; (2) the activation of the left primary motor cortex is the precondition of the activation of its right counterpart. If the reason is (1), then the hypothesis to test shall not hold. If it is (2), then finding slow task that could activate contralateral primary motor cortex would be very critical in near future. Raising the load of the slow task, would be one solution [8].

5 Conclusion

In order to test the hypothesis that a contralateral vibration stimulation synchronized with mirrored images of moving limb could improve the vision illusion, 2 experiments were done. The site for vibration stimulation for kinesthetic illusion was determined through the experiment 1. In the experiment 2, the effect of synchronized stimulation was investigated through a fast motor task and a slow motor task. Results showed that, the fast motor task could induce the activity of the primary motor cortex, however, it has fewer effect on the illusion, and rehabilitation. Whereas, it is difficult for the slow task to induce the activity of primary motor cortex, however, it is easy to elicit the illusion, which was shown by the activity in parietal lobe.

Our hypothesis might hold, however, it needs further research efforts.

References

1. Ramachandran, V.S., Rogers-Ramachandran, D.: Synaesthesia in phantom limbs induced with mirrors. *Proc. R. Soc. Lond. B Biol. Sci.* **263**, 377–386 (1996)
2. Ramachandran, V.S., Hirstein, W.: The perception of phantom limbs. The D.O. Hebb lecture. *Brain* **121**, 1603–1630 (1998)
3. Nasu, T., Hata, Y., Ishiguro, N.: The influence that an exercise illusion using the sight gives to cerebral cortex—Comparison with the mirror therapy (2012)
4. Goodwin, G.M., McCloskey, D.I., Matthews, P.B.C.: The contribution of muscle afferents to kinesthesia shown by vibration induced illusions of movement and by the effects of paralyzing joint afferents. *Brain* **95**(4), 705–748 (1972)
5. Burke, D., Lofstedt, L., Wallin, G.: The responses of human muscle spindle endings to vibration of noncontracting muscles. *J. Physiol. (Lond)* **261**, 673–693 (1976)
6. Naito, Eiichi, Nakashima, Tokuro, Tomonori Kito, Yu., Aramaki, Tomohisa Okada, Sadato, Norihiro: Human limb-specific and non-limb-specific brain representations during kinesthetic illusory movements of the upper and lower extremities. *Eur. J. Neurosci.* **25**, 3476–3487 (2007)
7. Naito, Eiichi, Nakashima, Tokuro, Tomonori Kito, Yu., Aramaki, Tomohisa Okada, Sadato, Norihiro: Human limb-specific and non-limb-specific brain representations during kinesthetic illusory movements of the upper and lower extremities. *Eur. J. Neurosci.* **25**, 3476–3487 (2007)
8. Soya, H., Kato, M., Oomori, T.: The local bloodstream change of the previous motor area judging from near infrared spectroscopy and relations with the exercise strength (In Japanese). *The Jpn. Soc. Phys. Fitness Sport Med.* 642 (2004)

Robot Patient Imitating Paralysis Patients for Nursing Students to Learn Patient Transfer Skill

Chingszu Lin, Zhifeng Huang, Masako Kanai-Pak, Jukai Maeda, Yasuko Kitajima, Mitsuhiro Nakamura, Noriaki Kuwahara, Taiki Ogata and Jun Ota

Abstract This research aims to develop a robot to help nursing students learn how to physically transfer paralyzed patients. Our prior robotic prototype had only been designed to imitate a patient with weak lower limbs; it was unable to imitate the imbalance and instability of the trunk. Therefore, we developed and waist joint on the prototype robot and also the control system in order to emulate quadriplegic and hemiplegic patients' tendency to fall over. The waist was designed with 2 DOF compliant joints. Evaluation of the robot showed that the robot was able to properly imitate the unstable waist movements of paralyzed patients.

Keywords Robot patient · Paralyzed patient · Skill learning · Wheelchair transfer

1 Introduction

1.1 Background

Approximately 1 in 50 people have currently been diagnosed with symptoms of transient or permanent paralysis [1]. This high statistic accordingly calls for an increasing demand in adequate medical care. Paralysis causes a loss of function in muscles and nerves, which makes it difficult, if not impossible, for paralyzed

C. Lin (✉) · T. Ogata · J. Ota

Research into Artifacts, Center for Engineering (RACE), The University of Tokyo, Chiba, Japan

e-mail: lin@race.u-tokyo.ac.jp

Z. Huang (✉)

School of Automation, Guangdong University of Technology, Guangzhou, China

e-mail: zhifeng@gdut.edu.cn

M. Kanai-Pak · J. Maeda · Y. Kitajima · M. Nakamura

Faculty of Nursing, Tokyo Ariake University of Medical and Health Sciences, Tokyo, Japan

N. Kuwahara

Department of Advanced Fibro-Science, Kyoto Institute of Technology, Kyoto, Japan

© Springer International Publishing AG 2017

W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,

Advances in Intelligent Systems and Computing 531,

DOI 10.1007/978-3-319-48036-7_40

patients to perform daily tasks. It is thus essential for nurses to support the patient as they transfer between their bed and their wheelchair. However, patient transfer is considered a difficult task in medical care, as improper support can result in injuries to both the patients and themselves [2].

Unfortunately, despite the importance of this skill, conventional training methods used for training nursing students are inadequate for simulating realistic situations. Students generally practice this skill with patients acted out by healthy people, such as their classmates. However, the behavior of paralyzed patients during transfer is highly different from that of healthy people—with the most obvious behavioral difference being imbalance and instability of the trunk. Hemiplegic patients usually tilt in a lateral (left and right) direction of their paralyzed side, while quadriplegic patients fall in a flexion or extension (front and back, respectively) direction in a flexion movement. Trunk instability requires nurses to constantly support the patient's shoulder, and thus significantly complicates the task of patient transfer in a way that healthy people cannot emulate.

Although the training with real patients would give nursing students first-hand experience at this task, schools cannot allow this because of the immorality, inhumanity, and safety issues involved in allowing untrained students to practice with real patients. Therefore, the next best option is to develop a robot that can simulate the symptoms of paralyzed patients for nursing students to train with.

1.2 Robotic Simulators in Nursing Education

Recent research as focused on replacing conventional healthy human actors with non-human simulators. These simulators allow students to gain experience similar to situations they would actually encounter in the workplace, and thereby allow them to efficiently prepare for the future [3–7].

There are typically two categories of simulators. The first category includes stationary simulators, such as mannequins or models of organs. Stationary mannequins are suitable for imitating motionless patients, with the most common use being for cardiopulmonary resuscitation (CPR) training. An example of the use of stationary mannequins in research includes the studying of swallowing disorders through the use of a model of the respiratory tract [8].

The second category includes robotic simulators equipped with mechanisms to actively imitate the behavior of patients. Robotic simulators can be designed to address a multitude of issues in medical care, as can be seen in studies [9–15]. These purposes include, but are not limited to: treatment of specific symptoms, such as the upper limb robot described in [9], which can mimic the behavior of arms afflicted by spasticity; diagnosis of diseases, such as the prostate simulator robot described in [15], which trains students in deciphering four different prostate conditions: normal state, prostatitis, symmetric inflammation (BPH), and carcinoma; and even awareness of patient mental conditions, such as the robot capable

of expressing chaotic emotion used for injection training described in [13], which trains students to look out for the patient’s emotions based on their body language.

In our previous work, we developed a prototype robot for patient transfer training (herein referred to as “robot patient”) that imitated a patient with weak lower limbs [16–18]. The robot patient utilized ten motors, two brakes, and angular sensors for each joint. The experimental results showed that training with our robot patient improved the patient transfer skills of nursing students.

However, our previous robot patient was only limited to simulating patients with weak lower limbs. As described in the previous section above, it is especially difficult for nurses to transfer the paralyzed patients with trunk instability. Our research aims to expand the functionality of the robot patient to emulate this symptom.

1.3 Purpose

The purpose of this study is to design a simulator robot that can imitate and reproduce the behavior of paralyzed patients, with particular emphasis on the waist joint. By controlling the waist joint, the robot will be able to emulate trunk instability. Hemiplegic and quadriplegic patients will be the targets for this simulation.

2 Patient Transfer and Specifications

2.1 Patient Transfer Skill

Patient transfer is defined as the moving of a patient from a bed to a wheelchair [19, 20]. The following describes the steps of the patient transfer (Fig. 1).

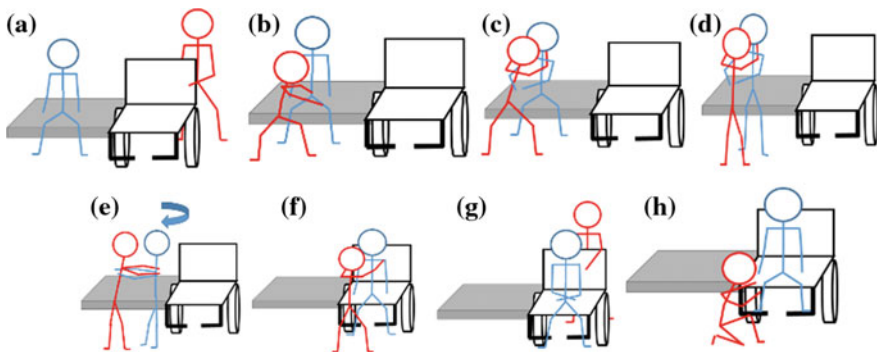


Fig. 1 Processes of patient transfer from step a to h, as described above. The nurse is indicated in red, and the patient is indicated in blue. The bed is on the left side

- (a) Preparing a wheelchair
- (b) Setting the patient in a position that is ready for transfer
- (c) Embracing the patient
- (d) Assisting the patient in standing up
- (e) Turning the patient in the right direction for sitting on the wheelchair
- (f) Assisting the patient in sitting down
- (g) Adjusting the patient’s position on the wheelchair
- (h) Placing the patient’s feet on the footrest

The first step is to park the wheelchair next to the bed, close to the patient. Next, the patient is set to a convenient position for transfer. The nurse swings the patient’s hips and feet to the side of the bed, and gradually moves the patient to the edge. The nurse and patient then embrace as the nurse supports the patient in standing up. Once the patient is standing, the nurse turns the patient towards the wheelchair, and assists the patient in sitting down. Finally, the nurse adjusts the patient to a more comfortable position and puts the patient’s feet on the footrests.

2.2 Paralyzed Patients

As previously mentioned, this study focused on two common types of paralyzed patients: quadriplegic and hemiplegic patients.

- **Quadriplegic Patients**

Patients that suffer from quadriplegia lose control of all four of their limbs as well as their waist [21] (Table 1). If the patient is not properly supported by the nurse during patient transfer, quadriplegic patients will tend to fall forward or backward (pitch rotation) due to trunk instability. Swaying to the left or right (roll rotation) may also occur. However, rotation in the yaw axis seldom occurs in quadriplegic patients, and thus will not be taken into account in this experiment.

- **Hemiplegic Patients**

Hemiplegic patients suffer from paralysis of one side of body (Table 2). For example, left-side hemiplegic patients are unable to control the limbs on the left side of their body, but are able to control those on the right side. In addition, hemiplegic patients often slant towards their paralyzed side because patients tend to

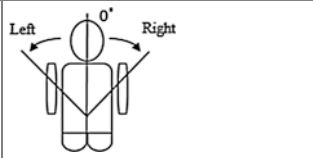
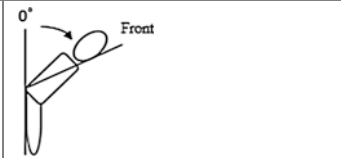
Table 1 Behaviors of quadriplegic patients, and corresponding mechanism functions

Behaviors	Mechanism function for emulation
No voluntary limbs	Do not actuate the motors of arms, hip, and knee in order to imitate loss of muscle function
Trunk instability	<ul style="list-style-type: none"> • If the pitch of the waist is not vertical, the motor will rotate from 0^+ to 60°, or from 0^- to -25° • If the roll of the waist is not vertical, the motor will rotate from 0^+ to 30°, or from 0^- to -30°

Table 2 Behaviors of quadriplegic and corresponding mechanism functions

Behaviors	Mechanism function for emulation
Paralysis of one side	Do not generate the force from the motors of the paralyzed side, in order to imitate the lack of muscle functionality
Trunk instability	<ul style="list-style-type: none"> • Fix the pitch of the waist joint to be vertical, as hemiplegic patients only exhibit unstable movement in the roll axis • Whenever the robot sitting without external support, the motor will rotate causing the robot trunk to slant towards the paralyzed side • When the nurse supports the robot patient’s trunk, the sensor inside the waist enables to detect the force, and follow the force direction

Table 3 Specification of the waist joint of the robot

		
	Roll rotation	Pitch rotation
Range of movement	+30° ~ -30°(Lateral bending)	+60° (Flexion) ~ -25° (Extension)
Maximum torque	22.05 Nm	38.2 Nm
Maximum speed	20°/s	20°/s

subconsciously place their weight in that direction due to their lack of feeling on that side [22]. However, despite being slanted, most hemiplegic patients can still properly sit on the bed. During patient transfer, the nurse needs to prevent the patient from falling towards their paralyzed side. In addition, the nurse needs to park the wheelchair on their non-paralyzed side—as by doing so, the patient can use the leg of their healthy side to better support their weight as they angle themselves to sit down in the wheelchair.

2.3 Specification of Waist Joint

Degrees of Freedom (DOF)

The human lumbar includes 5 vertebrae, each with 3 DOF. However, as mentioned earlier, only the lateral (left and right) and flexion/extension (front and back) movements of the patients were considered. Thus, the waist joint of the robot was designed correspondingly with only 2 DOF: roll and pitch, respectively.

Range of Movement (ROM)

For the ROM of the robot waist, we followed Dvoffik et al. [23] (Table 3). This study measured the ROM of a healthy waist. Paralyzed patients have a similar or smaller ROM compared with healthy people. Therefore, the ROM of a healthy waist should be satisfactory for the robot patient.

Torque

The required torque is calculated based on the mass of robot and the ROM. With a total robot mass of 30.8 kg, an upper body mass of 14.5 kg, and a 1.2 safety factor (SF), the maximum required torque was determined to be 45.6 Nm for pitch rotation and 26.5 Nm for roll rotation.

Angular Speed

The angular speed at which the lateral and flexion/extension movements occurred was determined using videos of paralyzed patients as a reference [24, 25]. The average angular speed of falling was approximated at 10°/s in the roll axis and 15°/s in the pitch axis. In order to account for the variance in waist instability among paralyzed patients, the maximum rotation speed was set to 20°/s for both roll and pitch rotations.

Rotation Centers of Pitch and Roll

The locations of the center of pitch and roll rotation were also a concern of the mechanical design. There are 5 vertebrae in the lumbar, numbered from L5 to L1 from bottom to top [26]. Lower vertebrae, L4 and L5, have the largest ROM for flexion and extension, while L1 has the largest ROM for lateral bending. As a result, the roll rotation center was designed to be in a higher position than the pitch rotation center.

Safety

The robot must be designed to be as safe as possible, as it will actively interact with human nurses. The safety of the nurses should be prioritized.

Detection of External Force

Patient transfer requires mutual cooperation with the nurse. As such, the robot patient should be able to detect and react to the external supporting forces from the nurse.

Flexibility

The flexibility of joints is also a concern. Human joints are flexible and compliant during both bending and stretching.

3 Hardware of Robot Patient

3.1 Mechanical Design of Entire Robot

We developed an entire robot based on the prototype from a previous study [16–18]. The prototype robot was designed using a rescue training mannequin (Laerdal Co., Ltd.). The configuration of the joints of the robot is shown in Fig. 2. There are a total of eighteen joints. The two knee joints were installed with electromagnetic brakes; the eight arm joints were installed with DC motors; and the two waist and hip joints were equipped with individual DC motors. Four joints are free joints without motors. The arms of the robot consist of shoulder, elbow, and finger joints.

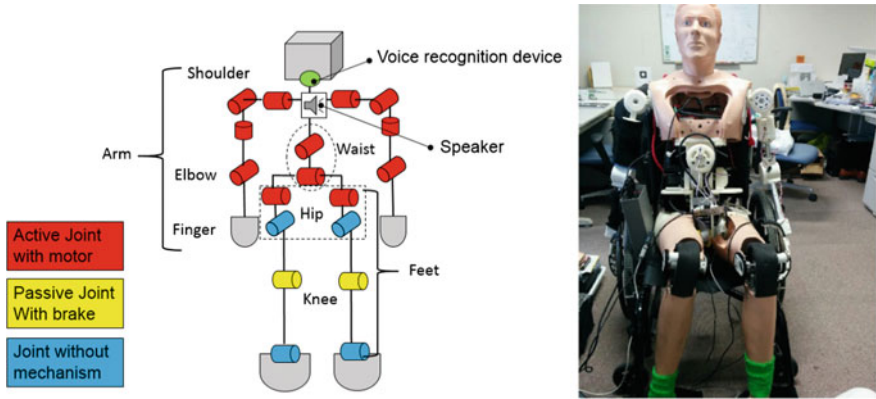


Fig. 2 The configuration of the robot patient

Table 4 Comparison between different types of joints

	Rigid joint	Rigid joint with torque sensor	Compliant joint
Safety	Worst	Middle	Best
Force detection	Current feedback	Torque sensor	Angular sensor
Flexibility	Rigidity	Rigidity	Compliance

The arm is designed with 3 DOF in the shoulder, and one DOF in the elbow joint. Other than the waist, all joints were developed in a previous study [16–18]. In this paper, we focused on the design of the waist joint.

3.2 Mechanical Design of Waist

Compliant Joints

There are three common types of mechanical joints: a rigid joint without sensors, a rigid joint with a torque sensor, and a compliant joint. Table 4 shows the comparison of the three types of joints with respect to safety, force detection, and flexibility. In terms of safety, the rigid joint is the most dangerous, as it can only detect an external force via current feedback of the motor; therefore, it will detect a collision slowest out of all of the types of joints [27]. The joint with a torque sensor is better, as it is able to detect an external force through the torque sensor; however, this type of joint is rigid and stiff, and thus would not be able to absorb the impact of a collision. A compliant joint can not only detect an external force through an angular sensor, but is also able to mimic the softness and flexibility of a real patient’s joints. Therefore, the compliant joint was deemed the most suitable type of joint for our robot.

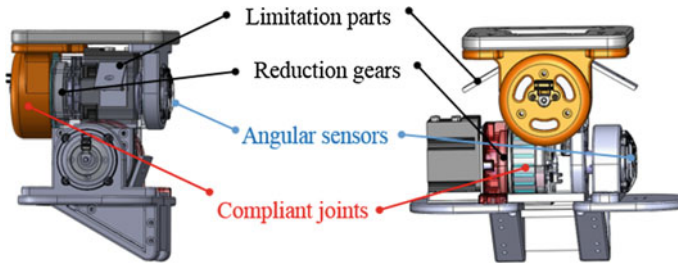


Fig. 3 Side and front views of the waist joint

Motors and Reduction Gear

The design of the waist joint was challenging due to the high 45 Nm torque required to raise the upper body. The maximum torque of servomotors is usually only around 1 Nm, so we installed a reduction gear (ratio 100:1) with each servomotor in order to increase it. We selected servomotors with built-in controllers (manufactured by Moog Animatics Company) in order to conserve the limited space inside of the mannequin. In addition, we also utilized compact harmonic drivers as a reduction gear. The speed of the motor was 5,200 RPM, but was decreased to 5.2 RPM (31°/s) through the reduction gear, which satisfies the 20°/s specification.

Angular Sensors

Angular sensors were installed at the end of the rotating joint in order to detect the position of waist joint. Through the angular data, we can interpret the waist behavior and position, and apply them to the motion control.

ROM Limitation Parts

In order to decrease the burden of the motors, we designed the iron parts to limit the ROM of the waist joint. According to previous session, the angle limit was (+60, -25) for pitch rotation, and (+30, -30) for roll rotation. The iron parts were assembled with the waist joint. Therefore, when the motors rotate to the limitation angle, these parts enable to stop the motor keep rotating and maintain the limitation angle instead of motor itself. There were four limitation parts: left, right, front, and back (Fig. 3). Each part was designed with a specific angle based on the limited ROM.

4 Evaluation

4.1 Method

As described previously, hemiplegic patients suffer from trunk instability in the left and right directions (roll rotation), while quadriplegic patients suffer from trunk instability in the front and back directions (pitch rotation). Therefore, we referred to

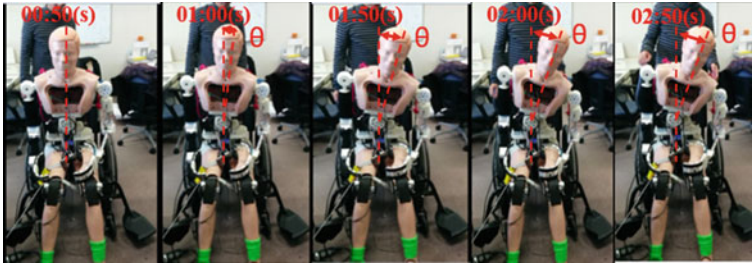
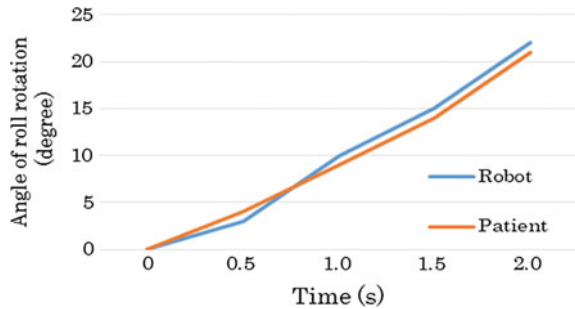


Fig. 4 The robot’s data obtained from the video

Fig. 5 Comparison of the waist joint between the robot and a hemiplegic patient



the clinical video of hemiplegic [24] and quadriplegic [25] patents. The robot behavior was also recorded. The evaluation was performed by comparing the pitch and roll angles over time. The angular data was measured from images that were extracted from the videos every 0.5 s (Fig. 4).

4.2 Results and Discussion

- Hemiplegic Patient (Left-side Paralysis)

Figure 5 shows the relation between angle and time for the robot and a hemiplegic patient as they tilt to their paralyzed side (left). Both the robot and the patient slants to 20° within 2 s. The result exhibits that robot performed similarly to the human patient.

- Quadriplegic Patient

Figure 6 shows the results of the quadriplegia patient and robot. For flexion, both robot and patient fell to 30° within 2 s. For extension, the robot and patient fell to the back side at -10° at about 1.5 s. Both the robot and the patient performed at similar speeds. The flexion speed was 15°/s, and extension speed was 5°/s.

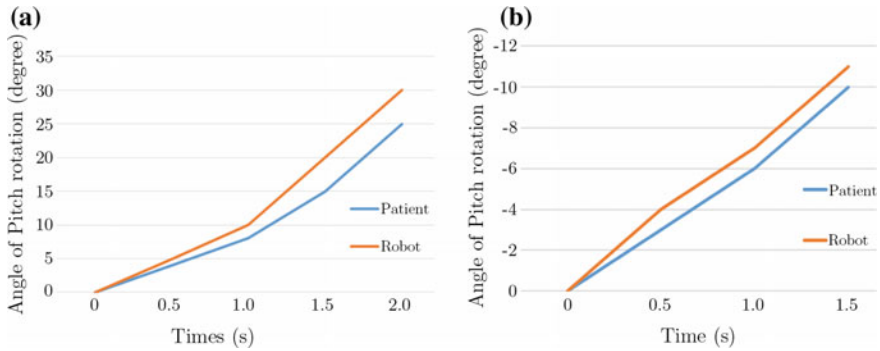


Fig. 6 Comparison of the waist joint between the robot and a quadriplegic patient for the **a** flexion and **b** extension movements

The results show that the angular speeds both of the robot and the patient increase with increasing angle. This is due to the effect of gravity having a larger effect at greater angles.

5 Conclusion and Future Works

This research aimed to develop the waist joint of a robot for nursing students to practice the transfer of paralyzed patients. This robot was able to simulate both hemiplegic and quadriplegic patients through the control of 2 DOF compliant joints. In order to evaluate the success of our robot design, we compared it with the falling movements of real paralyzed patients; the results showed that the waist movement of the robot was similar to actual patients.

In the future, we will attempt to develop the interactions between the robot and nurses that occur during patient transfer. For instance, hemiplegic patients will stop falling down if a nurse supports their shoulder or trunk—the robot should behave similarly in order to simulate a similar scenario for the nursing students. In addition, evaluation of the learning effect on nursing students should be performed.

References

1. Christopherreeveorg: One Degree of Separation, Paralysis and Spinal Cord Injury in the United States (2016). <http://www.christopherreeve.org/atf/cf/{3d83418f-b967-4c18-8ada-adc2e5355071}/8112REPTFINAL.PDF>. Accessed 13 Jan 2016
2. Garg, A., Owen, B.D., Carlson, B.: An ergonomic evaluation of nursing assistants' job in a nursing home. *Ergonomics* **35**(9), 979–995 (1992)

3. Cook, D.A., Hatala, R., Brydges, R., Zendejas, B., Szostek, J.H., Wang, A.T., Erwin, P.J., Stanley, M.J., Hamstra, S.J.: Technology-enhanced simulation for health professions education: a systematic review and meta-analysis. *JAMA* **306**(9), 978–988 (2011)
4. Teteris, E., Fraser, K., Wright, B., McLaughlin, K.: Does training learners on simulators benefit real patients? *Adv. Health Sci. Educ.* **17**(1), 137–144 (2012)
5. Nehring, W.M., Ellis, W.E., Lashley, F.R.: Human patient simulators in nursing education: an overview. *Simul. Gaming* **32**(2), 194–204 (2001)
6. McNett, S.: Teaching nursing psychomotor skills in a fundamentals laboratory. *Nurs. Educ. Perspect.* **33**(5), 328–333 (2012)
7. Alinier, G., Hunt, B., Gordon, R., Harwood, C.: Effectiveness of intermediate-fidelity simulation training technology in undergraduate nursing education. *J. Adv. Nurs.* **54**(3), 359–369 (2006)
8. Moodley, T., Gopalan, D.: Airway skills training using a human patient simulator. *South. Afr. J. Anaesth. Analg.* **20**(3), 147–151 (2014)
9. Fujisawa, T., Takagi, M., Takahashi, Y., Inoue, K., Terada, T., Kawakami, Y., Komeda, T.: Basic research on the upper limb patient simulator. In: *Proceedings of IEEE International Conference on ICORR*, pp. 48–51 (2007)
10. Noh, Y., Segawa, M., Sato, K., Wang, C., Ishii, H., Solis, J., Takanishi, A., Katsumata, A., Iida, Y.: Development of a robot which can simulate swallowing of food boluses with various properties for the study of rehabilitation of swallowing disorders. In: *Proceedings IEEE International Conference on ICRA*, pp. 4676–4681 (2011)
11. Takanobu, H., Omata, A., Takahashi, F., Yokota, K., Suzuki, K.: Dental patient robot as a mechanical human simulator. In: *Proceedings IEEE International Conference on Mechatronics, ICM*, pp. 1–6 (2007)
12. Morita, Y., Kawai, Y., Hayashi, Y., Hirano, T.: Development of knee joint robot for students becoming therapist: design of prototype and fundamental experiments, In: *Proceedings IEEE International Conference on ICRA*, pp. 151–155 (2010)
13. Kitagawa, Y., Ishikura, T., Song, W., Mae, Y., Minami, M., Tanaka, K.: Human-like patient robot with chaotic emotion for injection training. In: *Proceedings of IEEE International Conference on ICCAS-SICE*, pp. 4635–4640 (2009)
14. Goldmann, K., Ferson, D.Z.: “Harvey”, the cardiology patient simulator: Pilot studies on teaching effectiveness. *Am. J. Cardiol.* **45**(4), 791–796 (1980)
15. Gerling, G.J., Rigsbee, S., Childress, R.M., Martin, M.L.: The design and evaluation of a computerized and physical simulator for training clinical prostate exams. *Syst. Man Cybern. Part A: Syst. Humans*, *IEEE Trans.* **39**(2), 388–403 (2009)
16. Huang, Z., Katayama, T., Kanai-Pak, M., Maeda, J., Kitajima, Y., Nakamura, M., Ota, J.: Design and evaluation of robot patient for nursing skill training in patient transfer. *Adv. Robot.* **29**(19), 1269–1285 (2015)
17. Huang, Z., Nagata, A., Kanai-Pak, M., Maeda, J., Kitajima, Y., Nakamura, M., Kuwahara, N., Ogata, T., Ota, J.: Design of a robot for patient transfer training. In: *SICE Annual Conference*, pp. 1041–1046 (2013)
18. Huang, Z., Nagata, A., Kanai-Pak, M., Maeda, J., Kitajima, Y., Nakamura, M., Aida, K., Kuwahara, N., Ogata, T., Ota, J.: Robot patient for nursing self-training in transferring patient from bed to wheel chair, pp. 361–368. *Safety, Ergonomics and Risk Management*, In *Digital Human Modeling. Applications in Health* (2014)
19. Potter, P.A., Perry, A.G. (eds.): *Basic Nursing: Essentials for Practice*. Mosby Incorporated (2003)
20. Rosdahl, C.B., Kowalski, M.T. (eds.) *Textbook of Basic Nursing*. Lippincott Williams & Wilkins (2008)
21. Davies, P.M.: Problems associated with the loss of selective trunk activity in hemiplegia, pp. 31–65. Berlin, In *Right in the middle*. Springer (1990)
22. Karnath, H.O., Broetz, D.: Understanding and treating “pusher syndrome”. *Phys. Ther.* **83** (12), 1119–1125 (2003)

23. Dvoffik, J., Vajda, E.G., Grob, D., Panjabi, M.M.: Normal motion of the lumbar spine as related to age and gender. *Eur. Spine J.* **4**(1), 18–23 (1995)
24. Rehabilitatio7 (2009). Hémiplégie phénomène de gîte. <http://www.youtube.com/watch?v=HO8BT6vznLU>. Accessed 07 Feb 2009
25. Aaron Wood (2010). Quadriplegic working on balancing. <https://www.youtube.com/watch?v=N9boFebQbos>. Accessed 12 Mar 2010
26. Yamamoto, I., Panjabi, M.M., Crisco, T., Oxland, T.O.M.: Three-dimensional movements of the whole lumbar spine and lumbosacral joint. *Spine* **14**(11), 1256–1260 (1989)
27. Tsagarakis, N.G., Laffranchi, M., Vanderborght, B., Caldwell, D.G.: A compact soft actuator unit for small scale human friendly robots. In: *Proceedings of IEEE International Conference on ICRA*, pp. 4356–4362 (2009)

Part X
Multi-robot Systems

A Self-reconfigurable Robot M-Lattice

Shihe Tian, Zhen Yang, Zhuang Fu and Hui Zheng

Abstract Modular robotic systems consists of more than one modules that are homogeneous and this paper proposes a novel structure design of modular robots called M-Lattice, which stands out for its load-capacity and high efficiency in self-reconfiguration. Compressibility of M-Lattice is guaranteed by the topological structure and the amount of compression has been calculated. The motion space indicates that different types of motion can be accomplished without interference. Two prototypes have been made to verify the motion ability of the modules and experiment shows the motion process of M-Lattice system.

Keywords Modular robots · Motion space · Self-reconfigurable · M-Lattice

1 Introduction

Modular Robotic Systems (MRS) are systems composed of homogeneous modules. Modules in the systems can connect and disconnect to each other, and also have relative motion with each other. Thus the systems have the abilities of self-assembly and self-reconfiguration. These abilities give MRS the advantages of strong adaptability to complex and volatile environments.

Since 1980s, various types of MRS have been developed [1]. According to the topological structure, MRS are usually classified into 3 categories: Chain, Lattice

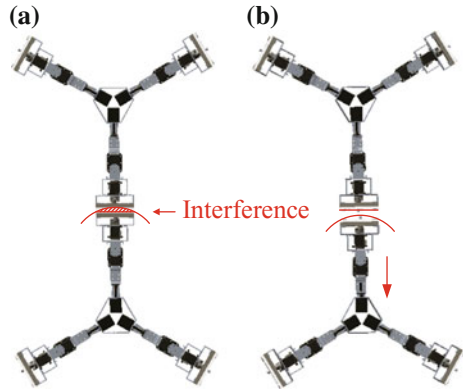
S. Tian · Z. Yang · Z. Fu (✉) · H. Zheng
State Key Laboratory of Mechanical Systems and Vibration, Shanghai Jiao
Tong University, Shanghai, P.R, China
e-mail: zhfu@sjtu.edu.cn

S. Tian
e-mail: tianshihe@sjtu.edu.cn

Z. Yang
e-mail: yangzhen0512@sjtu.edu.cn

H. Zheng
e-mail: huizheng@sjtu.edu.cn

Fig. 1 **a** Interference occurs when one module holds still and the adjacent module moves. **b** Compression of the module helps to escape the interference



and Mobile [2]. Typical chain systems such as PolyBot [3], CONRO [4], CKBot [5] have strong abilities of motion and lattice systems such as Crystalline [6], ATRON [7], Stochastic-3D [8], Soldercubes [1], 3D M-Blocks [9], Molecubes [10] are more structural stable, leading the systems to be larger. Mobile systems are systems whose modules can locomote without connection with each other, and these MRS are helpful in algorithm research, such as Kilobots [11]. To combine advantages of different types of MRS, some hybrid systems have been made. Roombots [12] is a hybrid of lattice and chain. JHU [13] and SMORES [14] are hybrid of lattice, chain and mobile.

M-Lattice is a hybrid system of chain and lattice type and some previous work has been done [15–17]. It stands out for its abilities of taking loads and fast assembly, which leads to superiority in large scale systems, such as space solar power satellite. However, there are some problems remaining. When a connector rotates around the rotation axis without compression, interference will occur. So a compressible module is needed to ensure successful motion, as shown in Fig. 1.

The rest of this paper is organized as follows. Section 1 introduce the structure and components of the module. Section 2 is a calculation of the length to compress and simulation to prove no interference occurs during the motion. Section 3 presents some experiment results using modules of new structure. Finally, Sect. 4 concludes the paper.

2 Structure of M-Lattice

A M-Lattice module consists of a central frame and three arms. Each arm has a sliding pair to enable the compression and a rotate joint to rotate the connector. The connector is at the end of each arm.

The sliding pair is realized by a stepping motor with a screw rod on it and a guide. The screw is used for its self-locking ability and the guide is used to bear the torque. These properties make the MRS more stable.

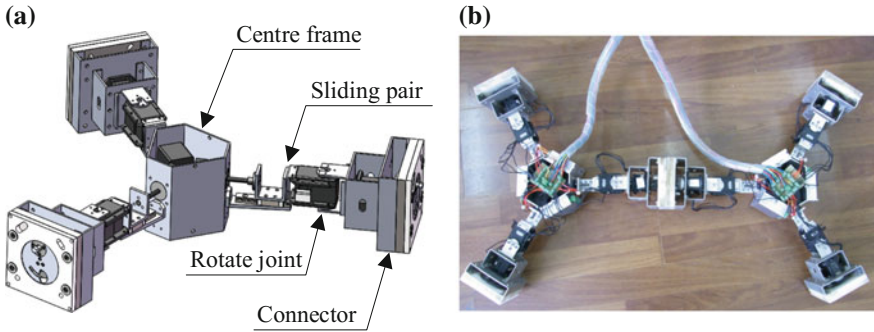


Fig. 2 a 3D model of a M-Lattice module. b Two connected prototypes

The rotate joint is realized by a Dynamixel RX-28 servo motor in consideration that the rotate angle is limited and high precision is needed. The RS485 communication interface of the motor package is convenient for there are several actuators needed in a module.

The connector is a pin-slot mechanism comprising three parts: the rotation disc, the casing fastener and the driving mechanism. There is a horn slot and a pin on the rotation disc. When the pin insert into another connector’s slot, the motor which we also use RX-28 drives the rotation disc to rotate, thus two connectors are fastened. Detailed structure of the connector can be found in [15].

Figure 2 shows the 3D model and prototype of M-Lattice.

3 Kinematics Analysis

3.1 Coordinates Establishment and Transformation

The kinematics of M-Lattice can be solved by transformation of coordinates. Since the M-Lattice System is composed of many homogeneous modules, this problem can be divided into transformation in a single module and transformation between two adjacent modules.

Denavi-Hartenberg (D-H) method [18] is used and Fig. 3a is the coordinates establishment of a single module. The coordinate transformation matrix from one rotate joint to another rotate joint in a module is

$$T_{in} = {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 T = \begin{bmatrix} s(\theta_3 + \theta_5) & c(\theta_3 + \theta_5) & 0 & d_4 s \theta_3 \\ -c(\theta_3 + \theta_5) & s(\theta_3 + \theta_5) & 0 & -d_2 - d_4 c \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

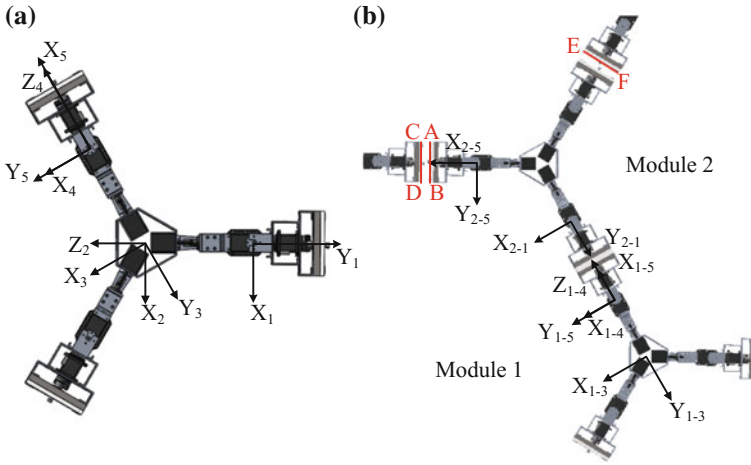


Fig. 3 **a** Coordinates of a single module. **b** Coordinates of two adjacent modules

where

$${}^{i-1}_i T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$s\theta = \sin \theta$
 $c\theta = \cos \theta$

Parameter d_i is the distance between the axis of rotate joint and the centre of the module, and θ_i is the rotation angle of rotate joint. Specially, θ_3 is not a rotation angle of any real rotate joints, it can be $\frac{\pi}{3}$ or $-\frac{\pi}{3}$ which leads to different arms.

As shown in Fig. 3b, when two modules are connected, the coordinate transformation of the two adjacent coordinates which belong to different modules is also in line with D-H method. Thus the coordinate transformation between two modules, as transformation from $X_{1-5}O_{1-5}Y_{1-5}$ to $X_{2-1}O_{2-1}Y_{2-1}$ in Fig. 3b, can be calculated by

$$T_{out} = \begin{bmatrix} -s\theta_{2-1} & -c\theta_{2-1} & 0 & l \\ c\theta_{2-1} & -s\theta_{2-1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where l is the distance between the coordinates.

To ensure no interference occur on condition that one module is moved by another one (condition 1v1), as shown in Fig. 3b, the distance between point A and

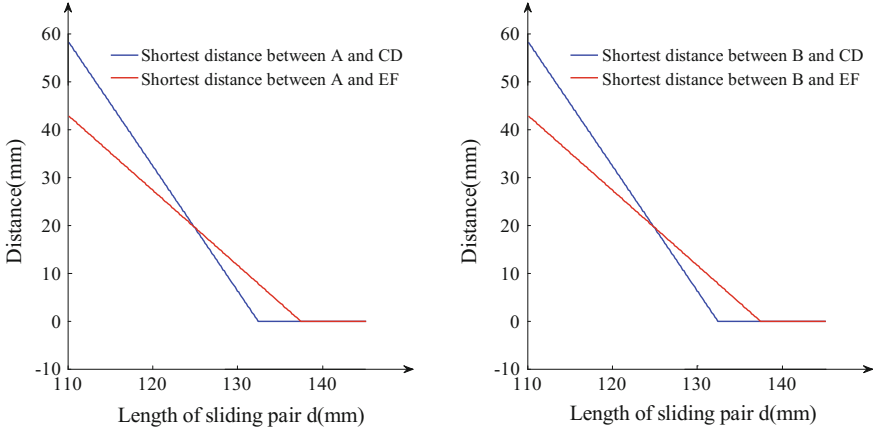


Fig. 4 Relationship between shortest distance of adjacent modules and length of compressed sliding pair

segment CD should be positive all the time, as well as it is between point A and segment EF, point B and segment CD, point B and segment EF.

Assume that the centre of module 1 is the original point of absolute coordinate system, the coordinates of point A and B relative to $X_{2-5}O_{2-5}Y_{2-5}$ can be expressed as vectors $P_A = [P_{AX} \ P_{AY} \ 0 \ 1]^T$ and $P_B = [P_{BX} \ P_{BY} \ 0 \ 1]^T$. Thus the absolute coordinate of point A is

$$P_{A0} = {}^0_{1-3}T_1^{1-3} {}^1_{1-4}T_1^{1-4} {}^4_{1-5}T_2^{1-4} T_{out} T_{in-2} P_A \tag{4}$$

On condition 1v1 in the process of self-reconfiguration, the rotate angle of each joint is limited in $-60^\circ-60^\circ$. When all the connectors are in the ultimate positions away from the centre frame, the modules are connected with each other. Refer to previous modules and in consideration that the distance between the connectors and the centre frame should be larger when we replace a revolute pair by a sliding pair, we assume that $d_{imax} = 160$ mm, and $P_{AX} = 98.5$ mm, $P_{AY} = -45$ mm. Through iterative programs, the relationship of d_i and smallest distance between point A and segment CD, as well as point A and segment EF, point B and segment CD, point B and segment EF, is shown in Fig. 4. When the stroke of the sliding pair is no less than 28 mm, there will be no interaction during the process of self-reconfiguration.

3.2 Motion Space

Motion space of connectors has been drawn to show no interference occur when modules move under reasonable constraints. Figure 5a shows condition 1v1, where the range of rotate angle of each joint is $-60^\circ-+60^\circ$.

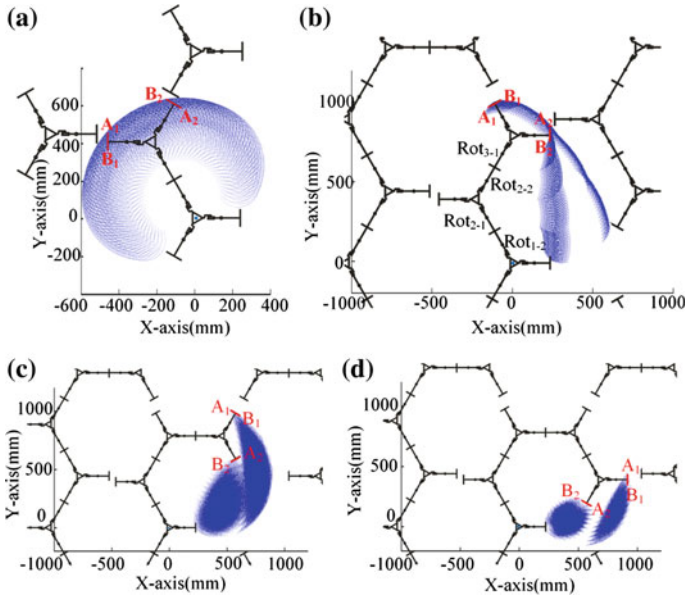


Fig. 5 Space motion under constraints of different conditions. *Blue areas* show the motion spaces of connectors that are marked *red* in the figure

On condition 1v2, the range of rotate angle of each joint is $-30^{\circ}-0^{\circ}$. The coordinates of boundary points A and B can be calculated by

$$P_{A0} = {}_{1-3}T_1^{-3}T_{1-4}^{-4}T_{1-5}^{-5}T_2^1T_{out}T_{in-2}^{-2}T_{out}T_{in-3}P_A$$

$$P_{B0} = {}_{1-3}T_1^{-3}T_{1-4}^{-4}T_{1-5}^{-5}T_2^1T_{out}T_{in-2}^{-2}T_{out}T_{in-3}P_B$$

To ensure no interference occur on this condition, some constrains have to be made: the rotate joints should actuate one by one from Rot_{3-1} to Rot_{1-2} . Once the previous rotate joint rotates to -30° can the next one actuate. Figure 5b shows the motion space of connectors of module 3 moving in this way.

On condition 1v3, the range of rotate angle of each joint is $-15^{\circ}-0^{\circ}$ and no more constrains is required. Calculation is similar to the two conditions above and Fig. 5c shows the motion space. On condition 1v4, the range of rotate angle of each joint is $-6^{\circ}-0^{\circ}$ and motion space is shown in Fig. 5d.

From the motion spaces we can see that if the modules are compressible, different types of motion can be realized under reasonable constraints.

4 Experiment

Two prototype modules have been fabricated to verify the motion ability of the modules. Dynamixel RX-28 servo motors are used for rotate joints and D.M.BERG stepping motors HL-28S113424 are chosen to actuate sliding pairs. Key parameters of prototype module are listed in Table 1.

Stepping motors are controlled by a multi-axis controller and servo motors are controlled by PC. The process is executed automatically under programs and lasts about 36 s, as shown in Fig. 6.

Table 1 M-Lattice specifications

Weight	3.2 kg
Length of short edge of centre frame	34.5 mm
Length of long edge of centre frame	62.5 mm
Length of mechanical arm	184–212 mm
Case material	Aluminium profile
Holding torque of stepping motor	35 mN.m
Max torque of servo motor	37.7 kgf.cm

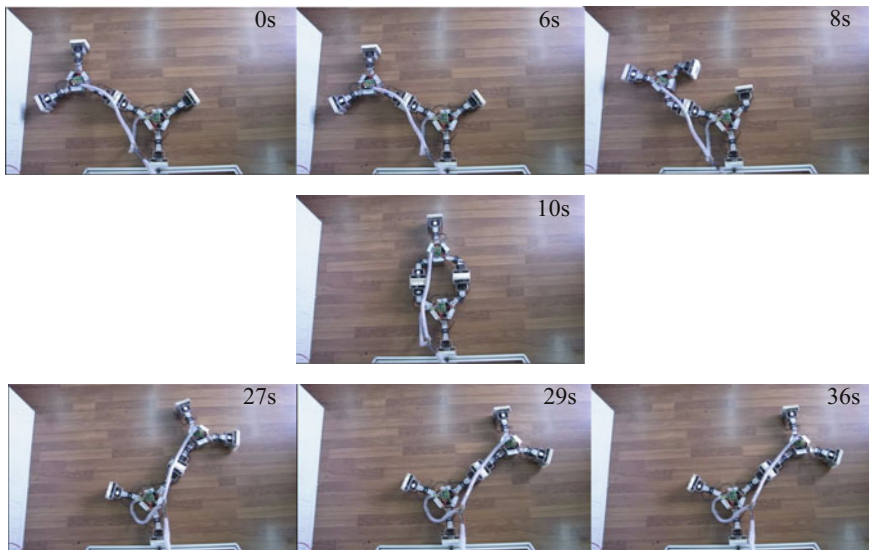


Fig. 6 Photos in the first row show the compression and rotation from initial position to connecting position. Photo in the second row shows the connecting status. Photos in the third row show the rotation and spread from connecting position to terminal position

5 Conclusion

We presented a new self-reconfigurable modular robot structure based on M-Lattice. New topological configuration has been proposed to solve the problem of motion interference. Kinematic analysis has been made to prove no interference occurred in all kinds of motion types. We made two prototypes to verify the motion ability of the new structure of M-Lattice.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China (Grant No. U1401240,61473192,61075086), National Basic Research Program of China(2014CB046302) and the medical cross Fund Project at SJTU (Grant No.YG2012MS53).

References

1. Neubert, J., Lipson, H.: Soldercubes: a self-soldering self-reconfiguring modular robot system. *Auton. Robot.* **40**, 139–158 (2016)
2. Yim, M., Shen, W.M., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., Chirikjian, G. S.: Modular self-reconfigurable robot systems [grand challenges of robotics]. *Robot. Auton. Syst.* **14**, 43–52 (2007)
3. Yim, M., Duff, D.G., Roufas, K.D.: PolyBot: a modular reconfigurable robot. In: ICRA'00. IEEE International Conference on Robotics and Automation, vol. 1, pp. 514–520. IEEE Press, San Francisco (2000)
4. Shen, W.M., Salemi, B., Will, P.: Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots. *Trans. Robot. Autom.* **18**, 700–712 (2002)
5. Fukuda, T., Nakagawa, S., Kawachi, Y., Buss, M.: Self organizing robots based on cell structures-CKBOT. In: IEEE International Workshop on Intelligent Robots, pp. 145–150. IEEE Press, Tokyo (1988)
6. Rus, D., Vona, M.: Crystalline robots: Self-reconfiguration with compressible unit modules. *Auton. Robot.* **10**, 107–124 (2001)
7. Jorgensen, M.W., Ostergaard, E.H., Lund, H.H.: Modular ATRON: Modules for a self-reconfigurable robot. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, pp. 2068–2073. IEEE Press, Sendai (2004)
8. Bishop, J., Burden, S., Klavins, E., Kreisberg, R., Malone, W., Napp, N., Nguyen, T.: Programmable parts: A demonstration of the grammatical approach to self-organization. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3684–3691. IEEE Press, Alberta (2005)
9. Romanishin, J.W., Gilpin, K., Claici, S., Rus, D.: 3D M-Blocks: Self-reconfiguring robots capable of locomotion via pivoting in three dimensions. In: 2015 IEEE International Conference on Robotics and Automation, pp. 1925–1932. IEEE Press, Seattle (2015)
10. Zykov, V., Mytilinaios, E., Desnoyer, M., Lipson, H.: Evolved and designed self-reproducing modular robotics. *Trans. Robot.* **23**, 308–319 (2007)
11. Rubenstein, M., Ahler, C., Hoff, N., Cabrera, A., Nagpal, R.: Kilobot: A low cost robot with scalable operations designed for collective behaviors. *Robot. Auton. Syst.* **62**, 966–975 (2014)
12. Spröwitz, A., Moeckel, R., Vespignani, M., Bonardi, S., Ijspeert, A.J.: Roombots: A hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot. *Robot. Auton. Syst.* **62**, 1016–1033 (2014)
13. Kutzer, M.D., Moses, M.S., Brown, C.Y., Scheidt, D.H., Chirikjian, G.S., Armand, M.: Design of a new independently-mobile reconfigurable modular robot. In: 2010 IEEE

- International Conference on Robotics and Automation, pp. 2758–2764. IEEE Pres., Anchorage (2010)
14. Davey, J., Kwok, N., Yim, M.: Emulating self-reconfigurable robots-design of the SMORES system. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4464–4469. IEEE Press, Vilamoura (2012)
 15. Guan, E., Yan, W., Jiang, D., Fu, Z., Zhao, Y.: A novel design for the self-reconfigurable robot module and connection mechanism. In: 3rd International Conference on Intelligent Robotics and Applications, pp. 400–408. Springer Press, Shanghai (2010)
 16. Guan, E., Fei, J., Pan, G., Fu, Z., Yan, W., Zhao, Y.: Fault self-diagnosis for modular robotic systems using M-lattice modules. *Int. J. Adv. Robot. Syst.* **12**, 31 (2015)
 17. Guan, E., Fu, Z., Fei, J., Zhai, J., Yan, W., Zhao, Y.: M-Lattice: A self-configurable modular robotic system for composing space solar panels. *P. I. Mech. Eng. I-J. Sys.* **229**, 406–418 (2015)
 18. Craig, J.J.: *Introduction to Robotics: Mechanics and Control*. Pearson Prentice Hall, Upper Saddle River (2005)

Adaptive Synchronized Formation Control Considering Communication Constraints

Zhe Liu, Weidong Chen, Junguo Lu, Jingchuan Wang
and Hesheng Wang

Abstract The adaptive synchronized formation control problem of multiple mobile robots is studied in this paper. The communication constraints, including time-varying delays and data sampling, are considered in problem formulation and system design. Furthermore, the parameter uncertainties in system dynamics have also been taken into account and an adaptive formation controller is presented which enables the robot network to achieve the synchronized formation task adaptively. Convergence analyses of the proposed method are presented and several useful properties are provided. Simulation results validate the effectiveness of the proposed adaptive synchronized formation control approach.

Keywords Adaptive control · Sampled data · Synchronized formation control · Time delay

1 Introduction

Multi-robot formation control has broad application prospects in cooperative tasks such as exploration [1] and spacecraft flight control [2]. Traditional formation control methods include the behaviour-based method [3, 4], the virtual structure method [5, 6] and the leader-follower method [7]. The behaviour-based control approach is inspired by animal colonies and designs several basic behaviours. The desired behaviours of mobile agents are accomplished by combining basic behaviours which have

Z. Liu (✉) · W. Chen (✉) · J. Lu · J. Wang · H. Wang
Key Laboratory of System Control and Information Processing,
Department of Automation, Ministry of Education of China,
Shanghai Jiao Tong University, Shanghai 200240, China
e-mail: liuzhesjtu@sjtu.edu.cn

W. Chen
e-mail: wdchen@sjtu.edu.cn

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_42

been given different weightings. The virtual structure method describes the desired formation as a single rigid entity. The desired trajectory of mobile robots is assigned to that of the corresponding virtual robots in the virtual structure. The leader-follower approach divides mobile robots into several leaders and followers. The formation control is achieved by maintaining the relative poses between leader robots and follower robots. The limitations of these existing approaches are described in detail in [8, 9]. For example, for the behaviour-based approach, it is hard to develop the dynamic model of the entire group and ensure the asymptotical convergence of the formation errors; for the virtual structure method, it might be difficult to implement the distributed framework and guarantee the scalability; for the leader-follower approach, “it is hard to take into account the functioning capabilities of different robot, i.e., ability gap of a robot” [8].

Recently, Sun et al. [9] presented a synchronization control approach for the coordination control of multiple motion axes, parallel robotic manipulators as well as multiple mobile robots. This approach guarantees agents to maintain the desired synchronization performance during the whole task accomplishing process instead of achieving motion synchronization objectives at the final time only. Since the synchronized formation control approach has simple control structures and strong robustness, it has received an increasing research interest and several further studies are presented in literatures. Such as the finite-time synchronized control for robot formation [10] and concurrent synchronization of mobile robots in trajectory tracking [11]. However, all the literatures mentioned above have not taken into account the practice constraints of uncertain dynamic system parameters, time delays, and data sampling in real applications simultaneously.

In real application, the practical communication constraints such as time-varying communication delay and data sampling in real communication systems are inevitable. What's more, the parameters in system dynamics are usually can not be estimated accurately and may change dynamically. Without considering these practical constraints, the performance of synchronized formation control approaches in real applications will be declined and even failed. So in this paper, we study the adaptive and synchronized formation control problem for multiple mobile robots under practical constraints.

In our previous work [12], we presented a formation control method using distributed controller with communication delays and sampled data. The contributions of this paper are different with our previous work [12] in following aspects: The parameter uncertainties in system dynamics are firstly considered in this paper and an adaptive control approach is presented. The proposed controller can achieve the synchronized formation performance under uncertainties and external disturbances. What's more, through additional theoretical analyses, several useful properties of the proposed approach are provided in this paper, which will greatly facilitate the applications of the proposed adaptive synchronized formation control approach.

2 Preliminary-System Model and Problem Formulation

2.1 Robot Dynamics Adaptation

In this paper, we consider N mobile robots with the following dynamic model:

$$M_i(x_i)\ddot{x}_i + F_i(x_i)\dot{x}_i + D_i(x_i, \dot{x}_i) = u_{xi}, \quad (1)$$

where $M_i(x_i)$ represents the inertia matrix of robot i , $x_i = [x_{i1}, x_{i2}]^T$ denotes the position of robot i , $F_i(x_i)\dot{x}_i$ represents the friction/damping force, $D_i(x_i, \dot{x}_i)$ is the external force exerted on the robot i (such as external disturbances) and u_{xi} is the control input.

By the adaptive control concept [13], the dynamics (1) can be linearly parameterized as:

$$M_i(x_i)\ddot{x}_i + F_i(x_i)\dot{x}_i + D_i(x_i, \dot{x}_i) = Y_i(x_i, \dot{x}_i, \ddot{x}_i)\theta_i, \quad (2)$$

where $Y_i(x, \dot{x}, \ddot{x})$ is a regression matrix, $\theta_i(t)$ denotes the parameterized vector which contains system parameters in dynamic system (1).

In real applications, since the system parameters of mobile robot usually can not be accurately achieved and may change dynamically during the task accomplishment process, the estimation of the parameters are usually used for control purpose. Denote $\hat{(\cdot)}$ as the estimation of (\cdot) , (2) can be re-wrote as:

$$\hat{M}_i(x_i)\ddot{x}_i + \hat{F}_i(x_i)\dot{x}_i + \hat{D}_i(x_i, \dot{x}_i) = Y_i(x_i, \dot{x}_i, \ddot{x}_i)\hat{\theta}_i. \quad (3)$$

2.2 Synchronized Formation

In this paper, time-varying desired formations are considered for mobile robot network. The desired trajectory of robot i is denoted as x_i^d , and we assume that x_i^d has first- and second-order derivatives. Then the position error of robot i can be represented as $e_i(t) = x_i(t) - x_i^d(t)$.

According to [9], the synchronization constraint of robot formation can be generally presented as follows:

$$c_1(t)e_1(t) = c_2(t)e_2(t) = c_3(t)e_3(t) = \dots = c_N(t)e_N(t), \quad (4)$$

where $c_i(t)$ is a parameter matrix which describes the position constraint of robot i in the desired formation structures. Consequently, the synchronization error can be defined as [9]:

$$\varpi_i(t) = c_i(t)e_i(t) - c_{i+1}(t)e_{i+1}(t). \quad (5)$$

Then one can easily conclude that if $e_i(t) \rightarrow 0$ and $\varpi_i(t) \rightarrow 0$, the synchronized formation control goal achieves.

2.2.1 An Example Case

We choose a time-varying circle formation shape as an example to show the synchronized formation control concept. The desired position of robot i is:

$$x_i^d(t) = \begin{bmatrix} \cos \phi_i(t) & \\ & \sin \phi_i(t) \end{bmatrix} \begin{bmatrix} r(t) \\ r(t) \end{bmatrix},$$

where $\phi_i(t)$ describes the position of robot i and $r(t)$ denotes the time-varying radii of the circle formation. Then the synchronization constraint is:

$$\begin{bmatrix} \cos \phi_i(t) & \\ & \sin \phi_i(t) \end{bmatrix}^{-1} \begin{bmatrix} x_{i1} - x_{i1}^d \\ x_{i2} - x_{i2}^d \end{bmatrix} = \begin{bmatrix} \cos \phi_j(t) & \\ & \sin \phi_j(t) \end{bmatrix}^{-1} \begin{bmatrix} x_{j1} - x_{j1}^d \\ x_{j2} - x_{j2}^d \end{bmatrix},$$

and

$$c_i(t) = \begin{bmatrix} \cos \phi_i(t) & \\ & \sin \phi_i(t) \end{bmatrix}^{-1}.$$

3 Adaptive Control Design Under Communication Constraints

3.1 Controller Design

Similar to [12], we first design a coupling error as:

$$\varepsilon_i(t) = \varepsilon_{i1}(t) + \int_0^t \varepsilon_{i2}(\lambda) d\lambda, \tag{6}$$

where

$$\varepsilon_{i1}(t) = c_i(t)e_i(t) + \int_0^t k_1 c_i(\lambda)e_i(\lambda) d\lambda, \tag{7}$$

$$\varepsilon_{i2}(t) = k_2[\varpi_i(t) - \varpi_{i-1}(t)], \tag{8}$$

and k_1, k_2 are positive parameter matrices.

Now we take into account the practical communication constraints, i.e., we introduce the sampled data with time delays into the coupled synchronization error $\varepsilon_{i2}(t)$:

$$\varepsilon_{i2}(t) = k_2[\varpi_i(t_k) - \varpi_{i-1}(t_k)], \quad t_k \leq t - h(t) < t_{k+1}, \quad (9)$$

where t_k denotes the sampling instant with sampling period T and $h(t)$ represented the time-varying communication delay.

Similar to [14], the sampling instant t_k can be represented as:

$$t_k = t - (t - t_k) = t - f(t), \quad (10)$$

where $h(t) \leq f(t) < h(t) + T$, and if we assume that $h_1 \leq h(t) \leq h_2$ and $\dot{h}(t) \leq \tau$, then one has $h_1 \leq f(t) < h_2 + T$ and $1 \leq \dot{f}(t) \leq \tau + 1$. So (9) can be re-wrote as [12]:

$$\varepsilon_{i2}(t) = k_2[\varpi_i(t - f(t)) - \varpi_{i-1}(t - f(t))]. \quad (11)$$

For each robot, the adaptive controller is:

$$u_{xi} = \hat{M}_i(x_i)c_i^{-1}(-\dot{z}_i - \dot{c}_i\dot{x}_i - \dot{\varepsilon}_{i2} - k_d\dot{\varepsilon}_i - k_p\varepsilon_i) + \hat{F}_i(x_i)\dot{x}_i + \hat{D}_i(x_i, \dot{x}_i), \quad (12)$$

where $z_i = -c_i\dot{x}_i^d + \dot{c}_ie_i + k_1c_ie_i$, k_p and k_d are positive parameter matrices, and the adaptive control law is:

$$\dot{\hat{\theta}}_i = -(c_i\hat{M}_i^{-1}Y_i)^T\dot{\varepsilon}_i. \quad (13)$$

3.2 Stability Analysis

From the definitions of ε_i and z_i , one has:

$$\dot{\varepsilon}_i = c_i\dot{x}_i + z_i + \varepsilon_{i2}, \quad \ddot{\varepsilon}_i = \dot{c}_i\dot{x}_i + c_i\ddot{x}_i + \dot{z}_i + \dot{\varepsilon}_{i2}. \quad (14)$$

Combining (1) and (12), one has:

$$\begin{aligned} & M_i(x_i)\ddot{x}_i + F_i(x_i)\dot{x}_i + D_i(x_i, \dot{x}_i) \\ &= \hat{M}_i(x_i)c_i^{-1}(-\dot{z}_i - \dot{c}_i\dot{x}_i - \dot{\varepsilon}_{i2} - k_d\dot{\varepsilon}_i - k_p\varepsilon_i) + \hat{F}_i(x_i)\dot{x}_i + \hat{D}_i(x_i, \dot{x}_i). \end{aligned} \quad (15)$$

Define $\tilde{M}_i(x_i) = M_i(x_i) - \hat{M}_i(x_i)$, $\tilde{F}_i(x_i) = F_i(x_i) - \hat{F}_i(x_i)$, $\tilde{D}_i(x_i, \dot{x}_i) = D_i(x_i, \dot{x}_i) - \hat{D}_i(x_i, \dot{x}_i)$ and $\tilde{\theta}_i = \theta_i - \hat{\theta}_i$, then (2) and (3) leads to:

$$\tilde{M}_i(x_i)\ddot{x}_i(t) + \tilde{F}_i(x_i)\dot{x}_i + \tilde{D}_i(x_i, \dot{x}_i) = Y_i(x_i, \dot{x}_i, \ddot{x}_i)\tilde{\theta}_i. \quad (16)$$

Then (15) can be further represented as:

$$\tilde{M}_i(x_i)\ddot{x}_i + \tilde{F}_i(x_i)\dot{x}_i + \tilde{D}_i(x_i, \dot{x}_i) + \hat{M}_i(x_i)\ddot{x}_i = \hat{M}_i(x_i)c_i^{-1}(-\dot{z}_i - \dot{c}_i\dot{x}_i - \dot{\varepsilon}_{i2} - k_d\dot{\varepsilon}_i - k_p\varepsilon_i). \quad (17)$$

Combining (14) and (17), one has:

$$Y_i(x_i, \dot{x}_i, \ddot{x}_i)\tilde{\theta}_i + \hat{M}_i(x_i)c_i^{-1}\ddot{\epsilon}_i + \hat{M}_i(x_i)c_i^{-1}k_d\dot{\epsilon}_i + \hat{M}_i(x_i)c_i^{-1}k_p\epsilon_i = 0, \tag{18}$$

and thus

$$\ddot{\epsilon}_i = -k_d\dot{\epsilon}_i - k_p\epsilon_i - c_i\hat{M}_i^{-1}(x_i)Y_i(x_i, \dot{x}_i, \ddot{x}_i)\tilde{\theta}_i. \tag{19}$$

Proposition 1 *Considering the close-loop dynamic system (18) and the adaptive control law (13), the coupling error of each robot converges to zero asymptotically, i.e., $\epsilon_i(t) \rightarrow 0$ and $\dot{\epsilon}_i(t) \rightarrow 0$ as $t \rightarrow \infty$.*

Proof Define the Lyapunov function candidate as follows:

$$V_1 = \frac{1}{2}(\dot{\epsilon}_i^T \dot{\epsilon}_i + \epsilon_i^T k_p \epsilon_i + \tilde{\theta}_i^T \tilde{\theta}_i). \tag{20}$$

Differentiating V_1 with time t and substituting (19) yield:

$$\begin{aligned} \dot{V}_1 &= \dot{\epsilon}_i^T(-k_d\dot{\epsilon}_i - k_p\epsilon_i - c_i\hat{M}_i^{-1}Y_i\tilde{\theta}_i) + \epsilon_i^T k_p \dot{\epsilon}_i + \tilde{\theta}_i^T \dot{\tilde{\theta}}_i \\ &= -\dot{\epsilon}_i^T k_d \dot{\epsilon}_i - \dot{\epsilon}_i^T c_i \hat{M}_i^{-1} Y_i \tilde{\theta}_i + \tilde{\theta}_i^T \dot{\tilde{\theta}}_i. \end{aligned} \tag{21}$$

From (13) one has:

$$\dot{\tilde{\theta}}_i = (c_i \hat{M}_i^{-1} Y_i)^T \dot{\epsilon}_i. \tag{22}$$

Then combining (21) and (22), one has:

$$\dot{V}_1 = -\dot{\epsilon}_i^T k_d \dot{\epsilon}_i. \tag{23}$$

Since $k_d > 0$, one has $\dot{V}_1 < 0$, and consequently, by using the LaShall's invariance principle, one has $\epsilon_i(t) \rightarrow 0$, $\dot{\epsilon}_i(t) \rightarrow 0$ and $\tilde{\theta}_i \rightarrow 0$ as $t \rightarrow \infty$. The proof is completed.

Now we prove $e_i(t) \rightarrow 0$ and $\varpi_i(t) \rightarrow 0$. From (6), (7), (11) and $\dot{\epsilon}_i(t) \rightarrow 0$, one has:

$$\dot{\tilde{\zeta}} = -\bar{k}_1 \tilde{\zeta} + \bar{k}_2 \Phi \tilde{\zeta} (t - f(t)). \tag{24}$$

where $\tilde{\zeta} = [\tilde{\zeta}_1, \tilde{\zeta}_2, \dots, \tilde{\zeta}_N]^T$, $\tilde{\zeta}_i = c_i e_i$, $\bar{k}_1 = I_N \otimes k_1$, $\bar{k}_2 = I_N \otimes k_2$, \otimes represents the Kronecker product, $\mathbf{1}_n$ is the n dimensional identity matrix, and

$$\Phi = \begin{bmatrix} -2 & 1 & 0 & 0 & \dots & 1 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ & & & \ddots & & \\ 1 & 0 & 0 & \dots & 1 & -2 \end{bmatrix} \otimes I_2.$$

Similar to [12], define

$$M = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 \\ & & & \ddots & & \\ 0 & 0 & 0 & \dots & 1 & -1 \end{bmatrix} \otimes I_2,$$

one can conclude that $\varpi_i(t) \rightarrow 0$ equals to $M\zeta \rightarrow 0$, then we define the Lyapunov function candidate as follows:

$$\begin{aligned} V_2 = & \zeta^T M^T L M \zeta + \int_{t-h_1}^t \zeta^T(w) M^T E_1 M \zeta(w) dw \\ & + \int_{t-f(t)}^{t-h_1} \zeta^T(w) M^T E_2 M \zeta(w) dw + \int_{-h_1}^0 \int_{t+v}^t \zeta^T(w) M^T F_1 M \zeta(w) dw dv \quad (25) \\ & + \int_{-(h_2+T)}^{-h_1} \int_{t+v}^t \zeta^T(w) M^T F_2 M \zeta(w) dw dv, \end{aligned}$$

where L , E_1 , E_2 , F_1 , and F_2 are positive-definite matrices. Then using the similar method presented in our previous work [12] (in order to save spaces and avoid repetition, the detailed proof procedure is omitted in this paper), we have the following results:

Theorem 1 *The proposed approach achieves the synchronized formation control goals $e_i(t) \rightarrow 0$ and $\varpi_i(t) \rightarrow 0$, if the following Linear Matrix Inequality (LMI) condition holds:*

$$\Gamma = \begin{bmatrix} \Gamma_{11} & \frac{1}{h_1} F_1 & \Gamma_{13} & 0 \\ \frac{1}{h_1} F_1 & \Gamma_{22} & \frac{1}{\tilde{h}} F_2 & 0 \\ \Gamma_{13}^T & \frac{1}{\tilde{h}} F_2 & \Gamma_{33} & \frac{1}{\tilde{h}} F_2 \\ 0 & 0 & \frac{1}{\tilde{h}} F_2 & -\frac{1}{\tilde{h}} F_2 \end{bmatrix} < 0, \quad (26)$$

where $\Gamma_{11} = -L\tilde{k}_1 - \tilde{k}_1^T L + E_1 + h_1 \tilde{k}_1^T F_1 \tilde{k}_1 + \tilde{h} \tilde{k}_1^T F_2 \tilde{k}_1 - \frac{1}{h_1} F_1$, $\Gamma_{13} = L\tilde{k}_2 G - h_1 \tilde{k}_1^T F_1 \tilde{k}_2 G - \tilde{h} \tilde{k}_1^T F_2 \tilde{k}_2 G$, $\Gamma_{22} = -E_1 + E_2 - \frac{1}{h_1} F_1 - \frac{1}{\tilde{h}} F_2$, $\Gamma_{33} = \tau E_2 + h_1 G^T \tilde{k}_2^T F_1 \tilde{k}_2 G + \tilde{h} G^T \tilde{k}_2^T F_2 \tilde{k}_2 G - \frac{2}{\tilde{h}} F_2$, and $\tilde{h} = h_2 + T - h_1$, $\tilde{k}_1 = \mathbf{I}_{N-1} \otimes k_1$, $\tilde{k}_2 = \mathbf{I}_{N-1} \otimes k_2$, and $G \in \mathfrak{R}^{2(N-1) \times 2(N-1)}$ as:

$$G = \begin{bmatrix} -3 & 0 & -1 & -1 & \dots & -1 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ & & & \ddots & & \\ -1 & -1 & -1 & \dots & 0 & -3 \end{bmatrix} \otimes I_2. \quad (27)$$

4 Property Analyses and Discussions

In this section, we will theoretically analyze the effects of the sampling period and delay parameters on the stability of the control system and present several properties of the proposed sufficient condition.

4.1 Effects of T and h_2 on the Stability of Formation Control System

The LMI condition (26) in Theorem 1 can be re-wrote as:

$$\Gamma = \Xi - \frac{1}{h} W^T \bar{F}_2 W + \tilde{h} U^T F_2 U < 0, \quad (28)$$

where

$$\Xi = \begin{bmatrix} \Xi_{11} & \frac{1}{h_1} F_1 & \Xi_{13} & 0 \\ \frac{1}{h_1} F_1 & \Xi_{22} & 0 & 0 \\ \Xi_{13}^T & 0 & \Xi_{33} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad W^T = \begin{bmatrix} 0 & 0 \\ 0 & I \\ I & -I \\ -I & 0 \end{bmatrix}, \quad U^T = \begin{bmatrix} -\tilde{k}_1^T \\ 0 \\ G^T \tilde{k}_2^T \\ 0 \end{bmatrix},$$

$\Xi_{11} = -L\tilde{k}_1 - \tilde{k}_1^T L + E_1 + h_1 \tilde{k}_1^T F_1 \tilde{k}_1 - \frac{1}{h_1} F_1$, $\Xi_{13} = L\tilde{k}_2 G - h_1 \tilde{k}_1^T F_1 \tilde{k}_2 G$, $\Xi_{22} = -E_1 + E_2 - \frac{1}{h_1} F_1$, $\Xi_{33} = \tau E_2 + h_1 G^T \tilde{k}_2^T F_1 \tilde{k}_2 G$, and $\bar{F}_2 = \text{diag}\{F_2, F_2\}$. By implementing the Schur complement lemma [15], the above inequality (28) equals to the following inequality:

$$\begin{bmatrix} \Xi - \frac{1}{h} W^T \bar{F}_2 W & U^T F_2 \\ F_2 U & -\frac{1}{h} F_2 \end{bmatrix} < 0, \quad (29)$$

which further equals to:

$$\begin{bmatrix} \Xi & U^T F_2 \\ F_2 U & 0 \end{bmatrix} < \frac{1}{\tilde{h}} \begin{bmatrix} W^T \bar{F}_2 W & 0 \\ 0 & F_2 \end{bmatrix}. \quad (30)$$

Note that the sampling period T and the upper bound h_2 of time delay only appears in the right side of the above inequality, and more specially, they appear in pairs (with the form $h_2 + T$ in \tilde{h}). This implies that, if h_1 is fixed, the effects of T and h_2 on the stability of formation control system are similar to each other and the acceptable upper bound of the sum of these two parameters $(h_2 + T)_{\max}$ remains constant. This property can be demonstrated by the following example:

Assume that eight robots are commanded to maintain an octangle formation to transport a cargo along a cosine trajectory, i.e.,

Table 1 Upper bounds of acceptable time delay and sampling period (with $h_1 = 0.2$ s)

Upper bounds of acceptable h_2 with fixed T	
$T = 0.10$ s	$h_{2\max} = 0.462$ s
$T = 0.15$ s	$h_{2\max} = 0.412$ s
$T = 0.20$ s	$h_{2\max} = 0.362$ s
$T = 0.25$ s	$h_{2\max} = 0.312$ s
$T = 0.30$ s	$h_{2\max} = 0.262$ s
Upper bounds of acceptable T with fixed h_2	
$h_2 = 0.3$ s	$T_{\max} = 0.262$ s
$h_2 = 0.4$ s	$T_{\max} = 0.162$ s
$h_2 = 0.5$ s	$T_{\max} = 0.062$ s
$h_2 = 0.55$ s	$T_{\max} = 0.012$ s

$$x_i^d(t) = \begin{bmatrix} \cos \phi_i & \\ & \sin \phi_i \end{bmatrix} \begin{bmatrix} r \\ r \end{bmatrix} + \begin{bmatrix} t \\ \cos(t) \end{bmatrix},$$

where $\phi_i = (i - 1) \times 45^\circ + 22.5^\circ$, $r = 5$, $k_1 = \text{diag}\{1, 1\}$, $k_2 = \text{diag}\{0.5, 0.5\}$. In this case, we assume that the communication delay $h(t) \geq 0.2$ and $\dot{h}(t) \leq 0.3$, i.e., $h_1 = 0.2$ and $\tau = 0.3$. Then by iteratively testing the LMI-based sufficient condition (26), the upper bound of time delay $h_{2\max}$ and sampling period T_{\max} can be achieved and are shown in Table 1. From Table 1 we can find that $(h_2 + T)_{\max}$ remains constant and is equal to 0.562.

This property is very useful in real applications. Since in practice, in order to save the energy and communication resources (such as communication bandwidth), we usually hope that the data exchanged between coupling robots can be as little as possible, i.e., the sampling period T is desired to be as large as possible. However, the time delay parameters in real communication systems always change dynamically. A large sample period may lead to an unstable control system under large communication delays. Under the above property, if the control system can realize the real-time estimation of the delay parameter h_2 , then in order to guarantee the stability, the system can adjust the real-time sampling period T accordingly. So this property will greatly facilitate the application of the proposed approach and ensure the desired performance of synchronized formation control.

4.2 Effects of τ on the Stability of Formation Control System

From an intuitive perspective, the effect of τ (the upper bound of the derivative of time delay) on the stability of the control system should be very large, the larger τ should leads to the smaller upper bounds of acceptable time delay and sampling period. However, we find that, if the upper bounds of time delay h_2 is fixed, the

derivative τ has no effect on the stability of the formation control system, which means that our proposed LMI condition is a delay-derivative-free condition. This can be explained by investigating the structure of the proposed LMI condition. Since in (26), τ only appears in Γ_{33} with the form τE_2 , and furthermore, except in Γ_{33} , E_2 only appears in Γ_{22} with the form $+E_2$, these imply that the LMI condition (26) in Theorem 1 can be re-wrote as:

$$\Gamma = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} & \Gamma_{13} & 0 \\ \Gamma_{12}^T & \bar{\Gamma}_{22} & \Gamma_{23} & 0 \\ \Gamma_{13}^T & \Gamma_{23}^T & \bar{\Gamma}_{33} & \Gamma_{34} \\ 0 & 0 & \Gamma_{34}^T & \Gamma_{44} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & E_2 & 0 & 0 \\ 0 & 0 & \tau E_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} < 0, \tag{31}$$

where $\bar{\Gamma}_{22} = -E_1 - \frac{1}{h_1}F_1 - \frac{1}{h}F_2$, $\bar{\Gamma}_{33} = h_1G^T\tilde{k}_2^T F_1\tilde{k}_2G + \tilde{h}G^T\tilde{k}_2^T F_2\tilde{k}_2G - \frac{2}{h}F_2$,

this further implies that no matter how large the value of τ is, there always exist a sufficient small E_2 , such that the LMI condition (31) holds, i.e., the LMI condition (26) holds.

This property can also be demonstrated by the above example: we set $h_1 = 0.2$ s and $h_2 = 0.4$ s, then by solving the condition (26), the upper bounds of sampling period T_{\max} can be achieved and are shown in Table 2, we can find that T_{\max} remains constant. The matrix E_2 solved under $\tau = 0.3$ and $\tau = 10$ are shown as follows:

$$E_2|_{\tau=0.3} = \begin{bmatrix} 0.4600 & 0.1994 & -0.0798 & -0.0531 & 0.0189 & -0.0359 & -0.2895 \\ 0.1994 & 0.5908 & 0.2256 & -0.1502 & -0.0097 & -0.0600 & -0.0359 \\ -0.0798 & 0.2256 & 0.6115 & 0.1951 & -0.1380 & -0.0097 & 0.0189 \\ -0.0531 & -0.1502 & 0.1951 & 0.5852 & 0.1951 & -0.1502 & -0.0531 \\ 0.0189 & -0.0097 & -0.1380 & 0.1951 & 0.6115 & 0.2256 & -0.0798 \\ -0.0359 & -0.0600 & -0.0097 & -0.1502 & 0.2256 & 0.5908 & 0.1994 \\ -0.2895 & -0.0359 & 0.0189 & -0.0531 & -0.0798 & 0.1994 & 0.4600 \end{bmatrix} \otimes I_2$$

Table 2 Upper bounds of acceptable sampling period

With $h_1 = 0.2$ s and $h_2 = 0.4$ s	
$\tau = 0.3$	$T_{\max} = 0.162$ s
$\tau = 0.5$	$T_{\max} = 0.162$ s
$\tau = 1.0$	$T_{\max} = 0.162$ s
$\tau = 10$	$T_{\max} = 0.162$ s

$$E_2|_{\tau=10} = \begin{bmatrix} 0.0424 & 0.0208 & -0.0077 & -0.0074 & 0.0011 & -0.0056 & -0.0278 \\ 0.0208 & 0.0555 & 0.0231 & -0.0143 & -0.0034 & -0.0059 & -0.0056 \\ -0.0077 & 0.0231 & 0.0593 & 0.0222 & -0.0108 & -0.0034 & 0.0011 \\ -0.0074 & -0.0143 & 0.0222 & 0.0582 & 0.0222 & -0.0143 & -0.0074 \\ 0.0011 & -0.0034 & -0.0108 & 0.0222 & 0.0593 & 0.02317 & -0.0077 \\ -0.0056 & -0.0059 & -0.0034 & -0.0143 & 0.0231 & 0.0555 & 0.0208 \\ -0.0278 & -0.0056 & 0.0011 & -0.0074 & -0.0077 & 0.0208 & 0.0424 \end{bmatrix} \otimes I_2$$

we can find that $E_2|_{\tau=10} \ll E_2|_{\tau=0.3}$, this implies that the larger τ will lead to a smaller E_2 , however, the stability of the formation control system will not be affected.

This property will also greatly facilitate the real applications of the proposed method. Since in practice, due to external disturbances, communication noise and data packet dropout, the time delay may change rapidly and dynamically, i.e., τ may change dynamically and can not be estimated accurately. However, in the proposed approach, one only need to estimate the upper bound of the time delay h_2 , which changes more slowly and is more easy to be estimated.

4.3 Effects of h_1 on the Stability of Formation Control System

In this paper, we assume that $h_1 \leq h(t) < h_2$, i.e., we divide communication delay $h(t)$ into two parts, where h_1 represents the constant part and $h_2 - h_1$ represents the time-varying part. Except the upper bound h_2 , the constant part h_1 (i.e., the lower bound) will also affect the stability of the control system. More specifically, if h_2 is fixed, the larger h_1 will lead to the larger upper bound of acceptable sampling period T_{\max} . This means that if the constant part in time delay is larger (i.e., the time-varying part in time delay is smaller), the system will more easy to be stable. This property can also be demonstrated by the above example: we set $h_2 = 0.55$ s, then by solving the condition (26), the upper bound of sampling period T_{\max} can be achieved and are shown in Table 3. We can find that T_{\max} increases with the increasing of h_1 , and when $h_1 = h_2 = 0.55$ s (i.e., the time delay is reduced to a constant value), the upper bound of sampling period T_{\max} reaches the maxima. So in practice, we hope that the time-varying part in time delay is as small as possible.

Table 3 Upper bounds of acceptable sampling period

With $h_2 = 0.55$ s	
$h_1 = 0.20$ s	$T_{\max} = 0.012$ s
$h_1 = 0.30$ s	$T_{\max} = 0.056$ s
$h_1 = 0.40$ s	$T_{\max} = 0.105$ s
$h_1 = 0.50$ s	$T_{\max} = 0.157$ s
$h_1 = 0.55$ s	$T_{\max} = 0.184$ s

5 Simulations

We use numerical simulations to validate the proposed adaptive synchronized formation control approach. Eight robots are required to transport an octangle cargo through an arc trajectory. Without motion synchronization, the cargo may be damaged. And the external force exerted on the robot (from the weight of the cargo) will

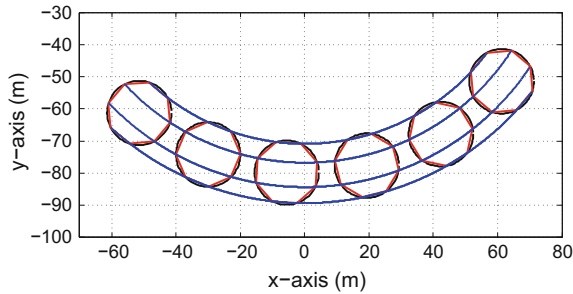


Fig. 1 Real formation shapes and the real trajectories of each robot in the simulation, where the *red octangles* represent the formation shapes and the *blue lines* represent the trajectories

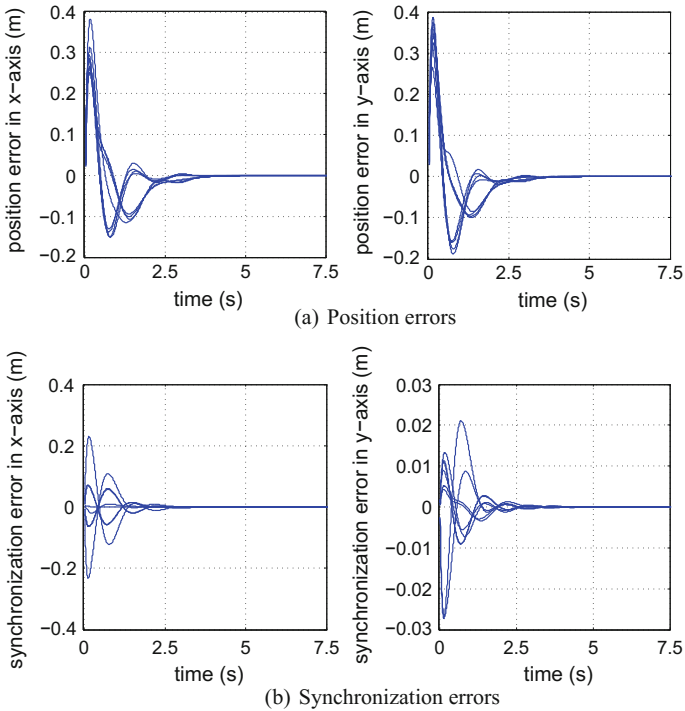


Fig. 2 Simulation results

also change its dynamics parameters. So in order to accomplish this task, the robots should synchronize their motion with their neighbors adaptively.

The desired trajectory of robot i is:

$$x_i^d(t) = \begin{bmatrix} \cos(\phi(t) - 90^\circ) & -\sin(\phi(t) - 90^\circ) \\ \sin(\phi(t) - 90^\circ) & \cos(\phi(t) - 90^\circ) \end{bmatrix} \begin{bmatrix} \cos \psi_i \\ \sin \psi_i \end{bmatrix} \begin{bmatrix} R \\ R \end{bmatrix} + \begin{bmatrix} \cos \phi(t) \rho \\ \sin \phi(t) \rho \end{bmatrix},$$

where $0 \leq t \leq 15s$, $\phi(t) = -\frac{1}{30}\pi t - 40^\circ$, $R = 10m$, $\rho = 80m$, $\psi_i = 22.5^\circ + (i - 1) \times 45^\circ$, $M_i = \text{diag}\{0.5 + 0.05i, 0.5 + 0.05i\}$, $k_1 = \text{diag}\{1, 1\}$, $k_2 = \text{diag}\{0.5, 0.5\}$, $k_p = \text{diag}\{30, 30\}$, $k_d = \text{diag}\{10, 10\}$. The initial estimation of the parameterized vector $\hat{\theta}_i(0) = [0; 0]$, the time delay $h(t) = 0.1 + 0.1 \times (1 + \sin(0.5t))s$ and the sampling period $T = 0.05$ s.

Figure 1 shows the real formation shapes in each 3 s as well as the real trajectories of each robots in the simulation. Figure 2 shows the simulation results, we can find that, the proposed formation controller can estimate the dynamics parameters adaptively and converge both the synchronization errors and position errors to zero asymptotically, i.e., the synchronized formation control task is achieved even in the presence of system parameter uncertainties and communication constraints.

6 Conclusion

In this paper, an adaptive control approach is proposed to resolve the synchronized formation control problem under parameter uncertainties and communication constraints. A distributed controller is presented which enables multiple mobile robots to maintain the desired time-varying formations synchronously. A sufficient condition as well as several useful properties are proposed to ensure the convergence of the control system, these properties will greatly facilitate the practical applications of the proposed approach. The results of numerical simulations demonstrate that the proposed approach achieves adaptive and synchronized formation control tasks in the presence of system parameter uncertainties and communication constraints.

Acknowledgements This work is partially supported by the China Domestic Research Project for the International Thermonuclear Experimental Reactor (ITER) under Grant 2012GB102001, the Natural Science Foundation of China under Grants 61175088, 61221003 and 61573243.

References

1. Burgard, W., Moors, M., Stachniss, C., Schneider, F.E.: Coordinated multi-robot exploration. *IEEE Trans. Robot.* **21**(3), 376–386 (2005)
2. Lavaei, J., Momeni, A., Aghdam, A.G.: A model predictive decentralized control scheme with reduced communication requirement for spacecraft formation. *IEEE Trans. Control Syst. Technol.* **16**(2), 268–278 (2008)

3. Lawton, J.R.T., Beard, R.W., Young, B.J.: A decentralized approach to formation maneuvers. *IEEE Trans. Robot. Autom.* **19**(6), 933–941 (2003)
4. Ahmad, S., Feng, Z., Hu, G.Q.: Multi-robot formation control using distributed null space behavioral approach. In: *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 3607–3612. IEEE (2014)
5. Beard, R.W., Lawton, J., Hadaegh, F.Y., et al.: A coordination architecture for spacecraft formation control. *IEEE Trans. Control Syst. Technol.* **9**(6), 777–790 (2001)
6. Yoo, S.J., Park, J.B., Choi, Y.H.: Adaptive formation tracking control of electrically driven multiple mobile robots. *IET Control Theory Appl.* **4**(8), 1489–1500 (2010)
7. Panagou, D., Kumar, V.: Cooperative visibility maintenance for leader-follower formations in obstacle environments. *IEEE Trans. Rob.* **30**(4), 831–844 (2014)
8. Takahashi, H., Nishi, H., Ohnishi, K.: Autonomous decentralized control for formation of multiple mobile robots considering ability of robot. *IEEE Trans. Industr. Electron.* **51**(6), 1272–1279 (2004)
9. Sun, D., Wang, C., Shang, W., Feng, G.: A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations. *IEEE Trans. Rob.* **25**(5), 1074–1086 (2009)
10. Zhao, D.Y., Zou, T.: A finite-time approach to formation control of multiple mobile robots with terminal sliding mode. *Int. J. Syst. Sci.* **43**(11), 1998–2014 (2012)
11. Chung, S.J., Slotine, J.J.: Cooperative robot control and concurrent synchronization of lagrangian systems. *IEEE Trans. Rob.* **25**(3), 686–700 (2009)
12. Liu, Z., Chen, W.D., Lu, J.G., Wang, H.S., Wang, J.C.: Formation control of mobile robots using distributed controller with sampled-data and communication delays. *IEEE Trans. Control Syst. Technol.* **24**(6), 2125–2132 (2016)
13. Cheah, C.C., Hou, S.P., Slotine, J.J.E.: Region-based shape control for a swarm of robots. *Automatica* **45**(10), 2406–2411 (2009)
14. Lu, J.G., Hill, D.J.: Global asymptotical synchronization of chaotic lur'e systems using sampled data: a linear matrix inequality approach. *IEEE Trans. Circuits Syst. II Express Briefs* **55**(6), 586–590 (2008)
15. Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: Linear matrix inequalities in system and control theory. *SIAM* **15** (1994)

A Distributed Self-healing Algorithm for Global Optimal Movement Synchronization of Multi-robot Formation Network

Xiangyu Fu, Weidong Chen, Zhe Liu, Jingchuan Wang
and Hesheng Wang

Abstract When multiple robots cooperate to carry out some tasks, the movement synchronization of the formation is essential. However, due to the uncertainty of the complicated environment, it is inevitable that certain robot fails. Therefore, it's worth to figure out some ways to minimize the damage caused by failed robot. In this paper, a distributed self-healing algorithm for multi-robot network global optimal movement synchronization is presented. The proposed algorithm can transform any robot's failure into the failure of the robot which has the least degree so that can minimize the influence on movement synchronization. In addition, gradient is involved in the algorithm to make repairing path shortest under the condition of maintaining global optimal movement synchronization. In the whole process of the self-healing, all robots only communicate with their own neighbors, so the robots are distributed controlled. Finally, the effectiveness of the proposed algorithm is validated by simulation experiments.

Keywords Multi-robot formation · Self-Healing · Movement synchronization · Global optimal · Distributed-control

1 Introduction

As the application of robot becomes more and more popular, single robot cannot meet the task requirement due to the limits of ability and cost. So, multi-robot has caused the extensive concern. Multi-robot network is often applied to environment exploration [1], disaster reconnaissance [2], search and rescue [3]. In these tasks,

X. Fu (✉) · W. Chen (✉) · Z. Liu · J. Wang · H. Wang
Key Laboratory of System Control and Information Processing,
Department of Automation, Ministry of Education of China,
Shanghai Jiao Tong University, Shanghai 200240, China
e-mail: fuxiangyu@sjtu.edu.cn

W. Chen
e-mail: wdchen@sjtu.edu.cn

it's inevitable that certain robot fails in severe environment. Therefore, in order to improve the robust of the system, network should have self-healing ability to minimize the influence caused by failed robot.

At present, existing self-healing algorithms for multi-robot and wireless sensor network system have mainly three types. The first type is called direct self-healing algorithm [4]. It points some robot in the network to fill the vacancy caused by the failed robot directly. This kind of algorithm needs a global decision maker. So, it's not totally distributed. The second one is called density self-healing algorithm [5]. Its implementation depends on the node density and it's often applied to multi-robot morphological formation and area coverage tasks. This algorithm needs a global parameter (e.g. node density) to restore the sensing dead zone caused by nodes failure. However, this algorithm mainly focuses on multi-robot system or sensor application which referred to morphological formation and area coverage and do not apply to the dynamic process of robot formation. The third type is called recursive self-healing algorithm [6, 7], which makes use of the interaction among robots and network topology switching to implement self-healing. All of above algorithms mainly concern connectivity or vertex distribution of multi-robot or sensor network. They cannot get good result when applied to dynamic problem which need to keep certain formation in the process of task. Such as multi-robot cooperative carrying, multi-UAV formation. In these problems, the movement synchronization of the multi-robot network is an essential index, and if certain robot fails, how to implement self-healing to minimize the influence on the system movement synchronization is a major issue to be addressed. In [8, 9], Zhang has proposed a recursive self-healing algorithm based on distributed topology switching control which focuses on the multi-robot movement synchronization. However, it still contains some flaws like that the improvement of synchronization is not optimal, too much robots involved into the self-healing process and temporary extension of the communication range.

Above all, it's worth to design a totally distributed self-healing algorithm which implements global optimal movement synchronization on multi-robot formation network and contains less robots. For this purpose, this paper introduces a distributed self-healing algorithm for global optimal movement synchronization of multi-robot formation network. The proposed algorithm makes use of the degree information of the topology network to implement gradient generation and diffusion so as to form stable gradient distribution in the multi-robot formation network. The gradient can not only reflect the different synchronization effects from different robots but also help to plan a shortest repairing path under the circumstance of the implement of global optimal movement synchronization.

The contribution of this paper includes: (1) Propose a distributed self-healing algorithm for global optimal movement synchronization of multi-robot network which can improve the movement synchronization to the greatest extent after certain robot failed. (2) Propose gradient generation and update rules which makes repairing path and repairing time shortest under the condition of global optimal movement synchronization. (3) Implement totally distributed control on individual robot. (4) Check algorithm performance through quantitative repairing index and validate the effectiveness through simulation experiments.

This paper is organized as follows: next section introduces the network model, topology analysis and some lemmas which describe the topology network structure and performance; Sect. 3 describes the details of the proposed algorithm; Sect. 4 analyses the feasibility of algorithm; Sect. 5 validates the algorithm effectiveness by simulation experiments; Sect. 6 concludes this paper.

2 Network Model and Topology Analysis

2.1 Modeling of Multi-robot Formation Network

Graph theory is adopted to express topology model [8]. Suppose that there are n robots in the formation, Graph G indicates the multi-robot formation network made by n robots, $G = (V, E)$, V indicates the set of vertexes, E indicates the set of edges. Topology vertex $v_i \in V, i = 1, 2, \dots, n$ indicates robot R_i , $(v_i, v_j) \in E, i, j = 1, 2, \dots, n$ indicates the connectivity between robot R_i and R_j . Coupling matrix $M = (m_{ij}), m_{ij} \in R^{(n \times n)}$ indicates the coupling relationship among all robots in the topology network, therein,

$$m_{ij} = m_{ji} = \begin{cases} 1, & (v_i, v_j) \in E \\ 0, & (v_i, v_j) \notin E \end{cases}, i \neq j. \quad (1)$$

diagonal element $m_{ii} = -\sum_{i=1, i \neq j}^n m_{ij}$. The robot that has connectivity with robot R_i is the neighbor of robot R_i . Define the neighbor set of robot R_i as

$$N_e(R_i) = \{\forall R_j \mid j \neq i, \text{ and } m_{ij} = 1\}. \quad (2)$$

Degree of robot R_i is defined as $d_i = \sum_{R_j \in N_e(R_i)} m_{ij}$, so the diagonal element of coupling matrix $m_{ii} = -d_i$.

2.2 Topology Analysis of Multi-robot Formation Network

In [10], Zhang etc. use the second largest eigenvalue λ_2 of the coupling matrix M to measure the stability and robustness of network movement synchronization. Smaller λ_2 means better stability and robustness of network movement synchronization. They have also studied the different influence on network movement synchronization caused by different robot's failure and introduced the topology switching rules for multi-robot formation self-healing:

Lemma 1 *In multi-robot formation network, the system movement synchronization would be improved if the failed robot which has higher degree is replaced by lower degree robot.*

Lemma 2 *The more the degree of repairing robot is increased, the better the system synchronization is improved.*

The proposed algorithm is designed mainly based on these two lemmas.

According to above two lemmas, if we want self-healing algorithm to achieve global optimal movement synchronization, we have to use least degree robot in the formation to repair the failed robot so that synchronization can be improved most. In addition we have to make the amount of repairing robots least. As shown in Fig. 1, all the black nodes have least degree. If the robot drawn by dotted line is failed, according to Lemma 2 if we use any one of the black nodes to replace the failed one, we will get global optimal movement synchronization improvement. But not all black nodes have the shortest distance away from the failed robot. So if we use the robot around by red dotted line circle to repair, then the self-healing is global optimal (Assuming that the robot around by red circle has the shortest distance away from failed robot).

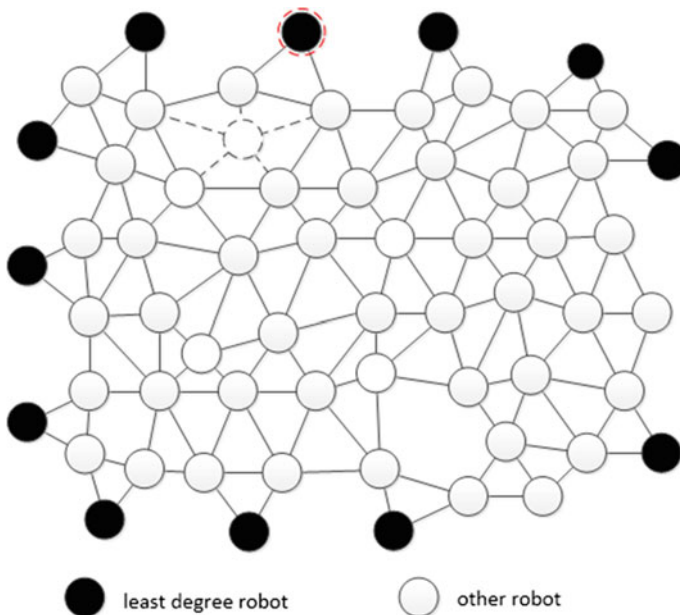


Fig. 1 The least degree node distribution in the formation topology, *black node* means the least degree robot, and *white node* means other robot, the node drawn by *dotted line* means the failed robot

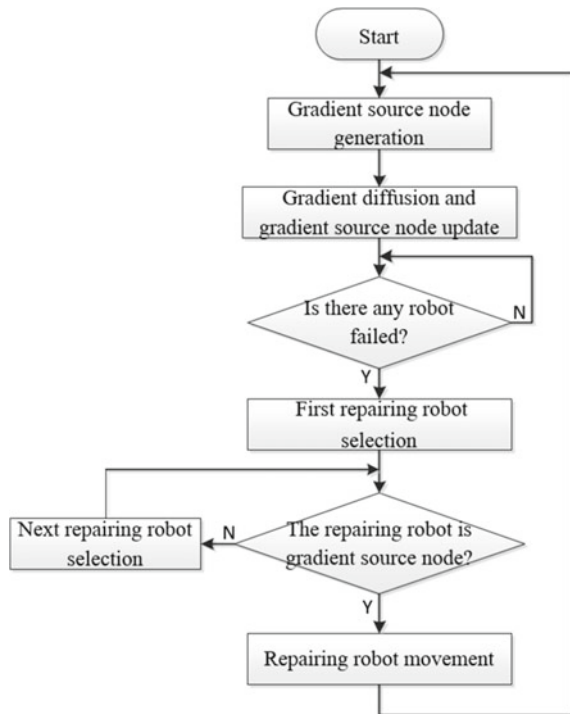
3 Distributed Self-healing Algorithm

In order to achieve global optimal movement synchronization, the easiest way is to make the least degree robot fill the vacancy left by failed robot directly. However, the proposed algorithm is totally distributed, it means that all robots can just communicate with their neighbors. So the direct way is impossible. Instead, we adopt recursive self-healing method which recursively selects one of neighbors as the repairing robot to repair the current vacancy. Eventually, it can get the same effect as direct way. Figure 2 shows the control flow of the distributed self-healing algorithm.

3.1 Gradient Source Node Generation

According to Lemmas 1 and 2, global optimal movement synchronization improvement depends on whether we can find the robot which has least degree. So how to find least degree robot in the network should be first to think about. As we know, gradient diffusion is a common method to estimate distributed distance which is extensively applied to problems such as sensor network node localization [11] and multi-robot self-organization [12, 13]. In [11–13] the gradient starts from predefined root node

Fig. 2 The control flow of the distributed self-healing algorithm



and diffuses to neighbors which make it possible for every node to calculate its distance from the root node. In this paper we also use gradient diffusion but do not need to predefine root node.

From Fig. 1, we can find out that all the least degree nodes have less degree than their neighbors. Therefore we select robots which has less degree than all its neighbors as gradient source node, from which gradient starts and then the process of gradient diffusion and gradient source node update which will be introduced in the following section can exclude some gradient source nodes which do not have global least degree.

Definition 1 A robot is gradient source node if it has less degree than all of its neighbors, which means if

$$d_i < \forall d_j, R_j \in N_e(R_i), d_i \in S \tag{3}$$

then S is the set of gradient source node.

3.2 Gradient Diffusion and Gradient Source Node Update

Figure 3 shows the topology network in which black node means the gradient source node. We let gradient value of gradient source node equals to zero ($d_s = 0$), the other

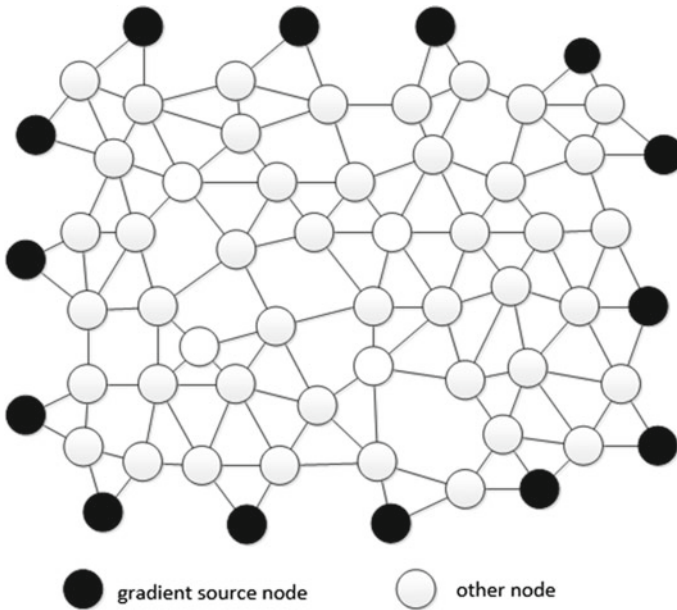


Fig. 3 Topology network which points out the gradient source node. The *black node* is local least degree node which is seem as gradient source node

robots have a very large initial gradient value. From Fig. 3 we can see that, not all of the gradient source nodes have least degree, some have degree two and others have degree three. We let gradient value diffuse from gradient source node and form stable gradient distribution in the network eventually. In the diffusion process the interaction message between neighbors is expressed by a data packet $w = [d_s, g_s]$, g_s means the robot gradient, d_s means the degree of gradient source node from which g_s is diffused. Message diffusion and update formula is shown as follows: $\forall i \in N_c(R)$, $\delta \geq 0$, $\Delta \geq 0$, $\Delta \gg \delta$,

$$w_R(t) = \begin{cases} \left[\begin{array}{l} d_{s_i}(t-1) \\ g_{s_i}(t-1) + \Delta \end{array} \right], d_{s_R}(t-1) > d_{s_i}(t-1) \vee (d_{s_R}(t-1) = d_{s_i}(t-1) \wedge g_{s_R}(t-1) + \delta > g_{s_i}(t-1) + \Delta) \\ \left[\begin{array}{l} d_{s_R}(t-1) \\ g_{s_R}(t-1) + \delta \end{array} \right], d_{s_R}(t-1) < d_{s_i}(t-1) \vee (d_{s_R}(t-1) = d_{s_i}(t-1) \wedge g_{s_R}(t-1) + \delta \leq g_{s_i}(t-1) + \Delta) \end{cases} \quad (4)$$

therein:

t – time, unit time means the time that message passed to its neighbors.

$w_R(t)$ – data packet of robot R at time t.

$d_{s_i}(t-1)$ – the gradient source node degree belongs to robot i at time t-1.

$d_{s_R}(t-1)$ – the gradient source node degree belongs to robot R at time t-1.

$g_{s_i}(t-1)$ – the gradient value of robot i at time t-1.

$d_{s_R}(t-1)$ – the gradient value of robot R at time t-1.

δ – robot gradient increment of unit time.

Δ – the gradient increment between neighbors.

The gradient diffusion and update process is explained as follows according to above formula:

1. After robot received message from neighbors, it compares the message with its current message. If the gradient source node degree it received is smaller than its own, then update its own message to the received one.

2. If the gradient source node degree it received equals to its own, then compare the gradient value of the message packet. If its own gradient value is larger than the result that the received gradient value plus the gradient increment between neighbors, then update its own gradient value to the result.

Figure 4 has shown the process of gradient generation and diffusion. In Fig. 4a the robots check itself whether it is gradient source node. If it is, set its gradient to zero and then pack its degree and gradient into a data packet. In Fig. 4b gradient is diffused from gradient source node to other robots according to above message diffusion and update formula. In Fig. 4c when a node whose gradient source node degree is not global least receive message from its neighbor whose gradient source node degree is global least, then it updates its data packet according to above formula. In addition if its gradient source node degree is already global least, but its gradient is larger than the result that the gradient value it received plus gradient increment it also updates. Finally the whole formation network constructs stable gradient distribution. Figure 5 is a formation network formed by 62 robots which has already had the stable gradient distribution after gradient diffusion and update. It shows that all robots have the same gradient source node degree which is the global least. It also means all the robots

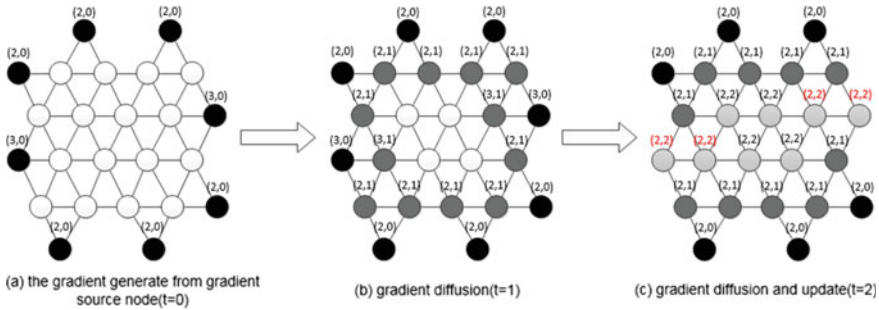


Fig. 4 The process of gradient generation and diffusion, the number pair over the node is $w_R(t)$ and the different shades means different gradient value, the darker the shade is the smaller the gradient value is

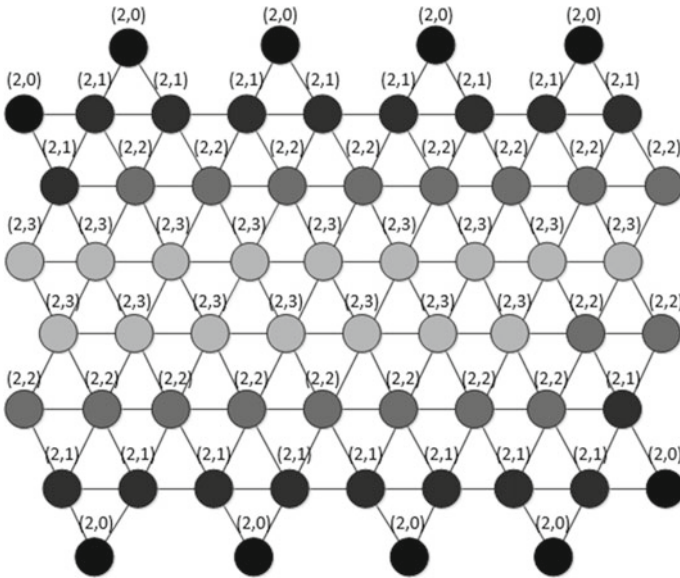


Fig. 5 The stable gradient distribution of the formation network after gradient diffusion and update

gradient is diffused from global least degree node. According to above process of gradient generation and diffusion, two propositions are concluded:

Proposition 1 *When multi-robot formation network has constructed stable gradient distribution, the gradient source node degree of all robots is the global least one.*

Proof As shown before, global least degree node must be gradient source node. We use proof by contradiction. Assumes that after network gradient distribution has been stable, the gradient source node degree of robot R_i is larger than global least degree node. Then, according to message diffusion and update formula, R_i updates its own

message when it receives neighbor's message. As a consequence, the network gradient distribution is not stable.

Proof complete.

Proposition 2 *When multi-robot formation network has constructed stable gradient distribution, all robots except gradient source node must have some neighbors whose gradient value is smaller than itself.*

Proof From the gradient diffusion and update process, we can know that the gradient diffusion is started from gradient source node, and all robots update its gradient from one of its neighbors. According to update rule, robot gradient must be the result that one of its neighbors' gradient value plus gradient increment between neighbors.

Proof complete.

3.3 Repairing Robots Selection

According to Proposition 1, when multi-robot formation network has constructed stable gradient distribution, all robots have the same gradient source node degree. Therefore, the repairing robot selection only bases on gradient value. The selection and movement process is similar to the method in [14] except that selection criteria has been changed to gradient only. The selection rules in this paper are as followed:

Rule 1: In the neighbors of failed robot, if a robot has smaller gradient value than failed robot, then it is a candidate for repairing robot.

Rule 2: Among candidates of repairing robots, the robot who has least gradient value is selected as the repairing robot.

Rule 3: If there are more than one robot that have least gradient value among candidates, then select the one whose ID is smallest as the repairing robot.

4 Analysis of Movement Synchronization

Proposition 3 *When there is a failed robot in multi-robot formation network, the proposed algorithm must be able to select a sequence of repairing robots in which the last one is the global least degree robot and it is equivalent to using global least degree robot to repair the failing robot.*

Proof Assume that the sequence of repairing robots is $\{R_{f1}, R_{f2}, \dots, R_{fn}\}$, R_{fn} is the last repairing robot. If R_{fn} is not the global least degree robot, then $d_{(R_{fn})} \neq 0$. From Proposition 2 we can know that R_{fn} must have a neighbor whose gradient is smaller than R_{fn} and according to the Nth repairing robot selection, R_{fn} will continue to select the next repairing robot until there is no robot whose gradient is less than itself in its neighbors. However, only the global least degree robot has no neighbors whose gradient is less than itself.

Proof complete.

Theorem 1 *Under the condition of stable gradient distribution, the proposed algorithm must be able to find a shortest path which can guarantee the global optimal movement synchronization improvement.*

Proof Using induction method, if the failed robot is gradient source node, then there exists no repairing robot, so the repairing path is sure to be shortest. According to Proposition 3, the repairing robots sequence contains limited repairing robots. Assume that the repairing robot sequence is $\{R_{f1}, R_{f2}, \dots, R_{fn-1}, R_{fn}\}$ and the path between R_{f1} and R_{fn-1} is the shortest. The following will prove the path between R_{fn-1} and R_{fn} is already the shortest. From Proposition 2 we can know that R_{fn} has at least one neighbor whose gradient is smaller than itself. In addition, according to gradient diffusion and update rules, the largest gradient difference between R_{fn} and its neighbors is Δ . In the repairing robot selection process, every selected repairing robot must be the one whose gradient is the smallest among neighbors. Therefore, the gradient difference between R_{fn} and R_{fn-1} is the largest. So the path between R_{fn} and R_{fn-1} is shortest.

Proof complete.

Above all, the proposed algorithm must be able to find a shortest path under the condition that can guarantee the global optimal movement synchronization.

5 Simulation

In simulation, 115 robots form a 6-neighbors topology network which is like Fig. 4 shown. In order to validate the feasibility and performance of proposed algorithm, the algorithm will be compared with the random-based self-healing algorithm with no gradient in [8]. Two algorithms will be applied repeatedly to above multi-robot formation network topology respectively. The simulation experiments has been executed ten times which is presented by simulation ID. The simulation results and analysis are as follows.

As shown in Fig. 6, simulation experiments has been conducted ten times. Every time we make the same robot fail. From Fig. 6 we can see that the proposed algorithm has significant advantage on movement synchronization improvement. In every experiment, the proposed algorithm has least value of second largest eigenvalue which also means it is stable. In contrast, ten results of random-based self-healing algorithm [8] vary randomly. Because there is no gradient distribution in random-based self-healing algorithm, so the algorithm cannot locate the global least degree node. As a result, the final repairing robot can only be local least degree node. Therefore, the effect of movement synchronization improvement is limited.

Table 1 has shown the comparative result of repairing robots amount. Random-based self-healing algorithm selects the least degree neighbor randomly as the repairing robot. Therefore, the number of repairing robots has much uncertainty. In addition the final repairing robot may be located at local least degree node which has less

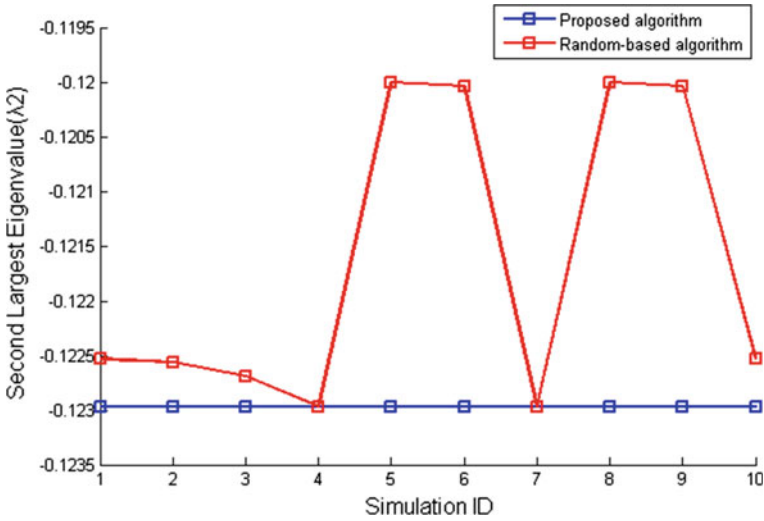


Fig. 6 Second largest eigenvalue comparative result

Table 1 Amount of repairing robots

Simulation ID	1	2	3	4	5	6	7	8	9	10
Proposed algorithm	4	4	4	4	4	4	4	4	4	4
Random-based algorithm	8	4	3	4	7	4	4	7	4	8

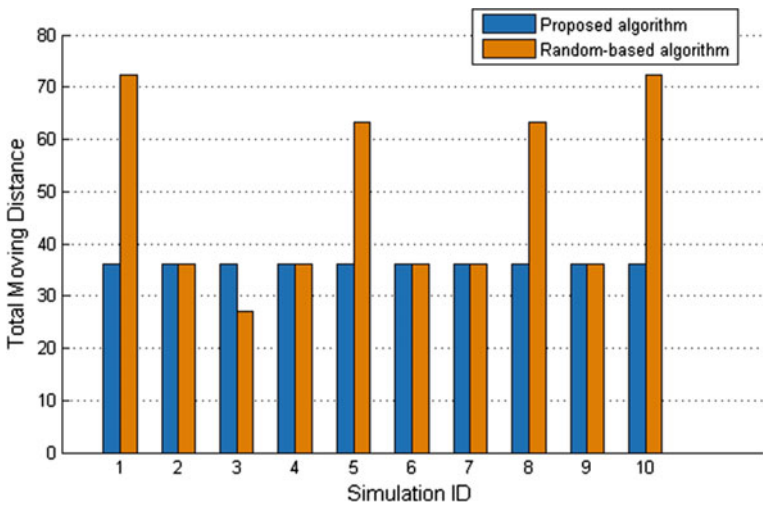


Fig. 7 Total moving distance comparative result

repairing robots number, but it sacrifices the movement synchronization improvement effect. Both of the proposed algorithm and random-based self-healing algorithm adopt recursive moving method which is that next repairing robot fills the vacancy previous repairing robot left until the final one. And all repairing robots move simultaneously. Every robot's moving distance is the distance of two adjacent robots. So the total moving distance is in proportion to the number of repairing robots. So the proposed algorithm also has advantage on total moving distance compared to random-based self-healing in most circumstance, as shown in Fig. 7.

6 Conclusion

In this paper, a distributed control self-healing algorithm for multi-robot network global optimal movement synchronization has been proposed. More specifically, gradient distribution and diffusion rules has been proposed to locate the global least degree robot in the formation. Based on the gradient distribution the proposed algorithm can improve formation movement synchronization at the most extend when some robot fail and it can also maintain repairing robots least. Finally simulation results show that the proposed algorithm has advantage of movement synchronization improvement and less repairing robots.

Acknowledgements This work is partially supported by the China Domestic Research Project for the International Thermonuclear Experimental Reactor (ITER) under Grant 2012GB102001, the Natural Science Foundation of China under Grants 61175088, 61221003 and 61573243.

References

1. Wang, B., Qin, S.: Multi-robot environment exploration based on label maps building via recognition of frontiers. In: IEEE International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI), pp.1–9 (2014)
2. Schneider, F.E., Welle, J., Wildermuth, D., Ducke, M.: Unmanned multi-robot CBRNE reconnaissance with mobile manipulation System description and technical validation. In: IEEE 13th International on Carpathian Control Conference (ICCC), pp.637–642 (2012)
3. Sugiyama, H., Tsujioka, T., Murata, M.: QoS Routing in a multi-robot network system for urban search and rescue. In: IEEE International Conference on Advanced Information Networking & Applications, pp. 323–330 (2006)
4. Corke, P., Hrabar, S., Peterson, R., Rus, D., Saripalli, S.: Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In: IEEE International Conference on Robotics and Automation(ICRA), pp. 3602–3608 (2004)
5. Arbuckle, D.J., Requicha, A.A.G.: Self-assembly and self-repair of arbitrary shapes by a swarm of reactive robots: algorithms and simulations. *Auton. Robot.* **28**(2), 197–211 (2009)
6. Abbasi, A.A., Younis, M., Akkaya, K.: Movement-assisted connectivity restoration in wireless sensor and actor networks. *IEEE Trans. Parallel Distrib. Syst.* **20**(9), 1366–1379 (2008)
7. Younis, M., Lee, S., Abbasi, A.A.: A localized algorithm for restoring internode connectivity in networks of moveable sensors. *IEEE Trans. Comput.* **59**(12), 1669–1682 (2010)

8. Zhang, F., Chen, W.D.: Self-healing for mobile robot networks with motion synchronization. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3107–3112 (2007)
9. Zhang, F., Chen, W.D.: Switched topology control for self-healing of mobile robot formation. In: Control Theory & Applications (2010)
10. Zhang, F., Chen, W.D., Xi, Y.: Motion synchronization in mobile robot networks: Robustness. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5570–5575 (2006)
11. Nagpal, R., Shrobe, H., Bachrach, J.: Organizing a global coordinate system from local information on an ad-hoc sensor network. In: Information Processing in Sensor Networks, Second International Workshop, pp. 333–348 (2003)
12. Rubenstein, M., Cornejo, A., Nagpal, R.: Robotics. Programmable self-assembly in a thousand-robot swarm. *Science* **345**(6198), 795–799 (2014)
13. Stoy, K.: Using cellular automata and gradients to control self-reconfiguration. *Robot. Auton. Syst.* **54**(2), 135–141 (2006)
14. Liu, Z., Ju, J.J., Chen, W.D., Fu, X.Y., Wang, H.S.: A gradient-based self-healing algorithm for mobile robot formation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2015)

Consensus of Discrete-Time Linear Networked Multi-agent Systems Subject to Actuator Saturation

DU Boyang, Zhang Guoliang, XU Jun, Zeng Jing and Zhang Yong

Abstract In this paper, we investigate the consensus problem of multi agent systems based on the discrete-time general linear model with actuator saturation. A new family of distributed parametric low-and-high gain control protocols is provided, and such control protocols rely on the discrete-time Parametric Algebraic Riccati Equation. Furthermore, the convergence results are also given as consensus function sequences. Finally, an illustrative example shows that the low-and-high gain control protocol is effective for the consensus problem of the general linear multi agent systems with actuator saturation.

Keywords Consensus · Discrete-time · Multi-agent systems · Actuator saturation

1 Introduction

Recently, the consensus problems of networked multi-agent systems (NMAS) has received great attention and been widely applied in cooperative control, such as formation control [1, 2], flocking [3, 4], distributed filtering [5, 6], synchronization of coupled chaotic oscillators [7–9] and so on. The main work of the consensus problem is to make all agents agree on some common values of interest by designing protocols based on the information of its neighbors.

The theoretical framework for posing and solving the consensus problem for NMAS was first introduced in [10–12]. Their work mostly focused on the first-order and second consensus in NMAS. The union of interaction topologies must contain a spanning tree if the NMAS is expected to achieve consensus asymptotically was developed in [13]. A framework of high-dimensional state

D. Boyang (✉) · Z. Guoliang (✉) · X. Jun · Z. Jing · Z. Yong
High-Tech Institute of Xi'an, Xi'an 710025, People's Republic of China
e-mail: huiyinniaoniao@163.com; duboyangepegc@gmail.com

Z. Guoliang
e-mail: zhgl@sohu.com

space for the consensus problems of NMAS was studied in [14], and then the consensus problems of high order or more general linear models NMAS were investigated in [15–17]. The consensus problem of the general linear model discrete-time NMAS (D-NMAS) was investigated in [18, 19]. The leader-following consensus problem of general linear model D-NMAS was studied in [20]. The distributed output feedback consensus problem of general linear model D-NMAS both fixed topology and stochastic switching topology was investigated in [21].

Actuator saturation is one of the most ubiquitous and important nonlinearities existing in the practical control system since the capability of any physical actuator is always limited to a finite range, and thus actuator saturation is very important to study. However, as to our knowledge, most consensus literature does not consider the case where the agents are subject to actuator saturation. The protocol design for achieving consensus in the case subject to actuator saturation is a challenging problem, and only few results in recently are available for NMAS. For the continues-model, the consensus problems of linear continues-time model NMAS with leader and without leader subject to actuator saturation were investigated in [22, 23]. For discrete-time case, the consensus problem of D-NMAS with actuator saturation was studied in [24], which assumed that the control input should be bounded with standard saturation function. This is not always consistent with the real situation, which motivates our study in this paper.

Based on the aforementioned discussion, we contempt to study the problem of consensus analysis for a general linear D-NMAS subject to actuator saturation. In the scenario where fixed undirected graph, the dynamics of each agent is considered for any order. Furthermore, the convergence results are also given as consensus function sequences. An illustrative example shows that the low-and-high gain control protocols are effective for the consensus problem of the general linear multi agent systems with actuator saturation.

The rest of the paper is organized as follows. The Preliminary of graph theory and problem formulation are presented in Sect. 2. A Scheduled low-and-high-gain feedback is designed in Sect. 3, and the consensus convergence result of D-NMAS is also provided by a consensus function sequence. In Sect. 4, a numerical example is given to verify the theoretical analysis. Some conclusions are finally drawn in Sect. 5 concludes the paper and proposes some possible future directions.

2 Preliminary of Graph Theory and Problem Formulation

2.1 Preliminary of Graph Theory

Let a weighted digraph (or directed graph) $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ of order N represents an interaction topology of a network of agents, with the set of nodes $\mathcal{V} = \{v_1, \dots, v_N\}$, set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and a weighted adjacency matrix $\mathcal{A} = [a_{ij}]$ with nonnegative adjacency elements a_{ij} .

The node indexes belong to a finite index set $\mathcal{I} = \{1, 2, \dots, N\}$. An edge of G is denoted by $e_{ij} = (v_i, v_j)$, where v_i and v_j are called the initial and terminal nodes. It implies that node v_j can receive information from node v_i , but not necessarily vice versa. The adjacency elements associated with the edges of the graph are positive if $e_{ij} \in \mathcal{E}$ while $a_{ij} = 0$ if $e_{ij} \notin \mathcal{E}$. Furthermore, we assume $a_{ii} = 0$ for all $i \in \mathcal{I}$. The set of neighbors of node v_i is denoted by $N_i = \{v_j \in \mathcal{V}: (v_i, v_j) \in \mathcal{E}\}$. A cluster is any subset $J \subseteq \mathcal{V}$ of the nodes of the graph. The set of neighbors of a cluster N_J is defined by $N_J = \cup_{v_i \in J} N_i = \{v_j \in \mathcal{V}: v_i \in J, (v_i, v_j) \in \mathcal{E}\}$. The in-degree and out-degree of node v_i are defined as $\text{deg}_{\text{in}}(v_i) = \sum_{j=1}^n a_{ji}$ and $\text{deg}_{\text{out}}(v_i) = \sum_{j=1}^n a_{ij}$ respectively. The degree matrix of the digraph G is a diagonal matrix $D = [D_{ij}]$, where

$$D_{ij} = \begin{cases} 0 & i \neq j \\ \text{deg}_{\text{out}}(v_i) & i = j \end{cases}$$

The graph Laplacian matrix associated with the digraph G is defined as $\mathcal{L}(G) = L = D - A$.

Lemma 1 ([25]) Let L be the Laplacian matrix of an undirected G . Then zero is a simple eigenvalue of L , and all the other eigenvalues of L are positive and real if and only if G is connected.

2.2 Problem Formulation

Consider a D-NMAS consisting N identical agents, in which the dynamics of agent $i (i = 1, \dots, N)$ is described by the following discrete-time linear system subject to actuator saturation.

$$\mathbf{x}_i(k + 1) = A\mathbf{x}_i(k) + B\sigma(\mathbf{u}_i(k)), \tag{1}$$

where, $\mathbf{u}_i(k) \in \mathbb{R}^m$ and $\mathbf{x}_i(k) \in \mathbb{R}^d$ are the control input and state vectors of agent i , respectively. A and B are the state and input matrices of compatible dimensions, and $\sigma(\bullet)$ denotes the standard saturation function defined as

$$\sigma(\mathbf{u}_i) = [\sigma(u_{i,1}), \sigma(u_{i,2}), \dots, \sigma(u_{i,m})]^T,$$

where

$$\sigma(u_{i,j}) = \text{sign}(u_{i,j}) \min\{|u_{i,j}|, \Delta\} \quad j = 1, 2, \dots, m$$

for some $\Delta > 0$.

For the D-NMAS (1), we use the following distributed linear consensus protocol

$$\mathbf{u}_i(k) = F \sum_{j=1}^N a_{ij}(\mathbf{x}_j(k) - \mathbf{x}_i(k)), \quad (2)$$

where F is a fixed feedback control gain matrix with appropriate dimensions and independent of agent index i .

Definition 1 ([26]) Given matrices $P = (p_{ij})_{n \times n} \in \mathbb{R}^{m \times n}$ and $Q = (q_{ij})_{n \times n} \in \mathbb{R}^{p \times q}$, their Kronecker product is defined as

$$P \otimes Q = [p_{ij}Q] \in \mathbb{R}^{mp \times nq}.$$

Lemma 2 ([26]) For matrices A, B, C and D , with appropriate dimensions, we have the following conditions.

- (1) $(\gamma A) \otimes B = A \otimes (\gamma B)$, where γ is a constant.
- (2) $(A + B) \otimes C = A \otimes C + B \otimes C$.
- (3) $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$.
- (4) $(A \otimes B)^T = A^T \otimes B^T$.

Definition 2 For a given gain matrix F of protocol (2), D-NMAS (1) is said to achieve consensus if for any given bounded initial condition, there exists a vector-valued function sequence $\mathbf{c}(k)$ dependent on the initial condition such that

$$\lim_{k \rightarrow \infty} (\mathbf{x}(k) - \mathbf{1} \otimes \mathbf{c}(k)) = \mathbf{0},$$

where $\mathbf{c}(k)$ is called a consensus function sequence.

3 Main Results

In this section, we would like to extend the low-and-high-gain design approach introduced earlier in [22] for discrete-time linear systems with actuator saturation to NMAS with actuator saturation, and provide a new family of scheduled low-and-high-gain control protocols that can be applied to the D-NMAS (1), and make D-NMAS (1) reach consensus.

3.1 Scheduled Low-and-High-Gain Feedback Design

Before design the low-and-high-gain feedback matrix, the following two assumptions are necessarily given to solve the consensus problem of D-NMAS (1) via protocol (2).

Assumption 1 The graph G is connected.

Lemma 3 ([27]) If matrices A and B satisfy the Assumption 2, then for any $\varepsilon \in (0, 1)$, the discrete-time Parametric Algebraic Riccati Equation (D-PARE) defined as

$$(1 - \varepsilon)P_\varepsilon = A^T P_\varepsilon A - A^T P_\varepsilon B (I + B^T P_\varepsilon B)^{-1} B^T P_\varepsilon A, \tag{3}$$

has a unique positive definite solution $P_\varepsilon = W_\varepsilon^{-1}$, where W_ε is the solution for W of $W - \frac{1}{1-\varepsilon}AWA^T = -BB^T$.

Moreover, this positive definite solution P_ε also holds the following properties.

- (1) for any $\varepsilon \in (0, 1)$, $A_c(\varepsilon) = A - B(I + B^T P_\varepsilon B)^{-1} B^T P_\varepsilon A$ is Schur stable.
- (2) for any $\varepsilon \in (0, 1)$, $\frac{dP_\varepsilon}{d\varepsilon} > 0$.
- (3) $\lim_{\varepsilon \rightarrow 0^+} P_\varepsilon = 0$.

Define $\delta_i = \sum_{j=1}^N a_{ij}(\mathbf{x}_j(k) - \mathbf{x}_i(k))$. For the system (1), a low-gain state feedback, which is a parameterized state feedback laws, is provided by

$$\begin{aligned} \tilde{\mathbf{u}}_{L,i} &= F_L \delta_i = F_{\varepsilon(\delta_i)} \delta_i \\ &= \frac{1}{\lambda_N} (I + B^T P_{\varepsilon(\delta_i)} B)^{-1} B^T P_{\varepsilon(\delta_i)} A \delta_i, \end{aligned} \tag{4}$$

where $(i=1, 2, \dots, N)$, and P_ε is the solution of D-PARE (3), and $\varepsilon(\delta_i)$ is the low-gain parameter. From the properties given by Lemma 3, it is easy to see that the norm of the feedback gain matrix F_L can be made arbitrarily by choosing $\varepsilon(\delta_i)$ sufficiently small so that the control input $\tilde{\mathbf{u}}_{L,i}$ never saturates for any initial conditions $\delta_i(0)$, and the scheduled parameter $\varepsilon(\delta_i)$ ($i=1, 2, \dots, N$) should such that the following properties.

- (1) $\varepsilon(\delta_i): \mathbb{R}^d \rightarrow (0, \varepsilon^*]$ is piecewise differentiable with continuous, where ε^* is a priori design parameter.
- (2) There exists an open neighborhood \mathcal{D} of the origin such that $\varepsilon(\delta_i) = 1$, for all $\delta_i \in \mathcal{D}$.
- (3) For any $\delta_i \in \mathbb{R}^d$, we have $\|F_{\varepsilon(\delta_i)} \delta_i\| \leq \Delta$.
- (4) $\varepsilon(\delta_i) \rightarrow 0$ as $\|\delta_i\| \rightarrow \infty$.
- (5) $\{\delta_i \in \mathbb{R}^d | \delta_i^T P_{\varepsilon(\delta_i)} \delta_i \leq c\}$ is a bounded set for all $c > 0$.

$$(6) \quad \varepsilon(\delta_i) = \max \left\{ r \in (0, \varepsilon^*] | \delta_i^T P_{\varepsilon(\delta_i)} \delta_i \text{trace}(P_r) \leq \frac{\Delta^2}{b} \right\} \tag{5}$$

where $\varepsilon^* \in (0, 1)$ is any a priori given constant, and $b = 2\text{trace}(BB^T)$ when P_r is the unique positive definite solution of D-PARE (3) with $\varepsilon = r$.

For the system (1), the scheduled low-and-high-gain state feedback is composed of a low gain state feedback and a high-gain state feedback defined as

$$\tilde{\mathbf{u}}_{LH,i} = \tilde{\mathbf{u}}_{L,i} + \tilde{\mathbf{u}}_{H,i} = (F_L + F_H)\boldsymbol{\delta}_i = F_{LH}\boldsymbol{\delta}_i = \frac{1}{\lambda_N}(1 + \rho)(I + B^T P_e B)^{-1} B^T P_e A \boldsymbol{\delta}_i \quad (6)$$

where, the low-gain state feedback is given by (4), the high-gain feedback is of the form, $\tilde{\mathbf{u}}_{H,i} = F_H \boldsymbol{\delta}_i = \rho F_L \boldsymbol{\delta}_i$, where ρ is called the high-gain parameter, which satisfies $\rho \in [0, \frac{2}{\|B^T P_e B\|}]$.

3.2 Consensus with Low-and-High-Gain Feedback

Substituting protocol (2) into system D-NMAS (1), the closed-loop system of D-NMAS (1) is

$$\mathbf{x}_i(k+1) = A\mathbf{x}_i(k) + B\sigma\left(F_{LH} \sum_{j=1}^N a_{ij}(\mathbf{x}_j(k) - \mathbf{x}_i(k))\right) \quad (7)$$

where $(i = 1, 2, \dots, N)$.

Let $\mathbf{x}(k) = [\mathbf{x}_1(k), \mathbf{x}_2(k), \dots, \mathbf{x}_N(k)]^T$, the closed-loop system (7) can be written in a vector form as

$$\mathbf{x}(k+1) = (I_N \otimes A)\mathbf{x}(k) - (I_N \otimes B)\sigma((L \otimes F_{LH})\mathbf{x}(k)) \quad (8)$$

Let $\lambda_i (i = 1, 2, \dots, N)$ be eigenvalues of the Laplacian matrix $L \in \mathbb{R}^N$ for a undirected topology G , where $\lambda_1 = 0$ with the associated eigenvector $\bar{\mathbf{u}}_1 = \frac{1}{\sqrt{N}}\mathbf{1}_N$, and $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$. There exists a an orthogonal matrix

$$U = \begin{bmatrix} \frac{1}{\sqrt{N}} & \frac{\mathbf{1}_{N-1}^T}{\sqrt{N}} \\ \frac{\mathbf{1}_{N-1}^T}{\sqrt{N}} & U \end{bmatrix}$$

such that $U^T L U = D_L = \text{diag}\{0, \lambda_1, \lambda_2, \dots, \lambda_N\}$ is a diagonal matrix.

Let $\tilde{\mathbf{x}}(k) = (U^T \otimes I_d)\tilde{\mathbf{x}}(k) = [\tilde{\mathbf{x}}_1^T(k), \tilde{\mathbf{x}}_2^T(k), \dots, \tilde{\mathbf{x}}_N^T(k)]^T$, then the closed-loop discrete-time system (8) is equivalent to

$$\tilde{\mathbf{x}}(k+1) = (I_N \otimes A)\tilde{\mathbf{x}}(k) - (U^T \otimes I_d)^*(I_N \otimes B)\sigma((U \otimes I_d)(D_L \otimes F_{LH})\tilde{\mathbf{x}}(k)) \quad (9)$$

It is easy to see this is a nonlinear process, therefore, we consider the following assumption.

Assumption 2 For the closed-loop system (9), choose the sufficient small ε , such that

$$\sigma((U \otimes I_d)(D_L \otimes F_{LH})\tilde{\mathbf{x}}(k)) = (U \otimes I_d)\sigma((D_L \otimes F_{LH})\tilde{\mathbf{x}}(k)) \quad (10)$$

Based on Assumption 2, and by Lemma 3, the system (9) can be rewritten as

$$\tilde{\mathbf{x}}(k+1) = (I_N \otimes A)\tilde{\mathbf{x}}(k) - (I_N \otimes B)\sigma((D_L \otimes F_{LH})\tilde{\mathbf{x}}(k)) \quad (11)$$

Based on Assumption 1, and by Lemma 1, the system (11) can be represented by

$$\tilde{\mathbf{x}}_{c,1}(k+1) = A\tilde{\mathbf{x}}_{c,1}(k) \quad (12)$$

$$\tilde{\mathbf{x}}_{r,i}(k+1) = A\tilde{\mathbf{x}}_{r,i}(k) - B\sigma(\lambda_i F_{LH}\tilde{\mathbf{x}}_{r,i}(k)) \quad (13)$$

where, $i = 2, 3, \dots, N$. If system (13) is stable, system (9) can reach consensus.

Theorem 1 Consider a D-MAS of N identical agents, Assume that the Assumptions of 1 and 2 are satisfied, then the protocol (6) is a solution of the consensus problem for D-NMAS (1).

Proof Before to prove the Theorem 1, we define a Lyapunov function given by

$$V(\delta_i) = \sum_{i=1}^N \delta_i^T P_\varepsilon \delta_i = \sum_{i=2}^N V_i(\tilde{\mathbf{x}}_{r,i}) = \sum_{i=2}^N \lambda_i^2 \tilde{\mathbf{x}}_{r,i}^T P_\varepsilon \tilde{\mathbf{x}}_{r,i} \quad (14)$$

and a level set $V(s) = \{\delta_i | V_i(\delta_i) < s, i = 1, 2, \dots, N\}$.

From Lemma 3, we can know that there exists an $\varepsilon \in (0, \varepsilon_1]$ and $\delta_i \in V(s)$, such that $\|(I + B^T P_\varepsilon B)^{-1} B^T P_\varepsilon A \delta_i\| \leq \Delta$.

Define $\mu(k) = \|B^T P_\varepsilon(k) B\|$. Noticing that $\|(I + B^T P_\varepsilon B)^{-1} B^T P_\varepsilon A \delta_i\| \leq \Delta$, we have $\|\lambda_i \tilde{\mathbf{u}}_{L,i}\| \leq \Delta$ for all $k > 0$, and there has $\|\lambda_i \tilde{\mathbf{u}}_{L,i}(k)\| \leq \|\sigma(\lambda_i \tilde{\mathbf{u}}_{LH,i}(k))\| \leq (1 + \frac{2}{\mu(k)}) \|\lambda_i \tilde{\mathbf{u}}_{L,i}(k)\|$.

Then we have

$$\begin{aligned} \Delta V_i(k) &= V_i(k+1) - V_i(k) \\ &\leq -\varepsilon(k)V_i(k) + \tilde{\mathbf{x}}_{r,i}^T(k+1)(P_\varepsilon(k+1) \\ &\quad - P_\varepsilon(k))\tilde{\mathbf{x}}_{r,i}(k+1). \end{aligned} \quad (15)$$

The Property (4) of Lemma 1 which implies that $V_i(k+1) - V_i(k)$ and $\tilde{\mathbf{x}}_{r,i}^T(k+1)(P_\varepsilon(k+1) - P_\varepsilon(k))\tilde{\mathbf{x}}_{r,i}(k+1)$ can not have the same sign. To see this, we assume that $V_i(k+1) > V_i(k)$ and $P_\varepsilon(k+1) > P_\varepsilon(k)$.this implies that $\varepsilon(k) < \varepsilon^*$. If $V_i(k)\text{trace}(P_\varepsilon(k)) < \frac{\Delta^2}{b}$, then we have $\varepsilon(k) = \varepsilon^*$, which yield a contradiction. If $V_i(k)\text{trace}(P_\varepsilon(k)) = \frac{\Delta^2}{b}$, then $V_i(k+1)\text{trace}(P_\varepsilon(k+1)) > \frac{\Delta^2}{b}$ because of the assumption that $V_i(k+1) > V_i(k)$ and $P_\varepsilon(k+1) > P_\varepsilon(k)$. However, this is impossible by our scheduling (5). It is similar to that $V_i(k+1) - V_i(k) < 0$ and

$P_e(k+1) - P_e(k) < 0$ may not occur simultaneously either. According to above analysis, (15) implies that for all $\tilde{\mathbf{x}}_{r,i} \neq 0$, we have $V_i(k+1) - V_i(k) < 0$.

Thus, system (1) reaches consensus with low-and-high-gain feedback protocol (6).

The proof of Theorem 1 is completed.

Theorem 2 *If the closed-loop D-NMAS (7) achieves consensus, then the consensus function sequence $\mathbf{c}(k)$ such that*

$$\mathbf{c}(k) = \mathbf{1}_N \otimes \left(A^k \left(\frac{1}{N} \sum_{i=1}^N \mathbf{x}_i(0) \right) \right). \quad k=0, 1, 2, \dots \quad (16)$$

Proof Let $\mathbf{x}_C(k) = (U \otimes I_d) [\tilde{\mathbf{x}}_C^T(k), 0]^T$ and $\mathbf{x}_{\bar{C}}(k) = (U \otimes I_d) [0, \tilde{\mathbf{x}}_1^T(k)]^T$, then $\mathbf{x}(k)$ can be uniquely decomposed as $\mathbf{x}(k) = \mathbf{x}_C(k) + \mathbf{x}_{\bar{C}}(k)$. As discussed above, we can know that if the system (7) achieves consensus, the subsystem (13) should be stable, which means that the response of system (7) due to $\mathbf{x}_{\bar{C}}(0)$ should satisfy $\lim_{k \rightarrow \infty} \mathbf{x}_{\bar{C}}(k) = 0$. Hence the consensus function sequence $\mathbf{c}(k)$ is determined solely upon $\mathbf{x}_C(k)$. Since $[\tilde{\mathbf{x}}_C^T(k), 0]^T = \mathbf{e}_1 \otimes \tilde{\mathbf{x}}(k)$, we have $\mathbf{x}_C(0) = \bar{\mathbf{u}}_1 \otimes \tilde{\mathbf{x}}_1(0) = \bar{\mathbf{u}}_1 \otimes ((\mathbf{e}_1^T \otimes I_d) \tilde{\mathbf{x}}(0))$, and because $\tilde{\mathbf{x}}(0) = (U^T \otimes I_d) \mathbf{x}(0)$, then we can obtain $\mathbf{x}_C(0) = \bar{\mathbf{u}}_1 \otimes \tilde{\mathbf{x}}_1(0) = \bar{\mathbf{u}}_1 \otimes ((\mathbf{e}_1^T \otimes I_d) \tilde{\mathbf{x}}(0))$, that is to say

$$\begin{aligned} \mathbf{x}_C(0) &= \bar{\mathbf{u}}_1 \otimes ((\mathbf{e}_1^T \otimes I_d) * (U^T \otimes I_d) \mathbf{x}(0)) \\ &= \bar{\mathbf{u}}_1 \otimes (\mathbf{e}_1^T U^T \otimes I_d) \mathbf{x}(0) \\ &= \bar{\mathbf{u}}_1 \otimes \left(\frac{1}{\sqrt{N}} \mathbf{1}_N^T \otimes I_d \right) \mathbf{x}(0) \\ &= \mathbf{1}_N^T \otimes \left(\frac{1}{N} \sum_{i=1}^N \mathbf{x}_i(0) \right). \end{aligned}$$

Hence we have $\mathbf{x}_C(k) = A^k \mathbf{x}_C(0) = \mathbf{1}_N \otimes \left(A^k \left(\frac{1}{N} \sum_{i=1}^N \mathbf{x}_i(0) \right) \right)$, Then the consensus function sequence of D-MAS (1) is $\mathbf{c}(k) = \mathbf{1}_N \otimes \left(A^k \left(\frac{1}{N} \sum_{i=1}^N \mathbf{x}_i(0) \right) \right) \quad k=0, 1, 2, \dots$

The proof of Theorem 2 is completed. \blacksquare

4 Simulations

In this section, a numerical example is given to illustrate the effectiveness of the proposed theoretical results. We apply the above proposed consensus protocol to achieve state alignment among 8 agents. The dynamics of them are described by (1), where

$$A = \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ -\sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.3 \\ -0.4 \\ 1 \end{bmatrix}$$

We apply the consensus protocol (2) to achieve consensus among above those 8 agents under a fixed topology G , which is showed in Fig. 1.

Let $W_\epsilon = \begin{bmatrix} w_{11}^\epsilon & w_{12}^\epsilon & w_{13}^\epsilon \\ * & w_{22}^\epsilon & w_{23}^\epsilon \\ * & * & w_{33}^\epsilon \end{bmatrix}$, by solving the D-PARE (3), then one can obtain

$$w_{12}^\epsilon = \frac{(1-\epsilon)(0.25 + 0.02\epsilon - 0.48\epsilon^2)}{4\epsilon^3 - 8\epsilon^2 + 10\epsilon}, \quad w_{33}^\epsilon = \frac{1-\epsilon}{\epsilon},$$

$$w_{13}^\epsilon = \frac{(1-\epsilon)(-0.3 + 0.35\sqrt{2} + 0.3\epsilon)}{\epsilon(2-\epsilon)}, \quad w_{23}^\epsilon = \frac{0.8(1-\epsilon) - \sqrt{2}w_{13}^\epsilon}{2(1-\epsilon) - \sqrt{2}}$$

$$w_{22}^\epsilon = \frac{(2-\epsilon)w_{12}^\epsilon + 0.04(1-\epsilon)}{1/2 + \epsilon}, \quad w_{11}^\epsilon = 2w_{12}^\epsilon + (1-2\epsilon)w_{22}^\epsilon + 0.32(1-\epsilon).$$

Choosing $\epsilon = 0.5$, $\Delta = 1.5$ and $\rho = \frac{1}{\|B^T P_\epsilon B\|}$, by the Theorem 1, we can get

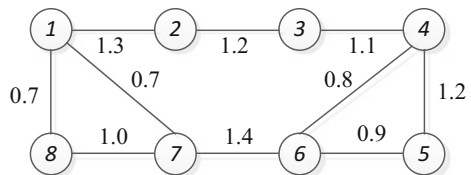
$$F_{LH} = \frac{1}{\lambda_N} (1 + \rho)(I + B^T P_\epsilon B)^{-1} B^T P_\epsilon A$$

$$= [0.2089, 0.0958, 0.0270].$$

Assume that the state initial values of the all agents $1, \dots, 8$ are randomly produced with $x_1(0) = [-4.6, 3.2, -1.5]^T$, $x_2(0) = [-2.5, -3.6, -2.7]^T$, $x_3(0) = [4.2, 1.5, 2.8]^T$, $x_4(0) = [-1.6, 2.9, -2.6]^T$, $x_5(0) = [1.2, 3.7, 1.4]^T$, $x_6(0) = [3.7, -1.6, 0.6]^T$. Each agent uses the protocol (5), and the simulation results are shown in Figs. 2, 3, 4 and 5.

In Figs. 2, 3 and 4, the state trajectories of D-MAS (1) are given. The trajectories of the saturated control signals of each agent are show in Fig. 5. It can be seen that the state trajectories of D-MAS (1) are asymptotically coverage to the common value $c(k)$, which are produced by Theorem 2 and marked by red asterisk, and the control inputs of D-MAS (1) are bounded. Therefore, the correctness of Theorems 1 and 2 are demonstrated, and which indicates that D-MAS (1) can achieve consensus with protocol (5) subject to actuator saturation.

Fig. 1 The interaction topology of 6 agents



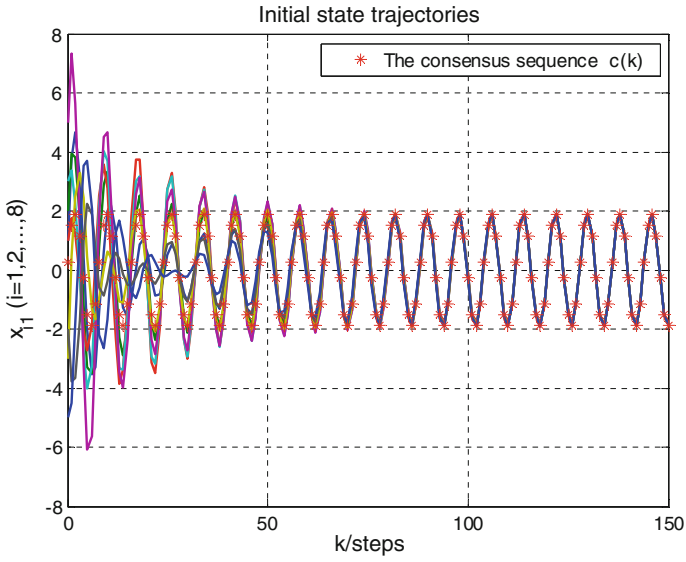


Fig. 2 The state 1 trajectories of D-NMAS (1)

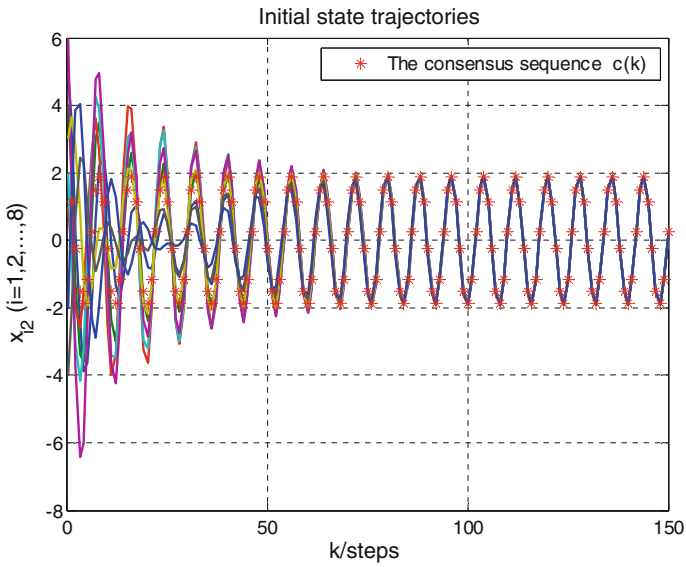


Fig. 3 The state 2 trajectories of D-NMAS (1)

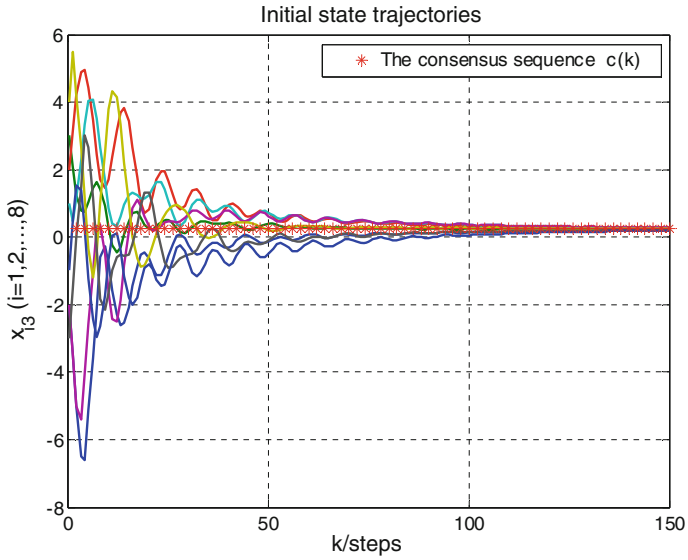


Fig. 4 The state 3 trajectories of D-NMAS (1)

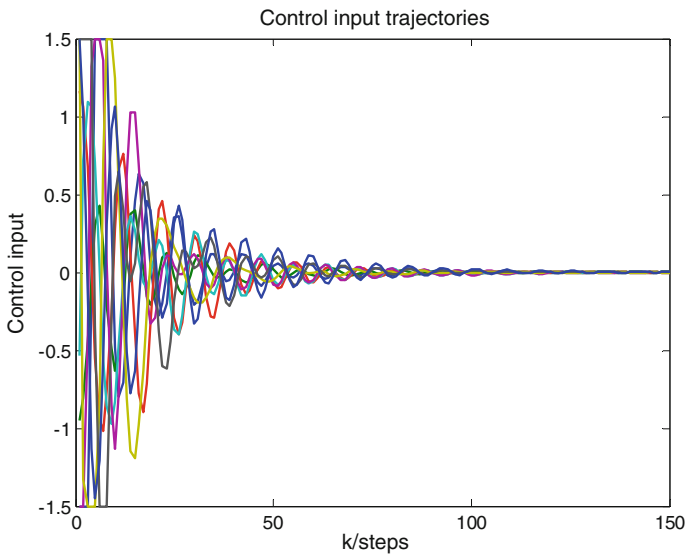


Fig. 5 The control input trajectories of D-NMAS (1)

5 Conclusions

The consensus problem of linear D-NMAS with actuator saturation and undirected graph is investigated in this paper. A new family of distributed parametric low-and-high gain control protocols is provided, and such control protocols rely on the discrete-time Parametric Algebraic Riccati Equation. Meanwhile, the convergence results are also given as consensus function sequences. Finally, an illustrative example is given to demonstrate the correctness and effectiveness of the theoretical results. Further research will be conducted to the consensus problem of D-NMAS with directed graph and subject to actuator saturation.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grants 61374054 and by Province Natural Science Foundation Research Projection of Shaanxi under Grants 2013JQ8038.

References

1. Sinha, A., Ghose, D.: Generation of linear cyclic pursuit with application to rendezvous of multiple autonomous agents. *IEEE Trans. Autom. Control* **51**(11), 1819–1824 (2006)
2. Cao, M., Yu, C., Anderson, B.D.: Formation control using range only measurements. *Automatica* **47**(4), 776–781 (2011)
3. Olfati-Saber, R.: Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Trans. Autom. Control* **51**(3), 401–420 (2006)
4. Zhang, H.T., Zhai, C., Chen, Z.: A general alignment repulsion algorithm for flocking of multi-agent systems. *IEEE Trans. Autom. Control* **56**(2), 430–435 (2011)
5. Olfati-Saber, R.: Distributed Kalman filter with embedded consensus filters. In: Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, 8179–8184. Seville, Spain (2005)
6. Açıkmeşe, B., Mandić, M., Speyer, J.L.: Decentralized observers with consensus filters for distributed discrete-time linear systems. *Automatica* **50**(4), 1037–1052 (2014)
7. Barahona, M., Pecora, L.M.: Synchronization in small-world systems. *Phys. Rev. Lett.* **89**(5), 054101 (2002)
8. Hengster-Movric, K., You, K., Lewis, F.L., et al.: Synchronization of discrete-time multi-agent systems on graphs using Riccati design. *Automatic* **49**(2), 414–423 (2013)
9. Wang, Z., Xu, J., Zhang, H.: Consensusability of multi-agent systems with time-varying communication delay. *Syst. Control Lett.* **65**, 37–42 (2014)
10. Saber, R.O., Murray, R.M.: Consensus protocols for networks of dynamic agents. *Am. Control Conf.* 951–956 (2003)
11. Olfati-Saber, R., Murray, R.M.: Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* **49**(9), 1520–1533 (2004)
12. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* **95**(1), 215–233 (2007)
13. Ren, W., Beard, R.W.: Consensus seeking in multi agent systems under dynamically changing interaction topologies. *IEEE Trans. Autom. Control* **50**(5), 655–661 (2005)
14. Xiao, F., Wang, L.: Consensus problems for high-dimensional multi-agent systems. *IET Control Theory Appl.* **1**(3), 830–837 (2007)
15. Xi, J., Cai, N., Zhong, Y.: Consensus problems for high-order linear time-invariant swarm systems. *Phys. A* **389**(24), 5619–5627 (2010)

16. Xi, J., Shi, Z., Zhong, Y.: Consensus and consensualization of high-order swarm systems with time delays and external disturbances. *J. Dyn. Syst. Meas. Control* **134**(4), 041011 (2012)
17. Xi, J., Shi, Z., Zhong, Y.: Consensus analysis and design for high-order linear swarm systems with time-varying delays. *Physica A* **390**(23), 4114–4123 (2011)
18. You, K., Xie, L.: Network topology and communication data rate for consensusability of discrete-time multi-agent systems. *IEEE Trans. Autom. Control* **56**(10), 2262–2275 (2011)
19. Gu, G., Marinovici, L., Lewis, F.L.: Consensusability of discrete-time dynamic multi agent Systems. *IEEE Trans. Autom. Control* **57**(8), 2085–2089 (2012)
20. Su, Y., Huang, J.: Two consensus problems for discrete-time multi-agent systems with switching network topology. *Automatica* **48**(9), 1988–1997 (2012)
21. Zhao, H., Park, J., Zhang, Y., et al.: Distributed output feedback consensus of discrete-time multi-agent systems. *Neurocomputing* **138**(9), 86–91 (2014)
22. Wei, A., Hu, X., Wang, Y.: Consensus of linear multi-agent systems subject to actuator saturation. *Int. J. Control Autom. Syst.* **11**(4), 649–656 (2013)
23. Meng, Z., Zhao, Z., Lin, Z.: On global leader-following consensus of identical linear dynamic systems subject to actuator saturation. *Syst. Control Lett.* **62**, 132–142 (2013)
24. Yang, T., Meng, Z., Dimarogonas, D.V., Johansson, K.H.: Global consensus for discrete-time multi-agent systems with input saturation constraints. *Automatica* **50**, 499–506 (2014)
25. Fax, J.A., Murray, R.M.: Information flow and cooperative control of vehicle formations. *IEEE Trans. Autom. Control* **49**(9), 1465–1476 (2004)
26. Horn, R.A., Johnson, C.R.: *Matrix Analysis*. Cambridge University Press (1999)
27. Wang, X., Saberi, A., Stoorvogel, A.A., Sannuti, P.: Simultaneous global external and internal stabilization of linear time-invariant discrete-time systems subject to actuator saturation. *Automatica* **48**, 699–711 (2012)

Partitioning Strategies for Multi-robot Area Coverage with No Communication

Cristiano Nattero and Fulvio Mastrogiovanni

Abstract In this paper, we model topology-based partitioning strategies for multi-robot area coverage with no communication as a balanced graph partitioning problem, which is subject to connectivity and reliability requirements. We formalise it as a combinatorial optimisation problem, propose a generalised formulation that encompasses its most relevant features, and we introduce two variants of the basic problem. Furthermore, we discuss a metaheuristic solution method based on a hybrid Ant Colony Optimisation approach. Finally, the performance of the solution is validated using simulations.

Keywords Graph partitioning · Multi-robot coverage · Ant colony optimisation

1 Introduction

During the past few years, much attention has been devoted to the problem of finding *good* region partitions in multi-robot area coverage scenarios. This problem is relevant in a number of applications in which robots must cooperate to fulfil a common objective, for example coordinated remote surveillance tasks, such as wildfire monitoring [1, 2], aerial mapping with UAVs [3] or multi-agent graph patrolling [4–8].

In this paper, we focus on topology-based strategies to partition a region in a number of smaller areas, to be covered by a number of robots with no communication capabilities. In particular, we assume: (i) the region to partition to be conveniently modelled by a topological, graph-like, map of the environment, (ii) robots must visit all the areas in the partition assigned to them and (iii) they move back and forth within the partition to interact with a base station.

C. Nattero · F. Mastrogiovanni (✉)
University of Genoa, Genoa, Italy
e-mail: fulvio.mastrogiovanni@unige.it

C. Nattero
e-mail: cristiano.nattero@unige.it

This scenario entails the existence of a specific graph partitioning problem [9], which is subject to joint balancing, connectivity and reliability requirements as far as partitions are concerned. We argue that such a problem is new in the literature at the best of our knowledge. The contribution of the article is two-fold: (i) a formulation of the *balanced graph partitioning problem with joint connectivity and reliability requirements*; (ii) a meta-heuristic approach based on a hybrid *Ant Colony Optimization* (ACO) algorithm.

In particular, we propose an off-line Algorithm that, given a region to cover using a team of robots, produce a (either with a predefined or variable) number of partitions, each one assigned to a robot. The obtained partitioning takes into account robot activity balancing, the constraint of homing to the base station and the robustness associated with an increased spread of robot paths.

The article is organised as follows. Section 2 provides a formal description of the problem and two relevant variants. Section 3 describes the metaheuristic approach and its extensions. Results in simulation are discussed in Sect. 4. Conclusions follow.

2 Requirements and Problem Definition

2.1 Requirements

Multi-robot area coverage with no communication is characterised by a number of requirements, which may be summarised as follows.

- R1 A given region (usually corresponding to an area to monitor) must be divided in a number of possibly distinct, non-overlapping, partitions taking into account *equity* or *balance* performance indicators. For instance, the number of areas robots are assigned with should be comparable.
- R2 Each partition must contain a specific or relevant location, which is important for a proper partitioning. Furthermore, every other area belonging to the partition should be connected to such a relevant location. This is the case, for instance, when robots have a base station to return to or when back and forth patrolling strategies are used.
- R3 The region to be partitioned may be globally subject to external events or interferences, whose effects should be locally managed by the available partitions in a reliable way. A first example is the presence of intruders in a specific area of the region, which may be detected by robots located in different partitions. A second example is the surveillance of huge areas to detect wildfires.

If the region to partition can be modelled using a topology-based map, i.e., a *graph*, the problem becomes finding a *good* graph partitioning strategy, where each partition serves as a competence area for each robot. In this case, R1 should be enforced to induce a well-balanced visit frequency in all the areas. Since partitions constitute topological trajectories to be followed by robots, all the nodes inside a

partition must be connected, which leads to R2. Furthermore, if we assume absence of communication, connectivity is important since returning to the base station may be the only way to report about a given remote area. Finally, the task to be carried out (e.g., detecting the occurrence of an event in the region) leads to R3. In order to increase the overall system’s reliability (e.g., the likelihood of detecting an intruder or a wildfire), it is necessary that a given area can be monitored by different robots. This leads to partitions which are guaranteed (off-line) as much woven and spread as possible.

2.2 Problem Statement

On the basis of the above considerations, it is possible to formalise the topology-based, multi-robot area coverage with no communication problem as a *balanced graph partitioning problem with joint connectivity and reliability requirements*. Let $G = (V, E)$ be an undirected, unweighted and planar graph, where $V = \{1, \dots, n\}$ is a set of n nodes and $E = \{1, \dots, m\}$ is a set of m edges. Let d^{uv} be a measure of the distance (e.g., Euclidean) from node u to node v . The problem is to partition the set of nodes V in disjoint sets V_1, \dots, V_q such that the corresponding subgraphs G_1, \dots, G_q induced on G are connected. The following requirements are in order.

- *Balancing*. Each subgraph G_i tends to the same order, which is:

$$\Lambda = \left\lceil \frac{n}{q} \right\rceil. \tag{1}$$

This means that subgraphs tend to be made by the same number of nodes, with few exceptions when the rest of the integer division between n and q is not null. In our scenario, balancing is an important requirement to fairly assign robots with areas to visit.

- *Cardinality*. The order of each subgraph is bounded by $C \geq \Lambda$. This and the previous requirement are used to guarantee a certain level of balancing as far as the number of nodes visited by different robots is concerned.
- *Reliability*. In the multi-robot area coverage scenario described above, reliability can be enforced either by spreading the nodes assigned to the same subgraph, i.e., by maximizing the sum of the distances between any pair of nodes assigned to the same partition, or by minimizing the overall degree $\text{deg}(G_i)$ of each subgraph. In this way, contiguous areas will be visited by different robots (possibly at different times). This means that, for each node $v \in V_j$, the degree $\text{deg}(v)$ should tend to 1 or 2. In this paper, we consider both approaches. However, we argue that the first is stronger in most problems involving planning over a surface. As a matter of fact, such problems can be described as planar Euclidean graphs where the first objective usually implies the second, whereas the opposite does not necessarily hold.

- *Connectivity*. Each subgraph G_i must be connected, i.e., for any pair of nodes u, v in V_i , there must be a path from v to u entirely contained in G_i . Note that, together with the previous requirement, this is necessary to enforce connectivity, otherwise better objectives could be obtained by splitting most partitions into separate components. In this way, we can enforce a coverage strategy where robots move back and forth along the assigned region to cover.
- *Separability*. No edges linking nodes from any couple of distinct partitions G_i and G_j must exist, i.e., the number of edges $e = \{u, v\}$, with $u \in G_i$ and $v \in G_j$ must be 0. This requirements allows us to avoid any superimposition of areas to cover.

In the literature, many approaches exist to deal with graph partitioning problems [9, 10]. In this paper we wish to focus on local partitioning methods only, i.e., constructive methods that proceed by growing partitions from assigned or chosen seed nodes, which correspond to initial robot locations. Among local approaches, stochastic search methods are commonly adopted, such as simulated annealing [11–13], heuristic search [14] or graph structural properties [15].

It can be argued that the reliability requirement introduced above is not addressed in the literature. However, this is not surprising. As a matter of fact, this notion of reliability emerges as the *interplay between local properties of graph partitioning and global properties related to subgraph morphologies*, which are seldom considered in the graph partitioning field.

2.3 Mathematical Programming Formulation

In order to avoid any ambiguity, it is useful to introduce a problem formulation based on mathematical programming. We exploit the observation that the connectivity implies the existence of a spanning tree in each subgraph, and we define it using an *arborescence*, i.e., a tree over a directed graph. Let us define:

$$A^0 = \{(u, v), (u, v) \mid \{u, v\} \in E\} \quad (2)$$

as the set of oriented arcs on G , and let us define:

$$V_0 = V \cup 0, \quad (3)$$

where 0 is a dummy node and $S \subseteq V$ is the subset of nodes that can be used as a root for the arborescence, i.e., as a seed for a partition. Although we prefer not to rely on this slightly simplifying assumption, we allow for it to be modelled since, in some cases, this information may be provided. Furthermore, let us define:

$$A = A^0 \cup \{(0, v) \mid v \in S\}. \quad (4)$$

It is now possible to switch from the problem of a constrained oriented spanning forest to the equivalent problem of a constrained spanning arborescence. Both problems are NP-hard [16]. Let us define:

$$B_v = \{u \mid \{u, v\} \in E\} \subseteq V \quad (5)$$

as the set of nodes adjacent to $v \in V$. Let also A_v^+ and A_v^- be the set of arcs entering and leaving node v , respectively. Let us refer to:

$$P = \{u, v \mid u, v \in V \wedge u < v\} \supseteq E \quad (6)$$

as the set of unique unordered pairs of nodes in V . Finally, let us define:

$$K = \{1, \dots, q\} \quad (7)$$

as the the of partitions.

We begin the analysis by considering the case where the number of partitions q and a desired order Λ are given. The following posits are in order.

- An arc decision variable $x_{uv}^k \in \mathbb{B}$ (alternatively x_a^k , with $a = (u, v)$), defined on arcs $(u, v) \in A$, is 1 if node v is assigned to partition k .
- $z_v \in \mathbb{B}$ is 1 if $v \in V$ is not assigned to any partition.
- $y_{uv} \in \mathbb{B}$ is 1 if both nodes in pair $\{u, v\} \in P$ are assigned to the same partition.
- The order of partition $k \in K$ is $f^k \in \mathbb{R}_+$, and its deviation from Λ is measured by $\delta^k \in \mathbb{R}_+$.
- Variables $r_v \in \mathbb{R}_+$, defined for each $v \in V$, are used to impose connectivity.

The vector of objectives $\mathbf{O} = [O_1 \parallel O_4]^t$ can then defined as follows:

$$O_1 = \sum_{v \in V} z_v, \quad (8)$$

$$O_2 = \sum_{k \in K} \delta^k, \quad (9)$$

$$O_3 = \sum_{p \in P} (d^{\max} - y_p d^p), \quad (10)$$

$$O_4 = m - \sum_{e \in E} (1 - y_e). \quad (11)$$

Objective O_1 (8) minimizes the number of nodes that are left unassigned to any partition. It is noteworthy that this objective is necessary to handle cases in which, with a given q , a complete solution cannot be found. Such situations may arise in case of non connected graphs and of connected graphs with particular structures. Objective O_2 (9) favours balanced partitioning schemes, whereas the third (10) and

fourth (11) objectives measure reliability as discussed above. One way to tackle the multi-objective problem is by scalarizing it with a conical combination of (8), (9), (10) and (11), as follows:

$$O = \sum_{i=1}^4 w_i v_i O_i. \tag{12}$$

In (12), v_i are normalizing coefficients which allow for comparisons and w_i are weights that can be tuned to represent the priority of a decision maker, such that $v_i, w_i \geq 0, i = 1, \dots, 4$. The optimization is subject to the following constraints:

$$\sum_{k \in K} \sum_{a \in A_v^+} x_a^k + z_v = 1, \quad \forall v \in V \tag{13}$$

$$x_{vu}^k \leq \sum_{a \in A_v^+} x_a^k, \quad \forall v \in V, k \in K \tag{14}$$

$$\sum_{a \in A_0^-} x_a^k \leq 1, \quad \forall k \in K \tag{15}$$

$$f^k = \sum_{v \in V} \sum_{a \in A_v^+} x_a^k \leq C, \quad \forall k \in K \tag{16}$$

$$r_u - r_v + C \sum_{k \in K} x_a^k \leq C - 1, \quad \forall a \in A \tag{17}$$

$$\delta^k \geq |f^k - \Lambda|, \quad \forall k \in K \tag{18}$$

$$y_{uv} \geq \sum_{a_1 \in A_u} x_{a_1}^k \sum_{a_2 \in A_v} x_{a_2}^k - 1. \quad \forall \{u, v\} \in P, k \in K \tag{19}$$

Since each node is constrained to be the head of one of its arcs, it is at most assigned to a single arborescence (13). An arc with origin in node v can be assigned to partition k only if at least another arc assigned to k enters in v (14). Constraints (15) impose that, for each arborescence, there is one entry arc at most. Cardinality constraints (16) limit the order of the partition to C . Sub-tour elimination constraints (17), introduced by [17], ensure that each partition is associated with an acyclic connected and directed sub-graph, i.e., an arborescence in our case. Constraints (18) set the value of the deviational variables. Constraints (19) sets y_{uv} to 1 if both nodes u and v are in the same partition. This formulation actually suffers from a series of drawbacks, including non linearity (18), non tight linear relaxation, which is due to *bigM* in (16) and (17), symmetry (due to equivalent arborescences on the same partition and equivalent permutation). These are all important issues which can be mitigated (and in fact all these topics are currently under investigation). However, we stick to this model for the sake of simplicity here.

Let us now consider the case where the number of partitions q and the balance level (i.e., the desired order Λ) are not known. In this case, it makes sense to minimize them replacing objective O_1 in (8) with the weighted sum of the two following objectives:

$$O_1^K = \sum_{k \in K} \sum_{v \in V} x_{0v}^k, \quad (20)$$

$$O_1^A = \Lambda. \quad (21)$$

To avoid solutions with empty partitions, it is also necessary to enforce the fact that each node is assigned to a partition, i.e., converting O_1 into a constraint as follows:

$$\sum_{v \in V} z_v = 0. \quad (22)$$

One possible approach to define the set K is by initializing $q = n$, although tighter bounds can be inferred (e.g., with a heuristic).

Finally, we wish to highlight that an excessive spreading of the partitions might be undesirable. As an example, area coverage paths may be spread to maximize the likelihood of detecting an interesting event in a given region by any robot covering that region. However, an excessive spreading may cause an excessive path length, which may undermine the overall quality coverage.

Similarly to the case of O_2 (9), we propose to tackle this issue with a *soft threshold* method, i.e., by replacing objective O_3 (10) with a penalization of the deviation from a desired sum of distances Γ :

$$O_3^\Gamma = \left| \sum_{p \in P} y_p d^p - \Gamma \right|. \quad (23)$$

In the next Section we describe an approach to tackle this problem and its variants.

3 A Hybrid Ant Colony Optimization Approach

3.1 A Solution Based on Ant Colony Optimization

In order to provide a solution to the problem as described in the previous Section, we implemented an Ant-Colony Optimization algorithm [18], outlined in Algorithm 1.

Basically, the algorithm is a repeated constructive procedure, which sequentially grows arborescences a node at a time, until the desired level is reached. The solution F built by an ant (line 6) is compared against the incumbent and saved if necessary (lines 7–10). Then, a local pheromone update phase (line 11) discourages the replication of the same solution by the following ant, in order to avoid potential *basins of attraction*. Finally, when every ant constructs a solution, the iteration terminates by undoing the local modifications, reinforcing the pheromone trails associated with good solution elements and penalizing those elements corresponding to solutions that have not been reproduced (line 13). The pheromone is independent from the

Algorithm 1 Generation of partitions

Require: $G = (V, E), K, A$

- 1: $F^* \leftarrow \emptyset$
- 2: $O_{\text{best}} \leftarrow +\infty$
- 3: $\pi \leftarrow \text{initPheromone}()$
- 4: **while** termination condition **not** met **do**
- 5: **for all** ant a in ants **do**
- 6: $F \leftarrow \text{buildSolution}(a)$
- 7: **if** $O(F) < O_{\text{best}}$ **then**
- 8: $F^* \leftarrow F$
- 9: $O_{\text{best}} \leftarrow O(F^*)$
- 10: **end if**
- 11: localPheromoneUpdate(F)
- 12: **end for**
- 13: globalPheromoneUpdate(F^*)
- 14: **end while**
- 15: **return** F^*

objective values and is updated in an asymptotic way which, in practice, removes one of the parameters to be tuned. Major details can be found in [16, 19]. However, our implementation is characterized by a number of effective modifications. Among them, the selection of the next node to add to current partition V_i is of particular note. The selection is informed not only by the pheromone trail π_v , but also by the so-called *ant product*

$$\Pi_v = \pi_v (g_v h_v)^\beta, \quad (24)$$

where β is a positive number that is decreased over iterations as $\beta \leftarrow \beta\psi$, $\psi \in [0, 1]$, thus limiting the influence of the heuristic in favour of the experience, whereas g_v is a greedy value representing the contribution of node v to the sum of distances of objective O_3 , i.e.,

$$g_v = \sum_{u \in B_v \cap V_i} d^{uv}. \quad (25)$$

The heuristic value h_v is computed as:

$$h_v = \frac{|B_v| + |H_v| + \epsilon}{2|B_v| + \epsilon}, \quad (26)$$

where $H_v \subseteq B_v$ is the set of nodes adjacent to v that have been assigned to other partitions and ϵ is a positive number preventing divide-by-zero issues. As far as ϵ is concerned, it is convenient to set it to a small number, but not too small to create numerical issues (after some tuning we posit $\epsilon = 0.5$). The rationale behind this choice is that solutions with a better O_4 are locally enforced.

At each insertion, the ant randomly chooses between *exploitation* and *exploration* with probability p and $1 - p$, respectively.

3.2 Candidate Strategy

In the ACO-based approach outlined in the previous Section, it is necessary to devise a candidate strategy to select the next node to add to the partition. In our case, the node to insert is selected from a subset θ of the nodes adjacent to V_i , which is generated in two steps, namely S1 and S2.

- S1 Only nodes with the minimum number of adjacent unassigned nodes are kept in θ . Let us define θ^u as the set of unassigned nodes adjacent to θ .
- S2 The procedure goes one level deeper and looks at the unassigned nodes which are adjacent to nodes in θ^u . Then, it computes the sum of their adjacent unassigned nodes. θ is further reduced by keeping only nodes adjacent to those for which this sum is minimal.

Considered together, S1 and S2 implement the so-called *Least Cumulative Unassigned* (LCU) rule, whose rationale is to select first those nodes whose likelihood to be selected is lower, since they are characterized by a lower number of ways to be reached. In fact, thanks to its *look ahead* effect, LCU allows for obtaining complete solutions very quickly in most cases.

3.3 Detecting the Number of Partitions

We extended Algorithm 1 to detect the number of partitions to identify. This can be done by exploiting the cardinality constraint as an iterative process involving two steps at each iteration. In each iteration, an initial guess about the number of partitions can be obtained as

$$q \leftarrow \left\lceil \frac{n}{C} \right\rceil. \tag{27}$$

Then, Algorithm 1 is executed for a given amount of time. If no complete solution is found, then q is increased, and the process is iteratively repeated. In a first tentative implementation q was incremented by one unit only. However, the process proved very slow to converge. We obtained a sensibly faster iterative Algorithm by incrementing q by the number u of nodes left unassigned, divided by C , as

$$q \leftarrow q + \left\lceil \frac{u}{C} \right\rceil. \tag{28}$$

However, this assignment might lead to overestimates of the number of partitions. A more precise approach involves both adding a smaller increment and, once a complete solution is found, to reduce q . Surprisingly enough, in our experiments the solver manages to obtain a solution decreasing q by 1 unit only on a few instances, and only in one case we found a better solution decreasing it by 2 units.

3.4 Adaptive Soft Thresholding

In order to find a good value for the penalization of the deviation from the desired sum of distances Γ , we adopted a two-step approach to link it to the current problem. In the first step, we optimize the objective function as described in the previous Section, and compute a reference value d for the sum of distances in objective O_3 (10). In the second step, we impose $\Gamma = \rho d$, where ρ is a desired ratio, and then we reiterate. However, it is necessary to introduce a reset in the pheromone trails between the two steps, otherwise the learning mechanism would take too long to get rid of its bias. We do not reset β to the initial value as the heuristic tends to produce solutions that are very spread and we want to contain this phenomenon.

4 Evaluation of Solutions

The solver has been implemented in Java. Tests on a 2.4GHz Intel i7-3630QM with 8GB of RAM machine running 64 bit Linux have been carried out. The following set of parameters has been adopted: $\epsilon = 0.5$, $p = 0.9$, $\beta = 1$, $\psi = 0.6$, 20 ants.

A set of instances based on regular triangular grids (f) has been randomly generated. A ratio σ of edges (c) or nodes (p) has been removed from regular grids. Values used for σ are 0.1, 0.3 and 0.6. Without loss of generality, this generation procedure also removed *singletons*. Instance names are coded as {f|c|p}_n_m_q. In f instances, q has been obtained using (27), whereas in c and p instances it has been determined with the procedure explained in Sect. 3.3.

Table 1 has a row for each instance which shows instance ID and the *relative percent deviation* (RPD) from the best found result averaged on 10 runs, respectively, for 10, 30, 60 and 300 seconds.

It can be noticed from the Table that there are some instances apparently harder than other as the ACO-based approach takes a longer time to converge. Furthermore, the difficulty the Algorithm encounters does not seem to be correlated to the adopted generation method, as there are high RPDs at 10 and 30 seconds for all the three classes of instances.

To obtain a better insight about the results, we summarize results in Table 2, which shows the standard deviation (stdev), the 95 % confidence intervals (lo and up), the average (avg) and the geometric mean (geomean) of the RPD. It can be noticed that it takes a while for the results to stabilise, as emphasized by the discrepancy between the average and the geometric mean. After 1 minute the average RPD drops below 4 % with respect to the initial one. This ratio further reduces to 2.75 % after 5 minutes, with a significative shrinking of the confidence interval.

Table 1 Detailed computational results

ID	10 s (%)	30 s (%)	60 s (%)	300 s (%)
c_2208_3023_140	10.18	8.85	8.31	5.88
c_2217_2088_295	10.54	9.98	7.83	5.64
c_2279_2693_177	10.27	10.27	7.83	7.83
c_2304_3154_147	10.25	9.89	8.97	8.97
c_2329_2175_326	10.16	10.16	9.89	9.89
c_2373_2808_181	10.61	9.08	8.44	8.44
c_2398_3288_154	10.04	9.42	9.37	7.33
c_2400_3288_153	9.67	9.67	8.27	8.27
c_2408_2265_327	9.74	9.74	9.74	6.56
c_2417_2265_328	212.80	212.80	9.54	7.83
c_2475_2925_191	10.36	10.36	10.27	5.65
c_2477_2925_191	215.49	10.21	8.45	6.20
c_2483_2356_339	896.67	896.00	10.28	8.11
c_2499_3425_159	45.28	9.35	9.35	8.46
c_2500_3425_160	261.50	261.50	8.12	5.89
f_2209_3243_139	767.04	766.87	9.36	7.41
f_2304_3384_144	925.31	10.14	8.90	8.15
f_2401_3528_151	40.89	39.94	39.94	5.86
f_2500_3675_157	879.44	10.17	9.95	9.23
p_2246_2980_148	30.61	30.41	7.78	7.78
p_2250_2980_147	952.53	8.67	8.67	6.12
p_2338_1703_650	966.89	966.67	8.91	8.36
p_2338_3094_154	13.05	13.02	9.43	9.43
p_2341_3101_154	23.15	9.23	9.23	7.03
p_2430_3221_161	60.46	59.64	8.08	8.08
p_2430_3224_158	768.71	9.86	7.66	6.73

Table 2 Statistical summary of results

	10 s (%)	30 s (%)	60 s (%)	300 s (%)
stdev	382.24	282.78	6.14	1.25
lo	128.52	22.53	7.74	7.02
avg	275.45	131.23	10.10	7.51
up	422.37	239.92	12.46	7.99
geomean	62.92	24.93	9.40	7.40

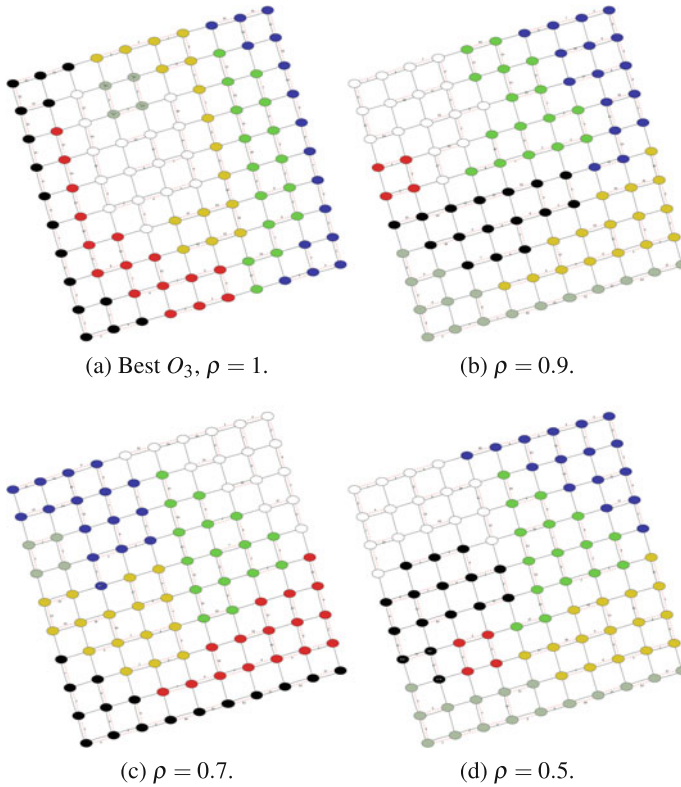


Fig. 1 The effect of soft thresholding

As far as the effect of soft thresholding is concerned, Fig. 1 shows how solutions on a 10×10 square grid change when lowering the threshold from 1 to 0.9, 0.7 and 0.5. Each partition is represented with a different color. In the Figure, red connections on links represent the insertion sequence in each partition. It is possible to observe that, within each partition, all nodes are connected. It can be noticed that, on average, as long as ρ decreases, partitions tend to be consolidated and less spread.

5 Conclusions

In this paper, we model multi-robot area coverage with no communication as a balanced graph partitioning problem with connectivity and reliability requirements. In particular, we focus on partitioning a topology-based representation of the area to cover. To this aim, we introduce a formalisation as well as a meta-heuristic solution method based on a hybrid ant colony optimisation approach. The solution is tested

using simulations. In particular, we observe that the method is flexible, since it is able to deal with different variants of the problem requiring the computation of the number of partitions or a soft thresholding to bound the spread of the arborescence. Current work is two-fold. On the one hand, it is focused on the automatic computation of an appropriate value for the ratio used in the threshold definition. On the other hand, this off-line procedure will be validated in a real-world multi-robot scenario, which will require us to consider specific information, such as the calculation of the yield of a partition, to be integrated within the model.

References

1. Ollero, A., de Dios, J.M., Merino, L.: Unmanned aerial vehicles as tools for forest-fire fighting. In: Proceedings of the 2006 International Conference on Forest Fire Research (CFFR 2006). Coimbra, Portugal, Nov 2006
2. Merino, L., Caballero, F., de Dios, J.M., Maza, I., Ollero, A.: An unmanned aircraft system for automatic forest fire monitoring and measurement. *J. Intell. Robot. Syst.* **65**(1–4), 533–548 (2012)
3. Artieda, J., Sebastian, F.J., Campoy, P., Correa, J., Mondragon, I., Martinez, C., Olivares, M.: Visual 3-D SLAM from UAVs. *J. Intell. Robot. Syst.* **55**(4–5), 299–321 (2009)
4. Kolling, A., Carpin, S.: The GRAPH-CLEAR problem: definition, theoretical properties and its connections to multirobot aided surveillance. In: Proceedings of the 2007 IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS 2007). San Diego, CA, USA, Nov 2007
5. Kolling, A., Carpin, S.: Multi-robot surveillance: an improved algorithm for the GRAPH-CLEAR problem. In: Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA 2008). Pasadena, CA, USA, May 2008
6. Kolling, A., Carpin, S.: Extracting surveillance graphs from robot maps. In: Proceedings of the 2008 IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS 2008). Nice, France, Sept 2008
7. Wurm, K., Stachniss, C., Burgard, W.: Coordinated multi-robot exploration using a segmentation of the environment. In: Proceedings of the 2008 IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS 2008). Nice, France, Sept 2008
8. Iocchi, L., Marchetti, L., Nardi, D.: Multi-robot patrolling with coordinated behaviours. In: Proceedings of the 2011 IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS 2011). San Francisco, CA, USA, Sept 2011
9. Bichot, C., Siarry, P.: Graph Partitioning. Wiley, New York, USA (2011)
10. Spielman, D., Teng, S.: Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In: Proceedings of the 26th Annual Symposium on Theory of Computing (STOC 2004). New York, NY, USA (2004)
11. Johnson, D., Aragon, C., McGeoch, L., Schevon, C.: Optimization by simulated annealing: an experimental evaluation. Part I, graph partitioning. *Oper. Res.* **37**(6), 865–892 (1989)
12. Klein, R., Dubes, R.: Experiments in projection and clustering by simulated annealing. *Pattern Recognit.* **22**(2), 213–220 (1989)
13. Schaeffer, A., Satu, E.: Stochastic local clustering for massive graphs. In: Ho, T., Cheung, D., Liu, H. (eds.) *Advances in Knowledge Discovery and Data Mining*, ser. Lecture Notes in Computer Science, vol. 3518, pp. 354–360. Springer, Berlin (2005)
14. Felner, A.: Finding optimal solutions to the graph partitioning problem with heuristic search. *Ann. Math. Artif. Intell.* **45**(3–4), 292–322 (2005)
15. Clauset, A.: Finding local community structure in networks. *Phys. Rev. E* **72**, 026132 (2005)

16. Anghinolfi, D., Cannata, G., Mastrogiovanni, F., Nattero, C., Paolucci, M.: Heuristic approaches for the optimal wiring in large-scale robotic skin design. *Comput. Oper. Res.* **39**(11), 2715–2724 (2012)
17. Miller, C., Tucker, A., Zemlin, R.: Integer programming formulation of traveling salesman problems. *J. ACM* **7**, 326–329 (1960)
18. Dorigo, M., Blum, C.: Ant colony optimization theory: a survey. *Theoret. Comput. Sci.* **344**, 243–278 (2005)
19. Nattero, C., Anghinolfi, D., Paolucci, M., Cannata, G., Mastrogiovanni, F.: Experimental analysis of different pheromone structures in ant colony optimization for robotic skin design. In: *Proceedings of the 2012 International Federated Conference on Computer Science and Information Systems (FedCSIS 2012)*. Wrocland, Poland, Sept 2012

Max-Sum for Allocation of Changing Cost Tasks

James Parker, Alessandro Farinelli and Maria Gini

Abstract We present a novel decentralized approach to allocate agents to tasks whose costs increase over time. Our model accounts for both the natural growth of the tasks and the effort of the agents at containing such growth. The objective is to minimize the increase in task costs. We show how a distributed coordination algorithm, which is based on max-sum, can be formulated to include costs of tasks that grow over time. Considering growing costs enables our approach to solve a wider range of problems than existing methods. We compare our approach against state-of-the-art methods in both a simple simulation and RoboCup Rescue simulation.

Keywords Multi-robot systems • Task allocation • Binary max-sum

1 Introduction

Task allocation problems are ubiquitous. We address task allocation for problems where the costs for completing the tasks increase over time. Applications for this type of problem include minimizing damage from invasive species, resource distribution for fighting epidemics, and containment of wild fires. In most practical applications, multiple agents need to cooperate to complete all the tasks. (i.e., fire fighters cooperate to extinguish large fires). Since costs grow over time, a task can grow indefinitely if not enough agents are assigned to it. If the growth rate surpasses the reduction the agents can provide, then that task cannot be completed.

J. Parker (✉) · M. Gini (✉)

Computer Science and Engineering, University of Minnesota, Minneapolis, USA
e-mail: jparker@cs.umn.edu; park0580@umn.edu

M. Gini

e-mail: gini@cs.umn.edu

A. Farinelli

Department of Computer Science, University of Verona, Verona, Italy
e-mail: alessandro.farinelli@univr.it

© Springer International Publishing AG 2017

W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_46

A formulation for this problem and a centralized heuristic method was proposed in [8]. We use the same formulation of the cost growth model and propose the first decentralized solution for this class of task allocation problems. Our method is based on max-sum [2], which we modify by projecting future growth into our formulation. We assume that an approximation of the growth functions is known upfront, but they could be learned. New tasks of any size can appear at any time and agents may disappear.

This paper provides the following main contributions: (i) a general approach to handle growth functions for problems where the costs of tasks increase over time; (ii) an empirical evaluation and comparison with state-of-the-art methods both in a simple simulation and in RoboCup Rescue [3]. The results show our decentralized method performs similarly to the state-of-the-art centralized method.

2 Related Work

Multi-agent task allocation with temporal constraints has been studied from a variety of angles. Efficient auction based methods for task with disjoint time windows are presented in [6]. Their method is not directly applicable to our problem because in our case the cost of the tasks changes over time and hence the rewards given to the agents for completing tasks have to depend on the actions of other agents. The work in [16] focuses on tasks that can be completed only with multiple agents but assumes task costs are fixed. In [1] Fisher market clearing with soft deadlines is used to assign agents to tasks when reward for task completion decreases over time. In their work tasks become less important over time, while in our work tasks become harder over time and there can be severe consequences if tasks are not completed (i.e., the city might burn completely).

A decentralized solution using max-sum to allocate tasks in RoboCup Rescue is presented in [14]. The work focuses on coalition formation among agents taking into account both temporal and spatial constraints. Their objective is to complete as many tasks as possible when each task has a deadline for being completed. In their case tasks have a fixed cost, while we assume costs grow over time. Their experimental results are obtained using synthetic data not in RoboCup Rescue. Other distributed approaches based on max-sum have been proposed for RoboCup Rescue, in particular using Tractable Higher Order Potentials (THOPs) [15] to improve the efficiency [11, 13]. The use of THOPs reduces the message passing of max-sum from exponential to polynomial by exploiting the structure of the problem. To gain these benefits, problems must be formulated with binary variables and with more limited types of factors. The closest to our work is the work in [12], which uses binary max-sum. They evaluate the utility of the current state using a domain specific heuristic based on distance, intensity of fire, and number of agents assigned to a task. Our approach is more general because it is based on a task cost growth model which is independent of the specific application domain. The incorporation of temporal reasoning on task

growth, estimation of solution cost and generality across domains makes our work significantly different.

3 Problem Definition

We assume homogeneous agents which have a spatial location and tasks which have a location and have a cost that changes over time. An agent must be at a task’s location in order to work on that task. We denote the set of homogeneous agents by $A = \{a_1, \dots, a_n\}$, where $n = |A|$ and the set of tasks by $B = \{b_1, \dots, b_m\}$, where $m = |B|$. The set of active agent assignments is denoted by $N^t = \{n_1^t, \dots, n_{|B|}^t\}$, where n_i^t is the set of agents from A that are currently working on task b_i at time t . An agent can only work on one task at a time, so $n_i^t \subseteq A$ and $\forall i \neq j, n_i^t \cap n_j^t = \emptyset$. The travel time, $TT(x, y)$, between two locations, x and y , is assumed to be computable. Travel time causes a delay between the time an agent works on one task and the time it can work on another. This means $\sum_i n_i^t \leq |A|$ as some agents might be in transit.

Each agent provides work amount w per time unit once the agent has reached a task. Every task $b_i \in B$ has a cost defined with the following recursive relationship:

$$f_i^{t+1} = f_i^t + h_i(f_i^t) - w \cdot |n_i^t|, \tag{1}$$

where f_i^t starts at some initial cost f_i^0 and h_i is a positive semidefinite monotonically increasing function, as in [8]. The growth function h_i is assumed to be known but it could be estimated. Here we treat f_i^t as a sequence due to the use of discrete time steps, but it could be treated as a continuous function if that is a better model for the domain.

If at time t the cost of task b_i , namely f_i^t , reaches or goes below zero the task is considered complete. We denote this completion time as ct_i . For this reason, when f_i^t is non-positive, $h_i(f_i^t)$ is assumed to be zero and we do not allow those tasks to be assigned to agents. The time when the last task is completed is defined as t_s , namely $\max_{b_i \in B} ct_i$. Our objective is to minimize the accumulated growth of all tasks, denoted by

$$R_{t_s} = \sum_{t < \max_{b_i \in B} ct_i} \sum_{b_i \in B} h(f_i^t) \tag{2}$$

This objective corresponds to minimizing the amount costs increase over all time. For instance, in an epidemic outbreak this would mean allocating a limited number of doctors (agents) to areas with sick people (tasks). The sickness is contagious so areas with the disease have a higher infection rate, based on the growth rate $h_i(x)$. Our model projects the infection rate and the rate of doctors curing to decide where doctors should be located and when and if they need to move. The people who are sick before the crisis is discovered do not affect our goal, but we want to minimize R_{t_s} , which is the number of new people who become sick before the epidemic is cured.

4 Max-Sum Formulation

A Distributed Constraint Optimization problem (DCOP) can be formally defined as a tuple $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$, where $\mathcal{X} = \{x_1, x_2, \dots, x_{|\mathcal{X}|}\}$ is a set variables with domains $\mathcal{D} = \{D_1, D_2, \dots, D_{|\mathcal{X}|}\}$ and $\mathcal{C} = \{c_1, c_2, \dots, c_{|\mathcal{C}|}\}$ are utility functions. Each utility function, c_i , has a value based on a subset of \mathcal{X} variables, $\{x_{i_1}, x_{i_2}, \dots\}$, specifically $c_i : D_{i_1} \times D_{i_2} \times \dots \rightarrow \mathfrak{R}$. We will denote the specific variables used for each utility function, c_i , by X_{c_i} . The max-sum algorithm finds an approximate solution to the sum of each utility function c_i , namely:

$$\text{maximize } \sum_{c_i \in \mathcal{C}} c_i(X_{c_i})$$

Max-sum finds a solution by alternating between passing messages from variables to functions and from functions to variables. Both types of messages are repeatedly sent until they cease to change or a maximum number of iterations has been reached. After convergence, each variable can determine the best value by examining the most recent messages passed to it. Both message types generate outgoing messages based on all the other incoming messages.

A message from a variable to function simply adds all the messages coming into the variable from all the other functions. A message from a variable x_i to function c_j would be generated only if $x_i \in X_{c_j}$, namely this function uses the variable. The message generated is:

$$\mu_{x_i \rightarrow c_j}(d) = \sum_{k \in C_{x_i}/c_j} \mu_{k \rightarrow x_i}(d),$$

where C_{x_i} is the set of all functions which involve x_i and $d \in D_i$ is a value in the domain of x_i . $\mu_{k \rightarrow x_i}$ is the most recent message x_i received from function k .

The message from a function to variable is then:

$$\mu_{c_j \rightarrow x_i}(d) = \max_Y \left(c_j(d, Y) + \sum_{k \in X_{c_j}/x_i} \mu_{k \rightarrow c_j}(y_i) \right),$$

where $y_i \in D_i$ is a value of variable x_i , and Y is a set of values for variables X_{c_j}/x_i . Once messages have converged, each variable x_i chooses the value that maximizes all of its incoming messages, namely $\text{argmax}_d \sum_{k \in C_{x_i}} \mu_{k \rightarrow x_i}(d)$.

Approaches to solve DCOPs range from optimal techniques [7] to heuristics [2]. Optimal solutions require an exponential coordination overhead (i.e. communication computation). We approximate an unknown process which involves a potentially large number of agents, hence we use heuristic methods, which do not guarantee a solution quality but have a lower coordination overhead. Optimal methods would only solve the approximate model without guaranteeing an actual optimal solution. Following [12] we use binary max-sum in conjunction with Tractable Higher Order

Potentials to more efficiently pass messages [15]. This can reduce the message passing complexity from $O(|B|^{|A|})$ to $O(|A| \log |A|)$. The use of binary variables means $D_i \in \{0, 1\}$.

The heuristics to prioritize tasks used in [12] are domain specific to RoboCup Rescue. Explicit values are assigned to fires based on their intensity and the number of agents assigned. A distance penalty discourages agents from selecting tasks far away. This is balanced with a utility heuristic, which discourages many agents from working on the same task. Our max-sum model instead assumes task cost size follows the recursive relationship in Eq. 1 and computes the expected effect of agent allocation on the tasks. We also compute a utility and distance penalty, but our distance penalty computes how much a task will grow before an agent reaches it, and our utility computes how much faster a task will be completed if an agent is assigned to it. This allows our max-sum formulation to generalize to any task allocation problem, as long as their cost can be approximated by the general recursive relationship in Eq. 1, unlike the work in [12]. Our method also solves a temporal component which is not addressed in [12].

Next we discuss the specifics of our max-sum problem formulation. We treat the cost, f_i^t , as an algebraic variable that depends on time t and the number of agents active $|n_i^t|$. The goal is to minimize R_{t_i} , the total amount of growth before all tasks are completed. Our task growth approximation for the model in Eq. 1 is $h(f_i^t) = g_i \cdot f_i^t$, where g_i is a constant found empirically. For ease of calculation, we assume that time is continuous rather than discrete so this means $\frac{\delta f_i^t}{\delta t} = g_i \cdot f_i^t$. Solving for f_i^t yields the well known exponential function: $f_i^t = K_i \cdot e^{g_i \cdot t}$, where K_i is the integration constant. The initial task cost, f_i^0 , is used to determine an appropriate K_i for each task. The cost reduction from agents working on a task is represented as $\frac{\delta f_i^t}{\delta t} = g_i \cdot f_i^t - |n_i^t| \cdot w$, which can again be solved for f_i^t :

$$f_i^t = \frac{|n_i^t| \cdot w}{g_i} + K_i \cdot e^{g_i \cdot t}, \tag{3}$$

where again K_i is an integration constant. K_i is negative if the agents are completing the task faster than it is growing, namely $K_i < 0$ if and only if $|n_i^t| \cdot w > g_i \cdot f_i^t$. If the number of agents changes, K_i has to be recomputed. We can use further algebra to get the following equations:

$$f_i^t = \frac{|n_i^t| \cdot w}{g_i} + \left(f_i^0 - \frac{|n_i^t| \cdot w}{g_i} \right) \cdot e^{g_i \cdot t} \tag{4}$$

$$-L(|n_i^t|) = \frac{|n_i^t| \cdot w}{g_i} \cdot (\ln(\alpha) + 1 - \alpha) + f_i^t \cdot (\alpha - 1), \tag{5}$$

where $\alpha = \frac{|n_i^t| \cdot w}{|n_i^t| \cdot w - f_i^t \cdot g_i}$

Equation 4 estimates the cost of task b_i at time t when $|n_i^t|$ agents are assigned to it. Equation 5 is the natural growth of task cost from time t until task i is finished. We

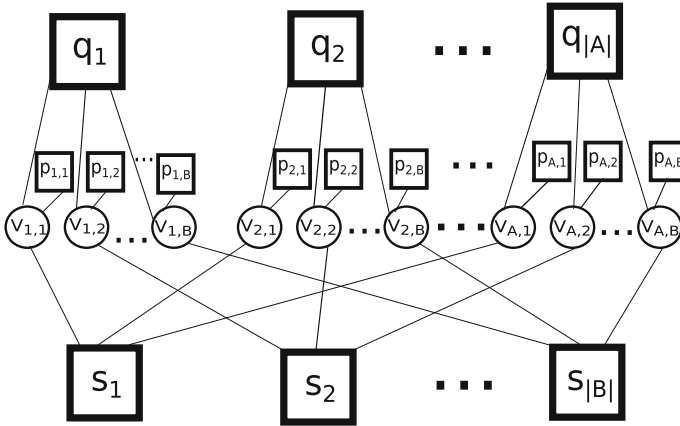


Fig. 1 The factor graph for our representation. v are binary indicators of an agent's assignment, p distance penalties, q constraints for ensuring an agent is active on one task, and s the utility function for the assignment (Eq. 5)

use a negative sign since max-sum maximizes but we want to minimize. This means $\sum_i -L(|n_i^t|) = -(R_t - R_t)$, where R_t is from Eq. 2.

The factor graph [5] used to model this problem is shown in Fig. 1, with circles for the variables and squares for the functions. Each agent a_i controls all $|B|$ binary task assignment indicator variables $v_{i,j}$, where j indicates the agent's assigned task, b_j . The constraint q_i ensures an agent is assigned to exactly one task, specifically:

$$q_i(v_{i,1}, v_{i,2}, \dots, v_{i,|B|}) = \begin{cases} 0 & \sum_j v_{i,j} = 1 \\ -\infty & \text{otherwise} \end{cases}. \quad (6)$$

s_j is the accumulated cost growth until the task is finished. Since we want to minimize the accumulated growth, the negative of this value is taken:

$$s_j(v_{1,j}, v_{2,j}, \dots, v_{|A|,j}) = -L\left(\sum_i v_{i,j}\right), \text{ with } L \text{ as in Eq. 5.} \quad (7)$$

$p_{i,j}(v_{i,j})$ is a travel penalty, since s_j assumes the agent arrives instantly, the correct amount is subtracted if agent a_i is still traveling to task b_j . $TT(v_{i,j}, b_j)$ denotes the travel time between agent a_i and task b_j . Knowledge of \hat{T} , the time it takes the currently assigned agents to complete the task, is approximated by the previous assignment to compute $p_{i,j}(v_{i,j})$ as follows:

$$p_{i,j}(v_{i,j}) = \begin{cases} \frac{w \cdot e^{g_i \hat{T}}}{g_i} \cdot (e^{-g_i \cdot TT(v_{i,j}, b_j)} - 1) & v_{i,j} = 1 \\ 0 & v_{i,j} = 0 \end{cases} \quad (8)$$

This factor graph is binary and composed of only Tractable Higher Order Potentials, specifically only the cardinality of the inputs is required for s_j (number of agents assigned to the task) and q_i (number of tasks assigned to an agent) [15].

Our global utility function is a sum over all functions:

$$\sum_i q_i(v_{i,1}, v_{i,2}, \dots) + \sum_i \sum_j p_{i,j}(v_{i,j}) + \sum_j s_j(v_{1,j}, v_{2,j}, \dots)$$

This factor graph is cyclical, so max-sum is not guaranteed to find a global optimum. Our max-sum use the same framework as [12], based on the work of [17] for an efficient anytime solution. Max-sum is run at every time step for multiple reasons. New tasks may appear, new agents may join and current agents may leave. Also as both the modeling and solution are approximate, readjusting projections with the real outcomes improves accuracy. This solutions is also decentralized, compared with past work [8], which proposed a centralized solution.

5 Results

We compare the performance of max-sum against a modified version of RT-LFF [8]. The original RT-LFF tried to minimize the time the latest task finished. We changed RT-LFF to instead minimize the accumulated task cost so it is consistent with our current problem formulation. RT-LFF uses LFF [8] for this initial assignment, which is optimal when agents cannot be reassigned. RT-LFF can change assignments if a large gain is detected. This modified RT-LFF has a worst-case running time of $O(|A||B|^2)$, but is typically significantly less in practice compared to $O(|B||A| \log |A|)$ of binary max-sum.

The comparisons are done in both a controlled simple simulator and in the RMAS-Bench [4] extension of RoboCup Rescue to show the applicability to complex environments and to make our results easily comparable.

5.1 Simple Simulator

The simple simulator gives us greater analytical power than RoboCup Rescue since every aspect can be controlled. Here we use the same h growth function for all the tasks except in the mixed experiments (see Table 1). We explicitly define the initial costs, f_i^0 , growth functions, $h_i(x)$, and the effect of agents on the task, w , for any number of tasks.

We consider the optimal, max-sum, and RT-LFF. For additional comparison, we tested two simple strategies: UNIFORM and ALLONONE. The UNIFORM strategy assigns an equal number of agents to all tasks that are not complete. If a task cost reaches zero, it is marked as complete and the agents are redistributed. The

Table 1 Accumulated growth cost, R_i , for different setups in the simple simulator

$h(x)$	Initial costs	Optimal	Max-sum	RT-LFF	AllOnOne	Uniform
$0.000016 \cdot x^3$	{20, 15}	8.1333	8.1334	8.3724	15.852	12.121
$0.00019 \cdot x^2$	{25, 20, 10}	22.761	22.761	23.632	39.268	49.076
$0.0036 \cdot x$	{50, 30}	188.72	188.72	188.72	188.76	189.41
$0.02 \cdot \sqrt{x}$	{5, 10, 15, 20}	43.398	43.398	144.60	69.117	79.878
$0.02 \cdot \ln(x + 1)$	{40, 30}	28.158	28.158	41.661	31.409	39.134
Mixed	{25, 10, 10}		24.485	29.064	28.110	40.948
Sigmoid	{40, 30}		41.218	100.99	26.127	92.530

ALLONONE strategy assigns all agents to a single task until that task is completed. The order in which tasks are selected is from left to right in the “Initial Costs” column of Table 1. We experimented with different types of growth functions (Table 1). All experiments in the section use 20 agents with work rate $w = 0.015$. The coefficients for w and in the growth functions are small to minimize the discretization of the time steps. A travel time of zero between all tasks is used to have a direct comparison as the optimal solution is only known for this case.

When there is no travel time, three different families of growth functions can be classified based on $\frac{\partial^2}{\partial x^2}h(x)$ as shown in [9]. If $\frac{\partial^2}{\partial x^2}h(x) < 0$, then the growth function is concave and the optimal solution is to have all agents on the task that is the smallest. In the case of a tie for the smallest, one task should be picked randomly and all agents should work on that one. When $\frac{\partial^2}{\partial x^2}h(x) = 0$, the growth is proportionally to the current cost of the task. In this linear case, any assignment that fully utilizes all agents at every time step is optimal. This is easily achieved since agents require no time to move between tasks. The last case is when $\frac{\partial^2}{\partial x^2}h(x) > 0$, which is when the growth is convex. This has an optimal solution of assigning agents to the largest task, and balancing the assignment equally among the largest tasks when ties exist. There are no known optimal solutions if tasks have different families of growth functions. For example if the first task grows convexly and the second task grows with a concave function, no known optimal solution exists even without travel time. Functions that have both concave and convex parts, such as the sigmoid function, also have unknown optimal solutions.

Table 1 compares the algorithms across seven different sets of growth functions, $h_i(x)$, which each have different optimal solutions. Each algorithm is largely deterministic, but we provided the average over five runs to ensure small tie breakers did not effect the results. The settings of the experiment, shown in Table 1, are: two convex, one linear, two concave, one group of tasks with mixed functions and one function that has both concave and convex parts. The first convex function has two tasks, both with growth rate $h_i(x) = 0.000016 \cdot x^3$. The first task has an initial cost of 20, while the second task has an initial cost of 15. The second set of convex growth tasks all have a growth rate, $h_i(x) = 0.00019 \cdot x^2$ and ini-

tial costs $f_1^0 = 25, f_2^0 = 20, f_3^0 = 10$. For linear growth functions, $h_i(x) = 0.00360 \cdot x$ with initial costs $f_1^0 = 40, f_2^0 = 30, f_3^0 = 10$. The first concave tasks all have $h_i(x) = 0.020 \cdot \sqrt{x}$ and initial costs $f_1^0 = 20, f_2^0 = 15, f_3^0 = 10, f_3^0 = 5$. The second set of concave tasks grow very slowly, all with $h_i(x) = 0.02 \cdot \ln(x + 1)$ and initial costs $f_1^0 = 40$ and $f_2^0 = 30$. The $x + 1$ inside the logarithmic function is to ensure continuity of task growth when tasks near completion. Optimal solutions are not provided for the last two experiments, as there are no known solutions. The mixed category has one of each type of growth function: $h_1(x) = 0.00019 \cdot x^2$ and $f_1^0 = 25, h_2(x) = 0.00360 \cdot x$ and $f_2^0 = 10$ and $h_3(x) = h_i(x) = 0.020 \cdot \sqrt{x}$ with $f_3^0 = 10$. A growth rate corresponding to the sigmoid function is $h_1(x) = h_2(x) = 0.1 \cdot \frac{e^{-0.1 \cdot x}}{(1 + e^{-0.1 \cdot x})^2}$ and has two tasks with initial costs $f_1^0 = 40$ and $f_2^0 = 30$. This is the only growth function which does not have a continuous first derivative when tasks are completed, namely $h(0) > 0$. However, when the cost of a task reaches zero at some time \hat{t} , i.e. $f_i^{\hat{t}} = 0$, we set $h(f_i^t) = 0$ for all $t \geq \hat{t}$.

We see that max-sum performs quite close to optimal in all cases when the optimal is known, and only loses to ALLONONE on the sigmoid function. All strategies perform approximately the same for linear growth functions, as the theory suggests. The small differences are discretization effects of the agents. RT-LFF and UNIFORM only do well comparatively on the convex functions. The ALLONONE performs well in general on the concave functions and the sigmoid. Next we will give a detailed analysis of the reasons for the algorithms performance.

The optimal solution for convex functions spends the majority of the time with an equal number of agents on all tasks. Max-sum follows very close to the optimal solution's assignment at every step, and thus performs similarly. RT-LFF attempts to minimize the number of agents switched and causes both tasks to finish at the same time. With $h_i(x) = 0.000016 \cdot x^3$, RT-LFF start with 15 agents on the first task, the larger, and 5 on the second. It then slowly switches agents from the larger task to smaller task, until about halfway through the simulation. At this point, both tasks have the same cost and agent are split 10 on each for the rest of the simulation. UNIFORM also performs well for convex functions for the same reason. ALLONONE has a strategy very different than the optimal, so it is not surprising that it does poorly for this group of functions.

The optimal strategy for concave growth functions is to have all the agents on the smallest task. Again, max-sum finds this solution and performs competitively with the optimal solution. RT-LFF and UNIFORM perform poorly on this family of functions, as they both spread agents over all tasks. Although ALLONONE is similar to the optimal solution, it assigns all agents to the larger tasks first instead of the smaller ones. This causes ALLONONE to perform sub-optimally, but not as poorly as the drastically different strategies of RT-LFF and UNIFORM.

The optimal solution is not known for mixed functions. Max-sum outperforms all others in this case, but the assignments vary drastically over time as shown in Fig. 2. ALLONONE attempts to finish the first task, which grows considerably more than the others. RT-LFF also assigns most of the agents to this first task for the majority

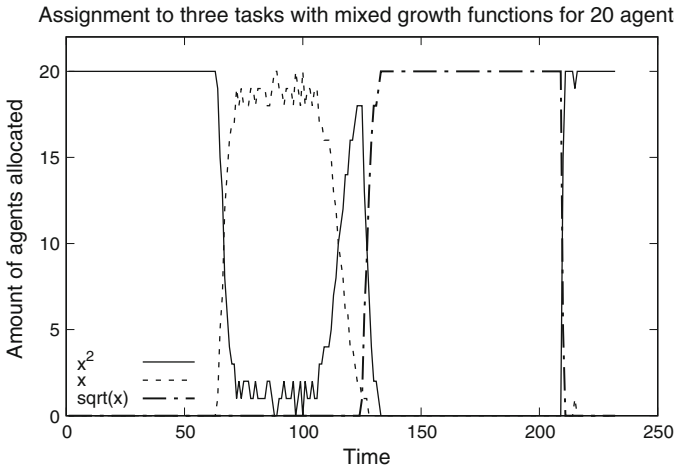


Fig. 2 Agent assignment across all three tasks with different growth types

of the simulation, causing RT-LFF and ALLONONE to perform similarly. UNIFORM does not assign enough agents to the important first task, initially 7, which is not enough to reduce the task growth. Eventually when the second task finishes about 25% through the simulation, the first task gets 10 agents which are able to overcome the task growth.

The sigmoid function has a steep slope when the cost is small. ALLONONE is able to overcome this fairly easily as all agents are on one task. Max-sum initially distributes agents across both tasks, but then moves them to a single task when the costs are smaller. Both RT-LFF and UNIFORM attempt to complete both tasks at the same time for the whole simulation, thus more effort to get past the steep slope when the cost is low.

Using ANOVA with repeated measures gives a p-value of 0.0716, which is close to being statistically significant so we did paired t-tests. The most different pair was max-sum and UNIFORM, with a p-value of 0.015. All other pairs did not reach statistical significance, including compared to the optimal when it was known. As this experiment spans different growth rates, no single algorithm could consistently be the best.

5.2 *RoboCup Rescue Simulator*

In this section we focus on fires in the RoboCup Rescue simulator. The RoboCup Rescue simulator is designed for urban search and rescue after an earthquake, where buildings collapse and fires start in buildings. The environment is complex with thousands of buildings and hundreds of agents in the full simulator which uses street

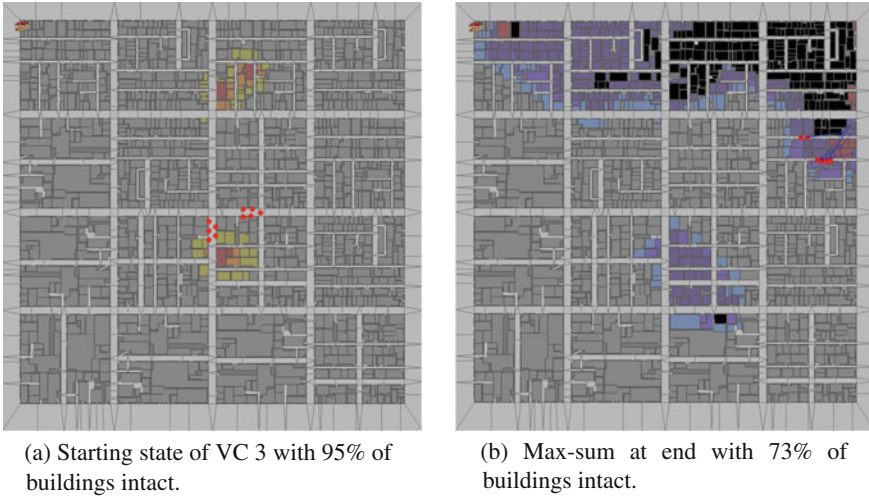


Fig. 3 Virtual City 4 (VC4)

maps of real cities. The full simulator uses heterogeneous agents, but for this work we focus only on the agents that can extinguish fires, i.e. firetrucks, through the use of the RMA SBench simulator extension [4]. Although RMA SBench is an extension of the RoboCup Rescue Agent Simulator, the communication is relaxed to allow many more messages to be sent than in the original simulator, along with multiple messages per time step.

Fires are the most dangerous hazard in RoboCup Rescue. Buildings heat up and catch on fire based on how many other nearby buildings are on fire. This creates a positive feedback loop, which causes fires to grow at exponential rates. A screenshot of the simulator is shown in Fig. 3a, b. Buildings on fire are yellow, orange and red in increasing intensity and temperature. When a fire is extinguished, buildings become blue or purple. When a building is destroyed it turns black.

We cluster nearby fires into a single task, where the cost is the number of buildings on fire. We use bottom-up hierarchical clustering with the Euclidean distance as the metric and the minimum distance between all pairs linkage criterion. Clustering ends when the distance between the closest pair of clusters is over 50 meters.

Both $h(x)$ and w need to be estimated. We approximate the cost growth function for all tasks as $h(f_i^t) = g \cdot f_i^t - |n_i^t| \cdot w$ as in Eqs. 3–5. To estimate g , buildings were allowed to burn unhindered for 100 simulation steps in 20 different tests. An exponential regression fit best, so we could model it as $a \cdot e^{b \cdot t}$ or simply $e^{b \cdot t}$, where b is the g we want. There are two regression models that minimize the sum of squares: $4.6852 \cdot e^{0.0393 \cdot t}$ or $e^{0.0561 \cdot t}$. We cannot use the first regression model since the scaling factor outside the exponent is reserved for the initial task cost. Thus we have to decide between the two g values, 0.0393 and 0.0561. We do it by looking at how they affect the distribution of w .

The work rate, w , was also empirically derived. A fixed small number of fires were repeatedly extinguished in 70 tests. If f_i^0 , g , $|n_i^t|$ and t_c are known, Eq. 4 can then be used to solve for w . The experiments found that w is rather noisy and typically has a few high outliers. This means in some cases, the fire trucks were able to extinguish the fire much more rapidly than the model anticipated. The lack of low outliers provides stability, since this indicates agents rarely extinguish poorly. We chose $g = 0.0393$, due to a smaller variance in w and fewer outliers.

This exponential model corresponds to the linear task growth from Sect. 5.1. In the simple simulator all strategies performed fairly identical in this case, but our model for RoboCup is an approximation of the real process. Max-sum is overly responsive to changes, so when a few buildings unexpectedly start on fire or get extinguished, agents are transferred. This can cause assignment thrashing for max-sum, as agents in transit are lost work and the solution quality decreases overall. RT-LFF is much more conservative and rarely thrashes, however it is unable to exploit smaller gains. This causes both methods to perform similarly in this domain on average.

Table 2 shows the percentage of damaged buildings for 5 variations of both the Virtual City (VC) and Paris map. Each variation is denoted by a number after the map name. Changes are to the number and location of fires along with the number and location of agents. Virtual City is a smaller map, so the number of agents and fires range between 8 to 15 and 30 to 50 respectively. In Paris there are between 10 to 50 agents with 50 to 120 fires. This randomization causes some configurations to be easier than others. Each configuration was run 5 times. Both RT-LFF and max-sum use the clustered building data as an approximation of f_i^t . Results on maps such as VC 5 are poor across all algorithms. This indicates more the hardness of the VC 5 configuration than the inefficiency of the algorithms. The opposite is the case in Paris 4, where the configuration is too easy and all algorithms performed well.

Often the initial configuration of a map makes it either too hard or too easy, which causes results to have bimodal distributions [10]. In this case the scores are similar

Table 2 Percent of buildings damaged at the end of simulation in RoboCup

Map	RT-LFF		Max-Sum	
	μ	σ	μ	σ
VC 1	72.14	7.23	83.82	6.03
VC 2	1.96	0.43	21.02	0.33
VC 3	50.41	9.70	26.19	6.36
VC 4	29.02	2.90	56.53	4.50
VC 5	76.90	2.69	82.42	1.34
Paris 1	15.47	4.55	27.46	5.03
Paris 2	58.89	7.00	68.63	8.73
Paris 3	39.27	4.69	31.19	3.92
Paris 4	8.76	0.45	9.01	0.52
Paris 5	25.22	6.36	20.36	5.66

across algorithms making differentiation more difficult. Due to the bimodal nature we apply the non-parametric Wilcoxon signed-rank test instead of the normal t-test. Neither method showed a statistically significant advantage over the other.

6 Conclusions and Future Work

In this work we focused on tasks with costs that grow over time and provided the first decentralized task assignment approach for this type of tasks. Our decentralized solution, which is based on the max-sum algorithm, performs similarly to the previous centralized solution in both a simple simulation and the RoboCup Rescue Agent Simulator. Our functions that predict the future cost of tasks assume that the number of agents active on a task is constant. However, sometimes it is beneficial to reassign agents before the task is completed. Future work will consider the effect of planned reassignments over time. The current work also assumes that these functions are estimated upfront, but they could be learned over the course of the simulation. Given different families of functions, the model could be refined over time to select the best function family with the best parameters. This could also help determine what errors come from modeling versus inherent randomness of the growth.

Acknowledgements Work supported in part by NSF-IIP-1439728 and the Graduate School of the University of Minnesota.

References

1. Amador, S., Okamoto, S., Zivan, R.: Dynamic multi-agent task allocation with spatial and temporal constraints. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1384–1390 (2014)
2. Farinelli, A., Rogers, A., Petcu, A., Jennings, N.R.: Decentralised coordination of low-power embedded devices using the max-sum algorithm. In: International Conference on Autonomous Agents and Multi-Agent Systems, pp. 639–646, May 2008
3. Kitano, H., Tadokoro, S.: RoboCup Rescue: a grand challenge for multiagent and intelligent systems. *AI Mag.* **22**(1), 39–52 (2001)
4. Kleiner, A., Farinelli, A., Ramchurn, S., Shi, B., Maffioletti, F., Reffato, R.: RMA SBench: Benchmarking dynamic multi-agent coordination in urban search and rescue. In: International Conference on Autonomous Agents and Multi-Agent Systems, pp. 1195–1196 (2013)
5. Kschischang, F., Frey, B., Loeliger, H.-A.: Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory* **47**(2), 498–519 (2001). Feb
6. Melvin, J., Keskinocak, P., Koenig, S., Tovey, C., Ozkaya, B.: Multi-robot routing with rewards and disjoint time windows. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2332–2337, Oct 2007
7. Modi, P.J., Shen, W.-M., Tambe, M., Yokoo, M.: ADOPT: asynchronous distributed constraint optimization with quality guarantees. *Artif. Intell.* **161**, 149–180 (2004)
8. Parker, J., Gini, M.: Tasks with cost growing over time and agent reallocation delays. In: Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems, pp. 381–388 (2014)

9. Parker, J., Gini, M.: Controlling growing tasks with heterogeneous agents. In: Proceedings of International Joint Conference on Artificial Intelligence (2016)
10. Parker, J., Godoy, J., Groves, W., Gini, M.: Issues with methods for scoring competitors in RoboCup Rescue. In: Autonomous Robots and Multirobot Systems at AAMAS (2014)
11. Peña-Alba, T., Vinyals, M., Cerquides, J., Rodríguez-Aguilar, J.A.: A scalable message-passing algorithm for supply chain formation. In: Proceedings of AAAI Conference on Artificial Intelligence, pp. 1436–1442 (2012)
12. Pujol-Gonzalez, M., Cerquides, J., Farinelli, A., Meseguer, P., Rodríguez-Aguilar, J.A.: Efficient inter-team task allocation in RoboCup Rescue. In: Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems, pp. 413–422 (2015)
13. Pujol-Gonzalez, M., Cerquides, J., Meseguer, P., Rodríguez-Aguilar, J.A., Tambe, M.: Engineering the decentralized coordination of UAVs with limited communication range. In: Advances in Artificial Intelligence, pp. 199–208. Springer (2013)
14. Ramchurn, S., Farinelli, A., Macarthur, K., Polukarov, M., Jennings, N.: Decentralised coordination in RoboCup Rescue. *Comput. J.* **53**(9), 1–15 (2010)
15. Tarlow, D., Givoni, I.E., Zemel, R.S.: Hop-map: efficient message passing with high order potentials. In: Proceedings of International Conference on Artificial Intelligence and Statistics, pp. 812–819 (2010)
16. Zheng, X., Koenig, S.: Reaction functions for task allocation to cooperative agents. In: Proceedings of International Conference Autonomous Agents and Multi-Agent Systems, pp. 559–566 (2008)
17. Zivan, R.: Anytime local search for distributed constraint optimization. In: Proceedings International Conference on Autonomous Agents and Multi-Agent Systems, pp. 1449–1452 (2008)

Part XI
Robot Control

LuGre Model Based Hysteresis Compensation of a Piezo-Actuated Mechanism

Guangwei Wang and Qingsong Xu

Abstract This paper presents a combined feedforward plus feedback control approach to compensate the hysteresis effect, which degrades the positioning accuracy of piezo-actuated mechanism. The LuGre friction model is extended to represent the nonlinear dynamics of the piezo-actuated positioning mechanism, and then the unknown model parameters are identified with the particle swarm optimization (PSO). Based on the developed mathematical model, the inverse LuGre model based feedforward plus feedback control is designed for the motion tracking control. Experimental results show that the LuGre model based hybrid control approach achieves a satisfactory position tracking performance. Owing to a simple structure, the proposed control approach can be implemented in other types of hysteretic systems.

Keywords Piezoelectric actuators · Hysteresis model · LuGre model · Micropositioning · Precision motion control

1 Introduction

Micropositioning system driven by piezoelectric actuators (PEAs) has become an important technique in modern manufacturing and process industries during the past decade. Due to its great capability in terms of high resolution, acceleration, and energy density, PEAs have been widely used in a variety of applications, such as scanning electron microscopes (SEM) [1], biological manipulation [2], and micro assembly [3]. However, PEAs usually exhibit inherit nonlinearities, mainly the hysteresis effect, in their response to applied input voltage, because the materials of PEAs are ferroelectric. This kind of drawback leads to serious problems of inaccuracy and unstable situation if the PEA is operated in an open-loop type. Thus, these nonlinearities have to be well compensated if a high accuracy is required.

G. Wang · Q. Xu (✉)

Faculty of Science and Technology, Department of Electromechanical Engineering,
University of Macau, Avenida da Universidade, Taipa, Macau, China
e-mail: qsxu@umac.mo

In the literature, numerous techniques have been proposed to analyze and compensate for the hysteresis nonlinearities of PEAs. Previous research in hysteresis modeling and compensation can be classified into three categories [4] in terms of: (1) electric charge control; (2) closed-loop position control; (3) inverse hysteresis model based feedforward control. The charge control method can reduce hysteresis and drift remarkably. The hysteresis is likely to be reduced to one-fifth of the original with the proposed differential feedback charge amplifier [5]. A feedforward charge control system, which controls the velocity of a piezoelectric actuator by the current fed to the actuator, is proposed in [6]. This work found that the motion of PEAs is a function of current and voltage. However, this approach requires extra electric circuits to amplify the induced charge, that leads to the increase of experimental complexity and hardware cost. Moreover, the output displacement of PEAs reduces significantly with this control method [6].

The second category is mostly used in commercial systems, in which strain gages, capacitive sensors, or optical sensors are used as the feedback sensors. For example, a closed-loop controlled nanomanipulation system for operation inside a scanning electron microscope (SEM) is reported in [1], where the position sensor-based closed-loop probing approach achieves a probing accuracy of 3 nm. Other related closed-loop approaches have been proposed, including the second-order discrete-time terminal sliding-mode control [7], nonlinear model predictive controller based on neural-network method [8], and adaptive controller with unknown actuator hysteresis [9]. However, these control algorithms are hard to realize due to the complexity and implementation cost in practical applications.

The third method employs a mathematical model that can describe the hysteretic phenomenon approximately, and then implements a feedforward controller based on the inverse hysteresis model to linearize the output of piezoelectric actuator. The hysteresis mathematical models can be grouped as static and dynamic models [10]. Although the static models can accurately represent the hysteresis of PEAs and they have been effectively used in model-inversion compensation approaches, the dynamics of PEAs cannot be described with these models. The dynamic models, that not only can describe the hysteretic characteristics mathematically but also can represent the dynamics of PEAs, are formulated by nonlinear differential equations. However, the modeling accuracy of the dynamic models is strongly dependent on parameter identification. In order to determine the parameters of dynamic models, intelligent algorithms have been used in system identification recently. For instance, recurrent fuzzy neural network [11] is proposed for hysteresis estimation of piezoelectric actuator to improve the control performance. The identification method based on real-coded genetic algorithm [12] is employed for the identification of Bouc-Wen hysteretic model. Particle swarm optimization (PSO) algorithm [13] is proposed for identification of Prandtl-Ishlinskii model for piezoelectric actuators.

This paper is focused on the third approach for achieving a precision position control. Conventionally, the LuGre friction model [14] shows majority of the known friction behaviors of mechanical systems, such as stiction effect, stick-slip fric-

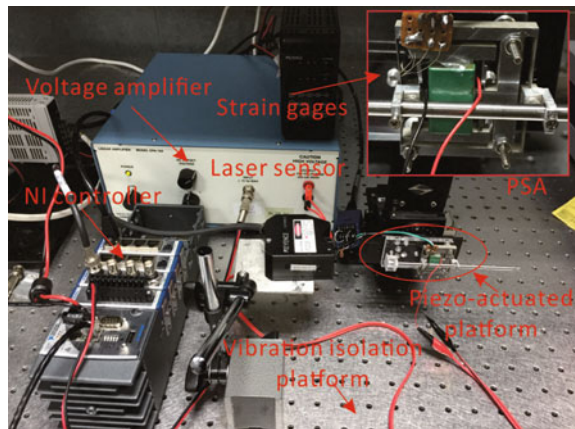
tion, coulomb friction and hysteresis in mechanical motion dynamics. It has been extended to describe the motion dynamics of piezo-actuated positioning system [2]. Owing to the promising capability for capturing the dynamic hysteretic effects, in this paper, the LuGre friction model is adopted to represent the hysteresis of piezo-electric actuator. As compared with other types of dynamic hysteresis model, the LuGre model is able to capture the hysteresis nonlinearity more precisely with fewer model parameters. In this work, a mathematical model is developed by a differential equation of the motion system with the addition of hysteresis function. PSO algorithm is employed for the LuGre hysteresis model identification. Then, the LuGre inverse model based feedforward controller plus an incremental PID feedback controller is designed to reveal the feasibility of the proposed model. The performance of the modeling and control approaches is validated by a series of experimental studies.

2 System Description

2.1 Experimental Setup

The experimental setup is shown in Fig. 1. The device is a flexure-based compliant mechanism, which is developed for biological cell injection. More details about this injector can be found in [15]. This mechanism is driven by a PEA (model: TS18-H5-202, from Piezo System, Inc.). The PEA is actuated by a high-voltage amplifier (model: EPA-104, from Piezo System, Inc.), by which the voltage will be enlarged by ten times to drive the flexure-based compliant injector and deliver a maximum displacement of 145 μm . Two strain gages (model: TP-3.8-350, from Bengbu Tian-

Fig. 1 Experimental setup for a piezo-actuated mechanism



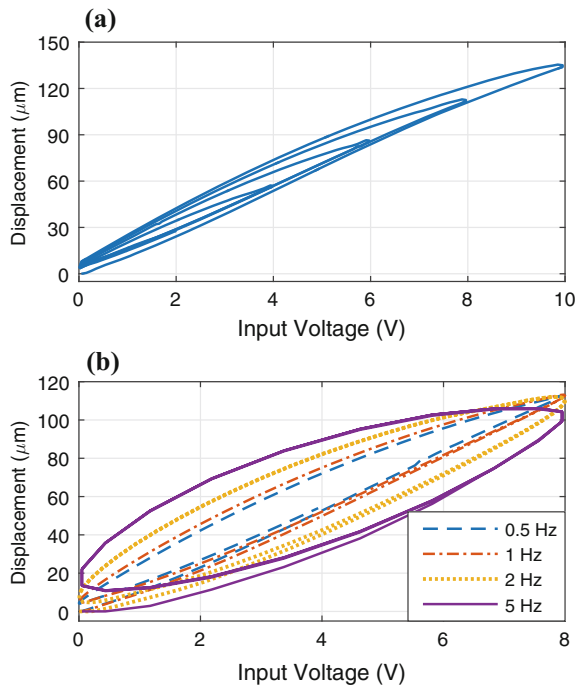
guang Sensor, Ltd.) construct a half-bridge circuit and are adopted to measure the output displacement of this stage. A laser sensor (model: LK-H055, from Keyence Corp.) is used for strain gages calibration. Natural Instruments (NI) cRIO-9022 real-time controller integrated with cRIO-9118 reconfigurable chassis, which contains a Field-Programmable Gate Array (FPGA) module, are adopted to implement the control law. NI-9263 analog output module and NI-9237 bridge analog input are employed to produce the driving voltage and obtain the strain gage sensor signals, respectively.

2.2 Hysteresis Modeling with LuGre Friction Model

By applying triangular wave voltage with variable amplitudes and frequencies, the curves between the displacement output and voltage input are depicted in Fig. 2. It can be observed that the hysteresis shape are dependent not only on the amplitude but also on the frequency of input voltage. Thus, a dynamic hysteresis model is required for the purpose of representing dynamics of piezo-actuated system accurately.

In this work, LuGre friction model is extended to describe the hysteresis behavior of the piezo-actuated system. In LuGre friction model, the friction phenomenon of two surfaces in contact exhibit a number of their asperities, which is considered

Fig. 2 Hysteresis loops obtained by applying triangular voltage input with different amplitudes **a** and sinusoidal inputs voltage with different frequencies **b**



as two rigid bodies in contact through elastic bristles. The bristles will deflect like springs which give rise to the friction force when a tangential force is applied. The hysteresis friction force F_H generated from bristle bending can be derived as follows:

$$F_H = \sigma_0 z + \sigma_1 \frac{dz}{dt} + \sigma_2 \dot{x} \tag{1}$$

with

$$\frac{dz}{dt} = \dot{x} - \frac{|\dot{x}|}{h(\dot{x})} z \tag{2}$$

where \dot{x} is the relative velocity between the contact surfaces, σ_0 , σ_1 and σ_2 are the hysteresis friction force parameters that can be physically explained as the bristle stiffness, bristle damping and viscous coefficients, respectively. z denotes the average deflection of the bristles with contact force applied and $h(\dot{x})$ denotes the Stribeck effect curve, which is described by:

$$h(\dot{x}) = \frac{F_c + (F_s - F_c)e^{-(\dot{x}-\dot{x}_s)^2}}{\sigma_0} \tag{3}$$

where F_c is the level of coulomb friction, F_s is the static friction force level, and \dot{x}_s is the stribeck velocity. By substituting (1) into (2), the hysteresis friction force is rewritten as

$$\begin{aligned} F_H &= \sigma_0 z - \sigma_1 \frac{|\dot{x}|}{h(\dot{x})} z + (\sigma_1 + \sigma_2) \dot{x} \\ &= (\sigma_1 + \sigma_2) \dot{x} + F_d(z, \dot{x}). \end{aligned} \tag{4}$$

The first term $(\sigma_1 + \sigma_2) \dot{x}$ is a function of the velocity. The second term $(\sigma_0 - \sigma_1 |\dot{x}|/h(\dot{x}))z$ is scaled by z due to the dynamical perturbations in hysteresis. Since z and $h(\dot{x})$ are bounded, it follows that

$$|F_d(z, \dot{x})| = |(\sigma_0 - \sigma_1 \frac{|\dot{x}|}{h(\dot{x})})z| \leq k_1 + k_2 |\dot{x}| \tag{5}$$

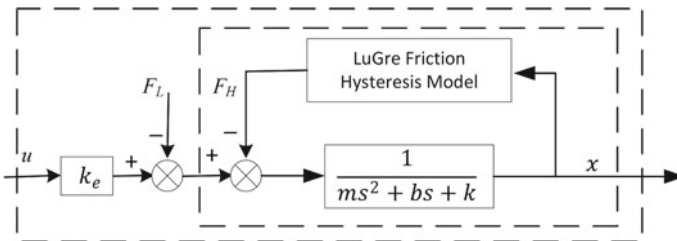


Fig. 3 Scheme of the proposed model of a piezo-positioning mechanism

The overall dynamics model of piezo-actuated positioning systems is shown in Fig. 3. The mathematical model can be expressed as

$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) + F_H + F_L = k_e u(t) \quad (6)$$

where m , k , and b are the mass, stiffness, and damping coefficient of the piezo-actuated positioning mechanism, respectively. x is the output displacement, u denotes the input voltage, F_H is the hysteresis friction function, F_L is the perturbation of the system arising from model parameter uncertainties, unmodeled dynamics and other unknown terms, and k_e denotes the nominal voltage-to-force coefficient.

Substituting (4) into (6), the LuGre model can be rewritten as

$$\begin{aligned} \ddot{x}(t) &= \frac{1}{m}(k_e u(t) - b\dot{x}(t) - kx(t) - F_H - F_L) \\ &= \frac{k_e}{m} u(t) - \frac{1}{m}(kx(t) + (\sigma_1 + \sigma_2 + b)\dot{x}(t) \\ &\quad + \sigma_0 z - \sigma_1 \frac{|\dot{x}(t)|}{h(\dot{x})} z + F_L). \end{aligned} \quad (7)$$

The hysteresis model describes the relationship between the input voltage and output displacement of the piezo-actuated stage. Whereas a corresponding input voltage can be produced with a desired output displacement by the inverse hysteresis model. The inverse model can be easily derived from (6).

3 Parameter Identification

3.1 Parameter Identification Method

Particle Swarm Optimization (PSO) algorithm is a stochastic optimization technique that was developed by Kennedy and Eberhart in 1995 [16]. PSO is randomly initialized by a population of particles in a n -dimensional search space (n is the number of identified variables) and optimized by updating generations. Each particle is related to two vectors: the position vector $x = [x^1, x^2, \dots, x^n]$ and the velocity vector $v = [v^1, v^2, \dots, v^n]$. The best solution is recorded as local best particle p_{id} in the population space, and the global best value p_{gd} is obtained from all local best values. The position and velocity of particles are updated as follows.

$$\begin{aligned} v_{id}^{k+1} &= \omega v_{id}^k + c_1 \xi (p_{id}^k - x_{id}^k) + c_2 \eta (p_{gd}^k - x_{id}^k) \\ x_{id}^{k+1} &= x_{id}^k + r v_{id}^{k+1} \end{aligned} \quad (8)$$

where v_{id}^k and x_{id}^k is the k th velocity and position of the id th particle respectively, k is the current iteration ($1 \leq k \leq m$), m is the maximum iteration number. id is the id th particle of the swarm ($1 \leq id \leq n$), n is the particle number of the swarm, c_1 and c_2 are constants that are acceleration coefficients called cognitive and social parameters, respectively. ξ and η are random numbers between $[0, 1]$. ω is inertia weight. p_{id} is the best position found by the id th particle, and p_{gd} is the global best position found by the swarm.

It has been shown that the linearly decreasing inertia weight has better convergence performance in the searching procedure. In this research, the weight ω is defined to decrease linearly from $\omega_{max} = 0.9$ to $\omega_{min} = 0.4$ as follows, while the PSO is evolving.

$$\omega_k = \omega_{max} - k \frac{\omega_{max} - \omega_{min}}{m} \quad (9)$$

This strategy aids to the global search at the beginning and to the local search at the end of the iterative process.

The parameters $(\sigma_0, \sigma_1, \sigma_2, \dot{x}_s, F_c, F_s, m, b, k, k_e)$ of the LuGre friction model are identified by minimizing the following fitness function:

$$F(\sigma_0, \sigma_1, \sigma_2, \dot{x}_s, F_c, F_s, m, b, k, k_e) = \frac{1}{N} \sum_{i=1}^N (x_i - x_i^m)^2 \quad (10)$$

where N is the total number of samples, and $(x_i - x_i^m)$ denotes the error of the i th sample which is calculated as the deviation of LuGre model output x_i^m from experimental result x_i .

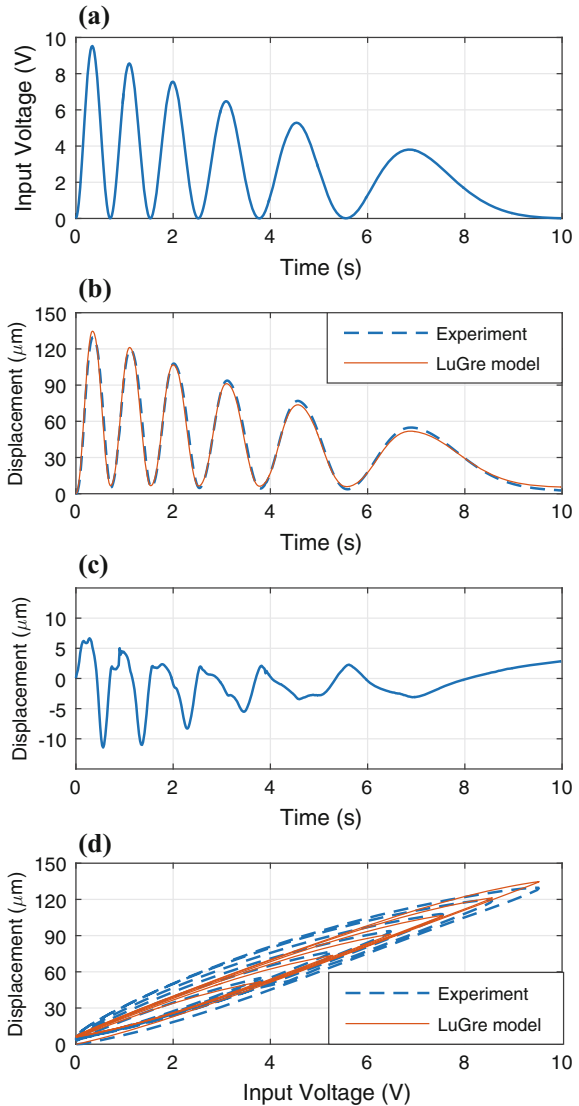
3.2 Parameter Identification

To identify the hysteresis model parameters, a sinusoidal voltage signal with varying amplitude and frequency (as shown in Fig. 4a), which can cover the dynamic characteristics in the targeted cell manipulation application, is chosen to collect the experimental data:

$$u(t) = 5e^{-0.1t} [\cos(3\pi te^{-0.092t} - \pi) + 10]. \quad (11)$$

By setting 500 training data sets with a sampling time 0.02 s as shown in Fig. 4a, b, the LuGre model is identified by optimizing the nine parameters to minimize the fitness function (10). The parameter identification is conducted with Matlab and the identified parameters are given in Table 1. According to the identified model, the displacement outputs from the experiment and LuGre model are compared in Fig. 4b–d.

Fig. 4 Testing results of the LuGre model. **a** Input voltage; **b** experimental result and LuGre model output; **c** LuGre model output errors; **d** voltage-displacement hysteresis loops



The results indicate that the LuGre model can match the actual output of the piezo-actuated system with a root-mean-square error (RMSE) of 2.93 % with respect to the travel range. In addition, Fig. 4c shows that the LuGre model errors are uniformly distributed even with the varying frequencies and amplitudes of the input signal. Therefore, the identified LuGre model captures the amplitude and rate-dependent hysteresis accurately.

Table 1 Identified dynamic model parameters

Parameter	Search range	Identified value
m	[0.01, 1]	1.33 kg
b	[10^3 , 10^4]	9.5×10^3 Ns/m
k	[10^6 , 10^7]	1.57×10^6 N/m
σ_0	[10^4 , 10^6]	8.96×10^5 N/m
σ_1	[100, 500]	480 Ns/m
σ_2	[0, 10]	1.656 Ns/m
F_c	[0, 30]	13.1 N
F_s	[0, 30]	20.4 N
\dot{x}_s	[0, 0.1]	0.001 m/s
k_e	[0, 10]	2.46 m/V

4 Experimental Verification

4.1 Controller Design

Based on the hysteresis model as built in the previous sections, the combined feed-forward (FF) plus feedback (FB) control algorithm is implemented and verified for the hysteresis compensation in this section. Based on the inverse LuGre model, the FF control is implemented for the nonlinearity compensation. The FB control based on incremental PID algorithm is realized owing to its robustness and popularity. The combined FF+FB control output can be derived as

$$\begin{aligned}
 u(t) &= u_{FF}(t) + u_{FB}(t) \\
 &= u_{FF}(t) + u_{FB}(t-1) + K_p[e(t) - e(t-1)] \\
 &\quad + K_i e(t) + K_d[e(t) - 2e(t-1) + e(t-2)]
 \end{aligned} \tag{12}$$

where e is the position tracking error, $u_{FB}(t-1)$ represents the FB control value in the previous time step, $u_{FF}(t)$ is the output of inverse LuGre hysteresis model based on the desired input position, and K_p , K_i and K_d are the proportional, integral, and derivative gains, respectively. In this control experiment, the PID gains are tuned as $K_p = 0.08$, $K_i = 1.12 \times 10^{-3}$, and $K_d = 1.03 \times 10^{-6}$.

4.2 Controller Verification

4.2.1 Generalization Study

In order to reveal the efficiency of the hybrid control, the testing results of stand-alone FF, FB and the combined FF+FB controllers based on the inverse LuGre model are depicted in Fig. 5. The control performances of these three controllers are shown in Table 2.

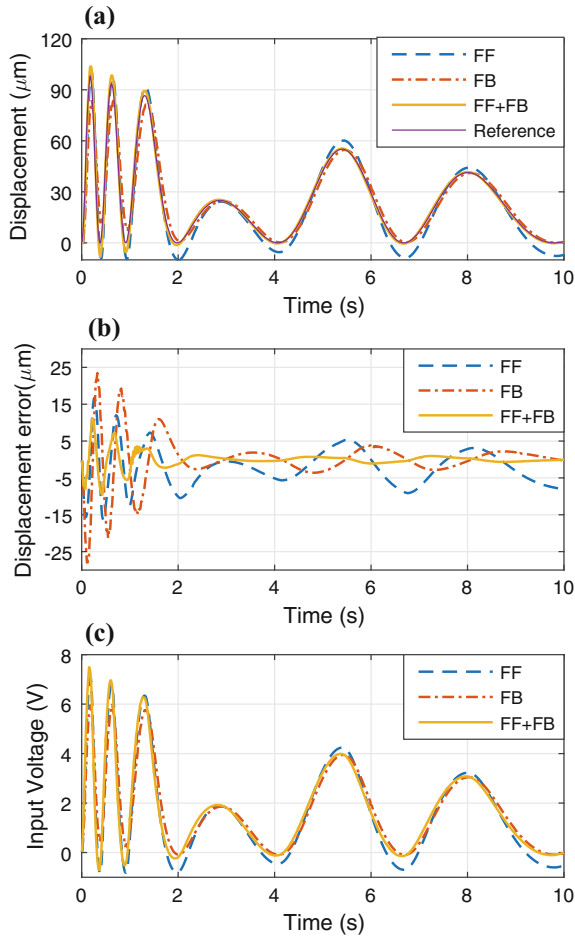


Fig. 5 LuGre hysteresis model based stand-alone and hybrid control results of reference signal #1. **a** Position tracking results; **b** position tracking error; **c** control effort of the controller

Table 2 Control performances of the different controllers

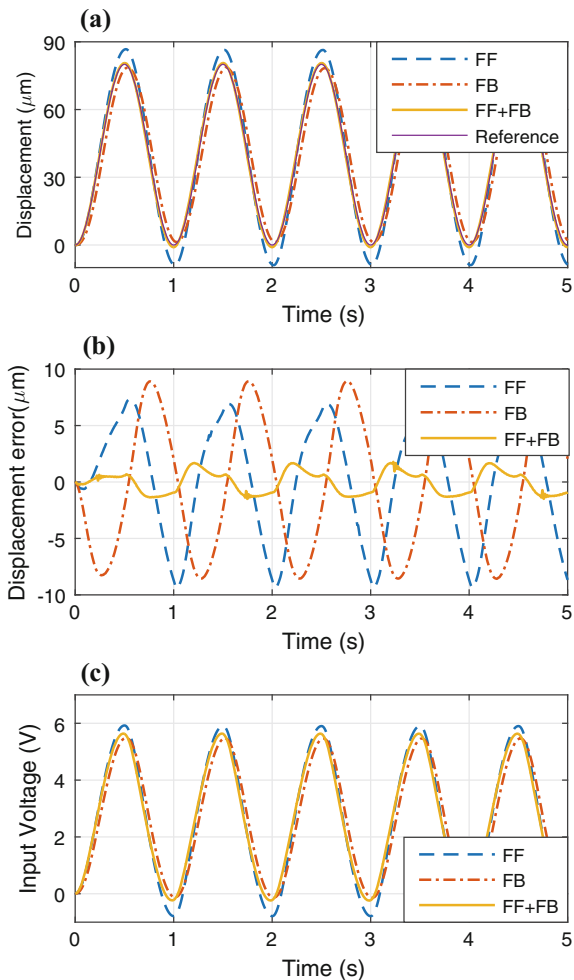
Input	Performance	FF	FB	FF+FB
Signal #1	MAE	4.56 μm	3.49 μm	1.08 μm
		3.51 %	2.66 %	0.81 %
	RMSE	5.64 μm	5.90 μm	1.96 μm
		3.82 %	4.54 %	1.48 %
Sine wave	MAE	4.60 μm	5.50 μm	0.86 μm
		5.11 %	6.11 %	0.95 %
	RMSE	5.26 μm	6.13 μm	0.97 μm
		5.84 %	6.81 %	1.07 %

It can be observed that the inverse LuGre model based FF controller results in a mean absolute error (MAE) of 3.51 % and a RMSE of 3.82 %. The FB controller leads to a MAE of 2.66 % and a RMSE of 4.54 %. As compared to the FF and FB controllers, the combined FF+FB controller can significantly improve the performance over stand-alone FF and FB control. It reduces the position tracking errors in terms of RMSE by 61.3 % and 67.4 %, respectively, which reveals the effectiveness of the proposed control scheme.

4.2.2 Sinusoidal Position Tracking

As for the biological cell injection application, the input signal usually adopts a lower frequency. In order to further evaluate the performance of proposed control scheme

Fig. 6 LuGre hysteresis model based control results with 1-Hz sinusoidal input. **a** Position tracking results; **b** position tracking error; **c** control effort of the controller



for the particular application, a 1-Hz sinusoidal desired displacement is implemented for the tracking control, as shown in Fig. 6.

It is observed that the stand-alone FF and FB controllers still produce relatively large tracking errors. While the maximum position error of the combined FF+FB controller is 1.67 μm . In addition, the MAE and RMSE are 0.86 and 0.97 μm , which are equivalent to 0.95 and 1.07 % of the overall motion range, respectively.

5 Conclusions

An inverse hysteresis model based feedforward plus feedback position controller is proposed in this paper for precision position control of a piezo-actuated mechanism. The LuGre model is extended to represent the dynamics of a piezo-actuated positioning mechanism. Then, PSO algorithm is implemented to identify the unknown parameters of LuGre model. The proposed LuGre model represents the hysteresis effects of piezo-actuated mechanism accurately and effectively as validated by experimental study. Experimental results also indicate that the hybrid control scheme using LuGre inverse model based feedforward control combined with an incremental PID control algorithm can effectively compensate hysteresis nonlinearity for the precise position tracking application. Moreover, the presented control scheme can be easily adapted to other smart actuator-driven mechanism. In the future, the proposed control scheme will be used for position tracking dedicated to biological cell injection application.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant No. 51575545, the Macao Science and Technology Development Fund under Grant No. 090/2015/A3 and 052/2014/A1, and the Research Committee of the University of Macau under Grant No. MYRG078(Y1-L2)-FST13-XQS.

References

1. Zhou, C., Gong, Z., Chen, B., Cao, Z., Yu, J., Ru, C., Tan, M., Xie, S., Sun, Y.: A closed-loop controlled nanomanipulation system for probing nanostructures inside scanning electron microscopes. *IEEE/ASME Trans. Mechatron.* **PP**(99), 1–9 (2016)
2. Putra, A.S., Huang, S., Tan, K.K., Panda, S.K., Lee, T.H.: Design, modeling, and control of piezoelectric actuators for intracytoplasmic sperm injection. *IEEE Trans. Control Syst. Technol.* **15**(5), 879–890 (2007)
3. Xu, Q.: Robust impedance control of a compliant microgripper for high-speed position/force regulation. *IEEE Trans. Ind. Electron.* **62**(2), 1201–1209 (2015)
4. Ang, W.T., Khosla, P.K., Riviere, C.N.: Feedforward controller with inverse rate-dependent model for piezoelectric actuators in trajectory-tracking applications. *IEEE/ASME Trans. Mechatron.* **12**(2), 134–142 (2007)
5. Comstock, R.H.: Charge control of piezoelectric actuators to reduce hysteresis effects. U.S. Patent No. 4,263,527. 21 Apr 1981

6. Ronkanen, P., Kallio, P., Vilkkö, M., Koivo, H.N.: Displacement control of piezoelectric actuators using current and voltage. *IEEE/ASME Trans. Mechatron.* **16**(1), 160–166 (2011)
7. Xu, Q.: Piezoelectric nanopositioning control using second-order discrete-time terminal sliding-mode strategy. *IEEE Trans. Ind. Electron.* **62**(12), 7738–7748 (2015)
8. Cheng, L., Liu, W., Hou, Z.G., Yu, J., Tan, M.: Neural-network-based nonlinear model predictive control for piezoelectric actuators. *IEEE Trans. Ind. Electron.* **62**(12), 7717–7727 (2015)
9. Wang, F., Liu, Z., Zhang, Y., Chen, C.P.: Adaptive robust image-based visual servoing control of robot with unknown actuator hysteresis. *Nonlinear Dyn.* 1–15 (2016)
10. Zhu, W., Wang, D.H.: Non-symmetrical bouccwen model for piezoelectric ceramic actuators. *Sens. Actuators A: Phys.* **181**, 51–60 (2012)
11. Lin, F.J., Shieh, H.J., Huang, P.K., Teng, L.T.: Adaptive control with hysteresis estimation and compensation using rfnn for piezo-actuator. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **53**(9), 1649–1661 (2006)
12. Charalampakis, A.E., Dimou, C.K.: Identification of bouccwen hysteretic systems using particle swarm optimization. *Comput. Struct.* **88**(21–22), 1197–1205 (2010)
13. Yang, M.J., Gu, G.Y., Zhu, L.M.: Parameter identification of the generalized prandtlcishlinskii model for piezoelectric actuators using modified particle swarm optimization. *Sens. Actuators A: Phys.* **189**, 254–265 (2013)
14. De Wit, C.C., Olsson, H., Astrom, K.J., Lischinsky, P.: A new model for control of systems with friction. *IEEE Trans. Autom. Control* **40**(3), 419–425 (1995)
15. Liu, Y., Zhang, X., Zhang, Y., Xu, Q.: Design and control of a novel piezo-driven micro-injector. In: 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1985–1990. IEEE (2015)
16. Kennedy, J.: Particle swarm optimization. In: *Encyclopedia of Machine Learning*. Springer, pp. 760–766 (2011)

Sample-Data Control of Optimal Tracking for a Class of Non-linear Systems via Discrete-Time State Dependent Riccati Equation

Fan Yang, Guoliang Zhang, Zhenan Pang and Lei Yuan

Abstract For the Optimal tracking problem of a class affine non-linear system, based on discrete-time State Dependent Riccati Equation(DSDRE), an optimal tracking sample-data control method is proposed. Moreover, the optimization is discussed with relation between HJBE and DSDRE; stability of close-loop system is investigated with Lyapunov theorem. Finally, the method is the verified in numerical simulations.

Keywords Optimal tracking control · Non-linear system · DSDRE

1 Introduction

The optimal tracking control of non-linear systems is a traditional problem, which attracts many researchers. In general, the sufficient condition of dealing with an optimal control problem is to solve the Hamilton-Jacobi-Bellman equation (HJBE). In Linear-Time-Invariant (LTI) systems, the HJBE can be expressed in an algebraic Riccati equation (ARE) in simplification, which can be solved analytically. However, in non-linear systems, HJBE is difficult to solve. In order to overcome this difficulty, many control methods has been proposed, such as inverse optimal adaptive control [1], control Lyapunov function technology [2], robust integral of the sign of the error (RISE) method [3, 4], adaptive dynamic programming [5] and so on.

In last decades, the State Dependent Riccati Equation (SDRE) method is well known and has become very popular in control problem of non-linear systems. The

F. Yang (✉) · G. Zhang · Z. Pang
Hi-Tech Institution of Xi'an, Xi'an, Shaanxi Province, China
e-mail: y_fanfan@yeah.net

F. Yang
High and New Technology Institution of Baoji, Baoji, Shaanxi Province, China

L. Yuan
R&D Center, AVIC Aircraft Co. Ltd., Yanliang, Shaanxi Province, China

methods based on SDRE are very effective algorithms in synthesizing non-linear feedback control. Using State-dependent coefficient (SDC) matrixes, the non-linear systems can be transform to a linear structure (nonunique), then, the ARE can be solved on-line to give the suboptimum control law. (In combination with State-dependent coefficient (SDC) matrices, the non-linear system can be expressed into non-unique linear structures. In this way, the suboptimum control law can be obtained through online solution of the ARE). In theory, Cloutier et al. [6], Mracek and Cloutier [7] noted that the SDRE feedback schemes for infinite-time nonlinear optimal control problem, and locally asymptotically stable and locally asymptotically optimal is proved. Ornelas-Tellez et al. [8] analysis the optimal tracking problem of non-linear systems and prove the asymptotically stable and optimal in continue-time system. A survey of SDRE control can be found in literature [9]. On the other hand, in practice, this method is widely used in manipulator control [10], the attitude control of satellite [11], tank motion control [12], the underwater vehicle control [13], and obtained better result.

The sample-data control is to design the discrete-time controller for continuous-time plants, which has recently been noted because of the advancement of digital techniques and the requirement of engineering practice, such as space robots and flight-manipulators. Generally, there are basically two classes of design method in sample-data control. In one class of design method, the sample-data controller comes from discretization of the continuous-time control law designed based on continuous-time plant [14, 15]. The other class of design method is to design the discrete-time controller based on discrete-time model directly [16, 17]. As is known to us all, it is very difficult to describe non-linear systems in the exact discretion. So no matter which method is applied, it is on the approximation of the original system.

As the characters of SDRE is bring up non-linear systems to the linear structure so the author think it is suitable for designing the sample-data controller. To deal with sample-data control in a class of non-linear systems, in this paper there presents a sample-data control of optimal tracking scheme via Discrete-time State Dependent Riccati Equation (DSDRE). At first, the non-linear systems are illustrated in a pseudo linear structure in advantage of SDC matrixes, and at the same time the discrete controller for the system is designed based on DSDRE theory. Finally, the stability and optimality of the control system is analyzed via Lyapunov theorem.

This paper is organized as follows. At beginning, the non-linear systems model and primary is introduced. Then, the sample-data controller is designed using DSDRE. In Sect. 3, the optimization is discussed with relation between HJBE and DSDRE; stability of close-loop system is investigated with Lyapunov theorem. The numerical simulations are presented to verify effectiveness of this method and the relation between parameter selection and sample period is briefly discussed via simulation in Sect. 4.

2 Model and Primary

In this section, a class of non-linear systems and SDRE tracking theory is introduced. Firstly, the infinite-time SDRE tracking control method is briefly expounded in order to be used next controller designing. After that, the feature of non-linear system is analyzed in this paper and according to which, the task of control is described.

2.1 The Infinite-Time SDRE Tracking Control Method

Let us consider the continuous-time non-linear system

$$\begin{cases} \dot{x}(t) = f(x) + g(x)u(t) \\ y(t) = h(x, t), \end{cases} \quad (1)$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the system input vector, $y \in \mathbb{R}^p$ is the system output vector; the function $f(x), g(x), h(x) \in C^k$ maps of appropriate dimensions.

If $f(x), g(x)$ and $h(x)$ satisfies the condition $f(0) = 0, g(x) \neq 0, \forall x$, the formula (1) can be transformed into the following pseudo linear structure via SDC matrixes

$$\begin{cases} \dot{x}(t) = A(x)x(t) + B(x)u(t) \\ y(t) = C(x)x(t), \end{cases} \quad (2)$$

where

$$f(x) = A(x)x(t), \quad g(x) = B(x)x(t), \quad h(x) = C(x)x(t).$$

If the $\{A(x), B(x)\}$ is controllable, $\{A(x), C(x)\}$ is observable in pointwise, the $A(x), B(x), C(x)$ are called SDC matrix.

If Defining $y_r(t)$ is the desired reference trajectory, which is tracked by system output $y(t)$, and the optimal cost function is following

$$\min J = \frac{1}{2} \int_0^\infty e^T Q(x) e + u^T R(x) u dt, \quad (3)$$

where $e(t) = y(t) - y_r(t)$, $Q > 0$ and $R > 0$, the optimal control law is [8, 12]

$$u^* = -R^{-1}(x)B^T(x)\{P(x)x - S(x)\}. \quad (4)$$

In this formula, the $P(x)$ satisfies the following differential Riccati equation

$$\begin{cases} P(x) = C^T(x)Q(x)C(x) + P(x)A(x) + \\ A^T(x)P(x) - P(x)B(x)R^{-1}(x)B^T(x)P(x) \\ P(\infty) = 0, \end{cases} \tag{5}$$

and $S(x)$ can be acquired by solving differential equation

$$\begin{cases} \dot{S}(x) = -k^{-1}(x)S(x) - C^T(x)Qy_r(t) \\ S(\infty) = 0 \end{cases} \tag{6}$$

where

$$K(x) = \left([A(x) - B(x)R^{-1}(x)B^T(x)P(x)]^T \right)^{-1} \tag{7}$$

Remark 1 Through transformation of formulas (5) and (6) in infinite-time from differential equation to algebraic equation, it is not difficult to solve $P(x)$ and $S(x)$. In the literature [13, 18] researchers suggest the solving procedure and its practice.

2.2 Modeling and Hypothesis

In this paper, the non-linear systems define as following

$$\dot{x}(t) = f(x) + g(x)u(t), \tag{8a}$$

$$\dot{y}(t) = J(x)x(t) \tag{8b}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input vector, $y \in \mathbb{R}^p$ is the system output vector; the function $f(x), g(x), J(x) \in C^k$ maps of appropriate dimensions, And $J(x)$ is the jacobian matrix, $J(x) = \frac{\partial h(x,t)}{\partial x}$.

Remark 2 Formula (8a) expresses a class of non-linear systems such as manipulator system, underwater robot, space robot and so on. In this class of systems, formula (8b) is named by kinematic equation. In general, it is more difficult to find the SDC matrix for output equation than kinematic equation.

To design the corresponding controller, these hypothesis are as follows

- (1) $y(t), y_r(t), x(t), \dot{x}(t)$ are bounded in tracking process, where $y(t)$ is system output, and $y_r(t)$ is the desired reference trajectory.
- (2) The output $y(t)$ can be accurately measured using sensor.
- (3) In formula (8a) and (8b), $\forall x, f(0) = 0, g(x) \neq 0$.

The purpose of controller designing is to find a sample-data control law, with which the system output $y(t)$ could catch up with the desired reference trajectory $y_r(t)$.

The purpose of controller designing is to find a sample-data control law, with which the system output $y(t)$ could catch up with the desired reference trajectory $y_r(t)$.

2.3 The Sample-Data Controller Designing

Based on formula (8a) and (8b), a new system is defined as following:

$$\begin{cases} \dot{x}(t) = f(x) + g(x)u(t) \\ y^*(t) = \dot{y}(t) = J(x)x(t), \end{cases} \tag{9}$$

As hypothesis (1) and (3), the formula (9) can be transformed into a linear structure with SDC matrixes

$$\begin{cases} \dot{x}(t) = A(x)x(t) + B(x)u(t) \\ y^*(t) = C(x)x(t), \end{cases} \tag{10}$$

In this paper, the designing method for the first class of sample-data controller is applied. Using zero-order hold method, the system (10) is discretized as following

$$\begin{cases} x_{k+1} = A_{dk}x_k + B_{dk}u_k \\ y_k^* = C_{dk}x_k, \end{cases} \tag{11}$$

where I is unit matrix; $x_k = x(kT)$, $y_k = y(kT)$, $u_k = u(kT)$; $A_{dk} = TA(x_k) + I$, $B_{dk} = TB(x_k)$, $C_{dk} = J(x_k)$, T is the sample period.

We define the tracking error as

$$e_k = y_k - y_{rk}, \tag{12}$$

Since the $\dot{y}(t)$ is not measurable, let us define a sliding variable z to estimate the $\dot{y}(t)$

$$z_k = \dot{e}_k + \lambda e_k, \tag{13}$$

where

$$\lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n), \lambda > 0.$$

Combining formula (12) and (13), we can get

$$y_k^* = z_k + \dot{y}_{rk} - \lambda e_k = C_d(x_k)x_k \tag{14}$$

Now let us restrict the desired reference trajectory

$$y_{rk}^* = \dot{y}_{rk} - \lambda e_k \tag{15}$$

and the optimal cost function

$$J(x_k, k) = \frac{1}{2} \sum_{n=k}^{\infty} (z_n^T Q z_n + u_n^T R u_n) \tag{16}$$

Theorem 1 *As to the system described by formula (11) in hypothesis (1), if $\{A(x), B(x)\}$ is controllable, and $\{A(x), C(x)\}$ is observable in pointwise. Q is positive semi-definite matrix; R is positive definite matrix. Then there is the optimal sample-data control law (17a) and (17b)*

$$u_k = - (R + B_{dk}^T P_{k+1} B_{dk})^{-1} [B_{dk}^T P_{k+1} A_{dk} x_k - s_k] \tag{17a}$$

$$S_k = \left[\frac{1}{2} [B_{dk} + I] \left[(R + B_{dk}^T P_{k+1} B_{dk})^{-1} B_{dk}^T P_k - I \right] A_{dk} + I \right]^{T-1} C_{dk}^T Q y_{rk}^* \tag{17b}$$

so as to trajectory tracking for system (11) along a desired trajectory y_{rk}^* , where P_k is the solution to the matrix difference equation

$$A_{dk}^T P_{k+1} A_{dk} - A_{dk}^T P_{k+1} B_{dk} (R + B_{dk}^T P_{k+1} B_{dk})^{-1} B_{dk}^T P_{k+1} A_{dk} - P_k + C_{dk}^T Q C_{dk} = 0, \tag{18}$$

and s_k is the solution to the vector difference Eq. (19).

$$\begin{aligned} & -y_{rk}^{*T} Q C_{dk} + \frac{1}{2} s_{k+1}^T (R + B_{dk}^T P_{k+1} B_{dk})^{T-1} B_{dk}^T P_{k+1} A_{dk} - s_{k+1}^T A_{dk} \\ & + \frac{1}{2} s_{k+1}^T B_{dk} (R + B_{dk}^T P_{k+1} B_{dk})^{-1} B_{dk}^T P_{k+1} A_{dk} + s_k^T = 0 \end{aligned} \tag{19}$$

Meanwhile, the control law (17a) and (17b) is optimal in the sense that it minimizes the cost function (16), which has an optimal value function given as

$$J^*(x_k, k) = \frac{1}{2} x_k^T P_k x_k - s_k^T x_k + \varphi_k, \tag{20}$$

where

$$\varphi_{k+1} - \varphi_k = \frac{1}{2} s_{k+1}^T B_d(k) (R + B_d^T(k) P_{k+1} B_d(k))^{-1} s_{k+1} - \frac{1}{2} r^T(k) Q r(k)$$

Moreover the system (8a) and (8b) output $y(t)$ is tracking the desired trajectory $y(t)$, and the tracking error e_k is exponential convergence to zero.

The proof of the Theorem 1 will spell out in next section.

Remark 3 Because of the boundary condition, it is very difficult to solve the Eqs. (18) and (19) in finite-time. But according to the optimal control theory, with $P_{k+1} = P_k$ and $s_{k+1} = s_k$, the formula (18) and (19) reduce to the algebra equation which can be on-line solve at every sample point in infinite-time condition.

3 Optimal and Stability

In this section, the relation between DSDRE and HJBE is presented, and the stable is investigated. Then the proof of Theorem 1 is descried in detail.

3.1 The Relation Between DSDRE and HJBE

Let us rearranging (16) to (21)

$$\begin{aligned}
 J(x_k, k) &= \frac{1}{2} (z_k^T Q z_k + u_k^T R u_k) + \frac{1}{2} \sum_{n=k+1}^{\infty} (z_n^T Q z_n + u_n^T R u_n) \\
 &= \frac{1}{2} (z_k^T Q z_k + u_k^T R u_k) + J(x_{k+1}, k + 1)
 \end{aligned}
 \tag{21}$$

According to the Bellman’s principle of optimality, the optimum J_k^* satisfies that the following discrete-time HJBE:

$$J_k^* = \min_{u_k} \left[\frac{1}{2} (z_k^T Q z_k + u_k^T R u_k) + J(x_{k+1}, k + 1) \right]
 \tag{22}$$

The optimal input u_k^* satisfies following partial derivative equation:

$$\frac{\partial \frac{1}{2} (z_k^T Q z_k + u_k^T R u_k)}{\partial u_k} + \left(\frac{\partial x_{k+1}}{\partial u_k} \right)^T \frac{\partial J^*(x_{k+1}, k + 1)}{\partial x_{k+1}} = 0
 \tag{23}$$

Then, the optimal control input u_k^* is

$$u^*(k) = -R^{-1} B_{dk}^T \frac{\partial J^*(x_{k+1}, k + 1)}{\partial x_{k+1}}
 \tag{24}$$

For acquire the optimal control law, the solution of $J^*(x_k, k)$ is the key point. We guess the a solution such that formula (23) is satisfied, hence, $J^*(x_k, k)$ is proposed as

$$\begin{aligned}
 J^*(x_k, k) &= \frac{1}{2}x_k^T P_k x_k - s_k^T x_k + \varphi_k \\
 J^*(x_{k+1}, k+1) &= \frac{1}{2}x_{k+1}^T P_{k+1} x_{k+1} - s_{k+1}^T x_{k+1} + \varphi_{k+1}
 \end{aligned}
 \tag{25}$$

thus

$$\frac{\partial J^*(x_{k+1}, k+1)}{\partial x_{k+1}} = P_{k+1} x_{k+1} - s_{k+1}
 \tag{26}$$

where P is a positive-definite symmetric matrix; z is the vector that will be discussed in next analysis; φ is a scalar.

From formula (21), (24)–(26), the (27) is obtained. And also formula (17a), (17b), (18) and (19) can be derived.

$$\begin{aligned}
 0 &= \frac{1}{2}r^T(k)Qr(k) - r^T(k)QC_d(k)x_k + \frac{1}{2}x_k^T C_d(k)QC_d(k)x_k + \frac{1}{2}x_k^T A_d(k)^T P_{k+1} A_d(k)x_k \\
 &+ \varphi_{k+1} - \varphi_k - \frac{1}{2}x_k^T P_k x_k + s_k^T x_k - s_{k+1}^T A_d(k)x_k \\
 &- \frac{1}{2}s_{k+1}^T B_d(k)(R + B_d^T(k)P_{k+1}B_d^T(k))^{-1}s_{k+1} \\
 &- \frac{1}{2}x_k^T A_d(k)^T P_{k+1} B_d(k)(R + B_d^T(k)P_{k+1}B_d^T(k))^{-1}B_d^T(k)P_{k+1}A_d(k)x_k \\
 &+ \frac{1}{2}x_k^T A_d(k)^T P_{k+1} B_d(k)(R + B_d^T(k)P_{k+1}B_d^T(k))^{-1}s_{k+1} \\
 &+ \frac{1}{2}s_{k+1}^T B_d(k)(R + B_d^T(k)P_{k+1}B_d^T(k))^{-1}B_d^T(k)P_{k+1}A_d(k)x_k
 \end{aligned}
 \tag{27}$$

Equation (18) is named by the Discrete-time State Dependent Difference Riccati Equation (DSDDRE). The vector s is a feed-forward, which can be acquired by Difference Eq. (19).

Remark 4 In infinite-time, the Eq. (18) reduces to frozen Riccati equation. When the state becomes stationary, the frozen Riccati equation is equal to the ordinary equation Riccati equation around the stationary point. The control method based on DSDRE strategy is regarded as optimal. So, the method proposed in this paper is suboptimal control methods.

3.2 Satiability of Close-Loop System

Form (17a) and (17b), the close-loop system of (11) can be written as

$$\begin{aligned}
x_{k+1} &= A_{clk}x_k + B_{dk}(R + B_{dk}^T P_{k+1} B_{dk})^{-1} s_k, \\
A_{clk} &= A_{dk} - B_{dk}(R + B_{dk}^T P_{k+1} B_{dk})^{-1} B_{dk}^T P_{k+1} A_{dk} \\
&= A_{dk} - F_k A_{dk}.
\end{aligned} \tag{28}$$

It is indicated that the system is consist of a feedback part and a feed-forward part from Eq. (28). Because the feed-forward part does not influence the matrix, the satiability of the close-loop system (28) only depends on the feedback part.

The feedback part of close-loop system is

$$x_{k+1} = A_{CLK}x_k \tag{29}$$

Now let's define the Lyapunov function

$$L(x_k) = x_k^T P_k x_k, \tag{30}$$

where $L(x_k) > 0 \forall x_k \neq 0$ and $L(x_k) = 0$ for $x_k = 0$. Based on formula (28), $L(x_{k+1})$ is going to be

$$L(x_{k+1}) = x_k^T A_{CLK}^T P_{k+1} A_{CLK} x_k \tag{31}$$

Then,

$$\begin{aligned}
L(x_{k+1}) - L(x_k) &= x_k^T A_{CLK}^T P_{k+1} A_{CLK} x_k - x_k^T P_k x_k \\
&= -x_k^T A_{dk}^T P_{k+1} F_k A_{dk} x_k + x_k^T A_{dk}^T F_k^T P_{k+1} F_k A_{dk} x_k - x_k^T C_{dk}^T Q C_{dk} x_k \\
&\leq -x_k^T [A_{dk}^T (I - F_k^T) P_{k+1} F_k A_{dk}] x_k.
\end{aligned} \tag{32}$$

From Eq. (28),

$$F_k = B_{dk}(R + B_{dk}^T P_{k+1} B_{dk})^{-1} B_{dk}^T P_{k+1} \tag{33}$$

Because R is positive definite matrix, $F_k^T = F_k > 0$

$$I - F_k^T > 0, \tag{34}$$

and

$$L(x_{k+1}) - L(x_k) < 0. \tag{35}$$

The Eq. (35) indicated that $L(x_k)$ is decreasing every time index k .

From (25), (26), (27) and (35), the system (11) is asymptotic stability, which means that $\lim_{t \rightarrow \infty} z = 0$. The trajectory tracking error e_k is going to be

$$\lim_{t \rightarrow \infty} e_k + \lambda e_k = 0. \tag{36}$$

From (36), trajectory tracking error e_k is exponential convergence to zero. The Theorem 1 is proved via Eqs. (25), (26), (27) and (36).

4 Numerical Simulations

For verifying Efficiency of the method, a numerical simulation is presented using two-link plane space robot model (Fig. 1).

Referring to the literature [19], the space robot model is proposed as

$$\begin{cases} H(q_b, q_m)\ddot{q}_m + C(q_b, q_m\dot{q}_b, \dot{q}_m)\dot{q}_m = \tau \\ \dot{y} = J^*(q_b, q_m)\dot{q}_m, \end{cases} \tag{37}$$

where $H(q_b, q_m)$ is the system inertia matrix; $C(q_b, q_m\dot{q}_b, \dot{q}_m)$ contains the nonlinear Coriolis centrifugal terms; q_b is space base angle; q_m is joint angle; and τ is input torque.

Let's define $x = q_m$, $\tau = u$, and assume that the space robot work in path-independent workspace(PIW) [19]. The system (37) can be rewritten as (38) using SDC.

$$\begin{cases} x_{k+1} = A_{dk}x_k + Bu_k \\ y_k^* = C_{dk}x_k, \end{cases} \tag{38}$$

Fig. 1 The sketch map of 2-link plane space robot

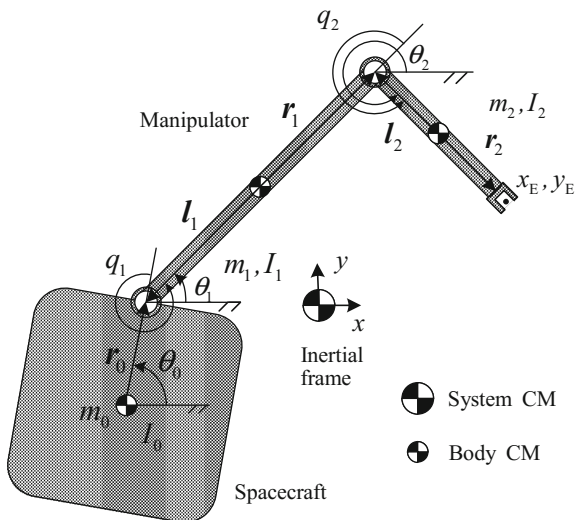


Table 1 The inertial parameters of model

Link	l_i (m)	r_i (m)	m_i (kg)	J_i (kg·m ²)
0	0.5	0.5	40	6.667
1	0.5	0.5	4	0.333
2	0.5	0.5	3	0.250

where $A_{dk} = -TH^{-1}C + I$, $B_{dk} = TH^{-1}$, $C_{dk} = J^*$.

The inertial parameters of space robot model are shown in Table 1.

The desired trajectory is

$$x = 0.28 \cos(0.1\pi t) + 0.85$$

$$y = 0.28 \cos(0.1\pi t) - 0.43.$$

The parameters of controller are choose as

$$\mathbf{Q} = \text{diag}(150, 100), \mathbf{R} = \text{diag}(0.1, 0.1), \lambda = \text{diag}(20, 20).$$

The sample period is 0.02 s. The simulate time is 40 s.

The simulation results are shown in Figs. 2, 3 and 4. Figure 2 is the tracking error of x and y; Fig. 3 is the control input; Fig. 4 is tracking result of x-y plant. From the Fig. 2, it indicates that the tracking error is convergence to neighborhood of zero at 5 s. The Fig. 4 indicates that the desired trajectory can be tracked using the sample-data controller. Using optimal control method, the tracking precision and input torque are taken the middle course, which is reflected in Figs. 2 and 3. Finally, the effectiveness of the method presented in this paper is verified by the numerical simulation.

Fig. 2 Tracking error

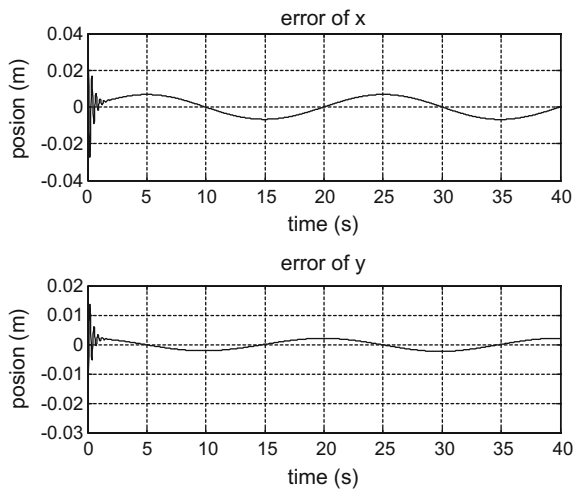


Fig. 3 Input control torque

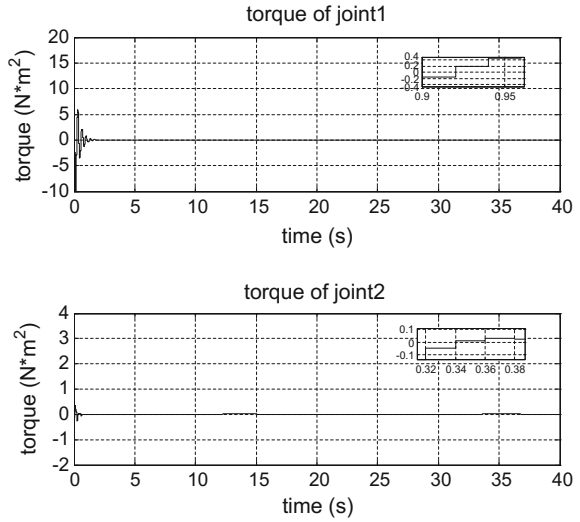
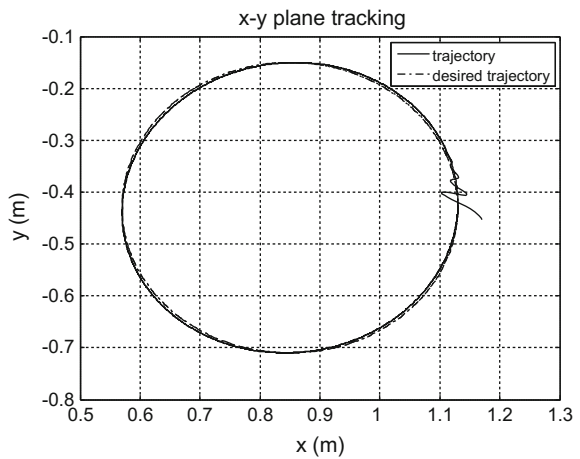


Fig. 4 The tracking result in x-y plane



Another key topic of sample-data control is how to choose simple period. Normally, sample-data control requires the simple period sufficient small in order to satisfy stability of system. Because of complexity of this problem, in this paper, we only give some brief result using simulation about sample period choosing. The result of simulation is show in Figs. 5, 6 and 7.

In the Fig. 5, the parameters of controller are not changed except simple period is increased to 0.06 s. In the Fig. 6, the simple period is 0.06 s, and the λ is increased to 35. In the Fig. 7, λ is decreased to 35, and simple period is increased to 0.1 s. From Figs. 5, 6 and 7, it is indicated that the upper bound of sample period is

Fig. 5 Tracking result in x-y plane with simple period is 0.06 s

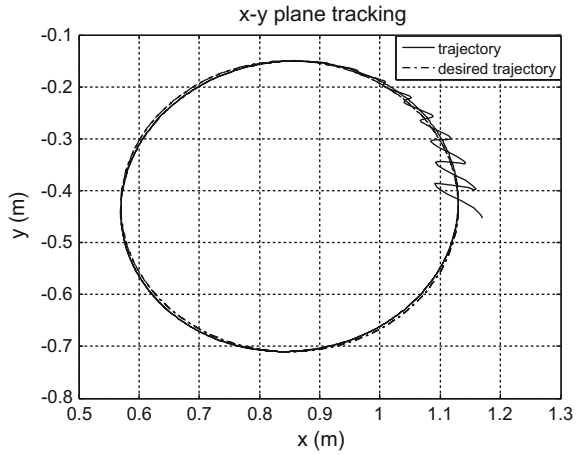


Fig. 6 Tracking result in x-y plane with simple period is 0.06 s, and $\lambda = \text{diag}(35, 35)$.

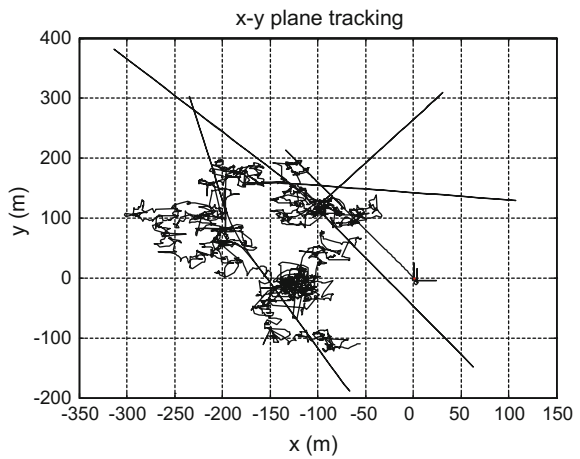
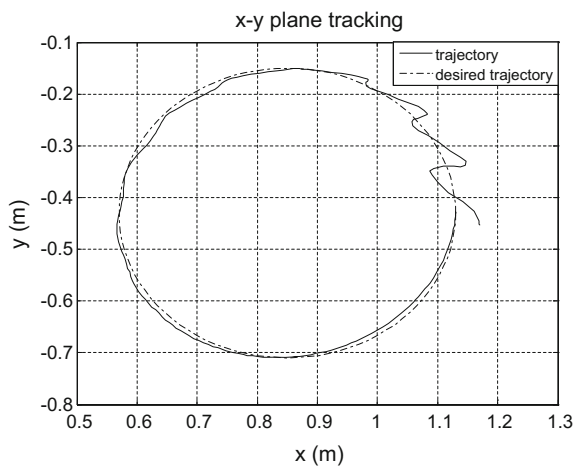


Fig. 7 Tracking result in x-y plane with simple period is 0.1 s, and $\lambda = \text{diag}(10, 10)$.



related to the sliding gain λ i.e. decrease λ can improve the upper bound of sample period, but the tracking precision will be reduction in low λ .

5 Conclusion

In this paper, we discuss optimal sample date control of a class of non-linear system. Based on DSDRE method, an optimal sample-data control method is proposed. The optimal and stable of close-loop system are investigated. The result of numerical simulation indicated using this method, the desired trajectory can be tracked by real trajectory and the tracking error and input torque can be compromised by adjusting the Q, R, λ . Finally, Using simulation we briefly investigate the relation of upper bound of sample period and the sliding gain λ .

References

1. Luo, W., Chu, Y., Ling, K.: Inverse optimal adaptive control for attitude tracking of spacecraft. *IEEE Trans. Autom. Control* **50**(11), 1639–1654 (2005)
2. Zhang, W., et al.: A control Lyapunov function approach to stabilization of affine nonlinear systems with bounded uncertain parameters. *Circ. Syst. Sig. Process.* **34**(1), 341–352 (2015)
3. Dupree, K., et al.: Asymptotic optimal control of uncertain nonlinear Euler–Lagrange systems. *Automatica* **47**(1), 99–107 (2011)
4. Patre, P.M., et al.: Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and RISE feedback control structure. *IEEE Trans. Autom. Control* **53**(9), 2180–2185 (2008)
5. Kamalapurkar, R., et al.: Approximate optimal trajectory tracking for continuous-time nonlinear systems. *Automatica* **51**, 40–48 (2015)
6. Cloutier, J.R., D’Souza, C.N., Mracek, C.P.: Nonlinear regulation and nonlinear H-infinity control via the state-dependent Riccati equation technique. I-Theory. In: 1st International Conference on Nonlinear Problems in Aviation and Aerospace, Daytona Beach, FL (1996)
7. Mracek, C.P., Cloutier, J.R.: Control designs for the nonlinear benchmark problem via the state-dependent Riccati equation method. *Int. J. Robust Nonlinear Control* **8**(4–5), 401–433 (1998)
8. Ornelas-Tellez, F., Rico, J.J., Ruiz-Cruz, R.: Optimal tracking for state-dependent coefficient factorized nonlinear systems. *Asian J. Control* **16**(3), 890–903 (2014)
9. Çimen, T.: State-dependent Riccati equation (SDRE) control: a survey. In: The International Federation of Automatic Control (2008)
10. Korayem, M.H., Nekoo, S.R.: Finite-time state-dependent Riccati equation for time-varying nonaffine systems: rigid and flexible joint manipulator control. *ISA Trans.* (2014)
11. Abdelrahman, M., Chang, I., Park, S.-Y.: Magnetic torque attitude control of a satellite using the state-dependent Riccati equation technique. *Int. J. Non-linear Mech.* **46**, 758–771 (2011)
12. Cimen, T.: Development and validation of a mathematical model for control of constrained non-linear oil tanker motion. *Math. Comput. Modell. Dyn. Syst.* **15**(1), 17–49 (2009)
13. Geranmehr, B., Nekoo, S.R.: Nonlinear suboptimal control of fully coupled non-affine six-DOF autonomous underwater vehicle using the state-dependent Riccati equation. *Ocean Eng.* **96**, 248–257 (2015)

14. Ahrens, J.H., Tan, X., Khalil, H.K.: Multirate sampled-data output feedback control with application to smart material actuated systems. *IEEE Trans. Autom. Control* **54**(11), 2518–2529 (2009)
15. Grüne, L., Worthmann, K., Nešić, D.: Continuous-time controller redesign for digital implementation: a trajectory based approach. *Automatica* **44**(1), 225–232 (2008)
16. Liu, X., Marquez, H.J., Lin, Y.: Input-to-state stabilization for nonlinear dual-rate sampled-data systems via approximate discrete-time model. *Automatica* **44**(12), 3157–3161 (2008)
17. Postoyan, R., Ahmed-Ali, T., Lamnabhi-Lagarrigue, F.: Robust backstepping for the Euler approximate model of sampled-data strict-feedback systems. *Automatica* **45**(9), 2164–2168 (2009)
18. Saeed, R., Behdad, G.: The state-dependent set-point regulation and tracking control of horizontal motion of AUV (2014)
19. Papadopoulos, E., Dubowsky, S.: On the nature of control algorithms for free-floating space manipulators. *IEEE Trans. Robot. Autom.* **7**(6) (1991)

Real-Time Flight Test Track Filtering and Association Using Kalman Filter and QDA Classifier

Kundong Wang and Yao Ge

Abstract In this paper an on-line track filtering and association algorithm for flight test was proposed. Firstly, a K-means clustering based scheme was used for track initialization and initial state and corresponding covariance matrix estimation for second-order extended Kalman filter. After that, track filtering and association and frequency estimation of Dutch roll were achieved through interactive use of the second-order extended Kalman-filter and the Kalman-predictor based QDA minimum error rate Bayesian classifier. Experimental results had shown that the algorithm can initialize the track reliably, filter and associate tracks in real time and estimate the frequency of flight testing precisely.

Keywords Real-Time flight • Filter • Kalman • QDS classifier

1 Introduction

Flight test is a key procedure in aeronautical engineering for the analysis of the aerodynamic characteristics of flights. Because the radar system could not provide precise observation during the initial stage of flight test process, track analysis is generally needed so as to attain a reasonable estimation of the initial tracks. The analysis is generally performed through the filtering and association of the observed tracks. Existing methods for track filtering and association can be classified into two major categories, namely off-line batch processing and on-line real-time processing. Off-line batch processing could provide precise estimation. However it's not suitable for real-time application due to its heavy computing load. On-line real-time processing could estimate the tracks efficiently. However it's only applicable in simple situations. So the design of a track filtering and association algorithm that

K. Wang (✉) · Y. Ge
Shanghai Jiao Tong University, Dongchuan Rd. 800, Shanghai 200240, China
e-mail: kdwang@sjtu.edu.cn

Y. Ge
e-mail: alexgecontrol@sjtu.edu.cn

could be used in real-time application while could also provide relatively precise estimation is very important for flight test application.

In this paper on-line track filtering and association algorithm for flight test was proposed. The algorithm uses a variation for K-means algorithm for track initialization. After initiation, the algorithm uses second-order Kalman filtering for track filtering and the Kalman predictor based QDA classifier for track association. The track filtering and association algorithms are used interactively for each newly arrived observation so as to accomplish real-time track filtering and association.

The paper is organized as follows. In Sect. 2 the K-means based track initialization scheme was presented. The second-order EKF for track filtering was derived in Sect. 3 with the focus on the derivation of continuous-time system equation for typical flight test motion and its discretization. The Kalman predictor based QDA classifier for track association was discussed in Sect. 4 with the focus on the approximation of posterior distribution parameters through Kalman predictor and the derivation of decision function for classifier using QDA. Experimental results on flight test data was presented in Sect. 5.

2 Track Initialization

Track initialization is, in essence, a real-time clustering problem. We chose K-means based scheme for this problem based on the following two considerations. Firstly, K-means algorithm could be implemented iteratively which means its computational workload is suitable for real-time application. Secondly, typical initial conditions of flight test will not introduce strong random disturbance for stable initialization of K-means algorithm.

Define k as the number of clusters, i as the cluster index, x^t as the observed coordinates of flights, Ω_i^t as the set of observations belong to the i th cluster at time t , μ_i^t as the cluster mean of the i th cluster at time t , d^t as the distance between the two cluster mean at time t . The K-means based scheme for track initialization is as follows.

- (1) Algorithm initialization. Set k according to the application scenario and $\mu_1^t, \mu_2^t, \dots, \mu_k^t$ as the first, second and the k th observations respectively. Take the arrival time of the $(k + 1)$ th observation as the initial time;
- (2) Clustering. Classify the newly arrived observation as follows:

$$x^t \in \Omega_i^t, i = \arg \min_j \{ \|x^t - \mu_j^t\|, 1 \leq j \leq k \} \quad (1)$$

- (3) Updation. Update the cluster means and the distance between the two cluster mean and the difference of the distance as follows.

$$\mu_i^{t+1} = \frac{1}{|\Omega_i^t|} \sum_{x \in \Omega_i^t} x, 1 \leq i \leq k \quad (2)$$

$$d^{t+1} = \|\mu_i^{t+1} - \mu_i^t\|, 1 \leq i \leq k \quad (3)$$

$$\Delta d^{t+1} = d^{t+1} - d^t \quad (4)$$

(4) Stopping condition. Stop the clustering process if Eq. (5) is satisfied.

$$\Delta d^{t+1} \leq \Delta d^t \quad (5)$$

The scheme for track initialization is attained through modification of stopping condition of traditional K-means algorithm. Considering in the typical flight test process, the tracks will intersect as the time goes on, the above-mentioned stopping condition was chosen so as to keep the effectiveness of the initialization and to start the filtering and association process as soon as possible.

3 Track Filtering

After track initialization, the track filtering and association process is performed as interactive use of filtering and association algorithms. In this section, we start the presentation of the process with the algorithm for track filtering.

3.1 Continuous-Time System Equations

Considering the Dutch roll mode is used in typical flight test, the continuous-time system equations were built as follows. The Dutch roll motion could be modeled as a combination of constant acceleration movement along the two axes and damp-free vibration along the other axis, Here we assume target flights move as damp-free vibration along the x axis and as constant acceleration along the y and z axes. Taking the typical working condition of radar system into consideration, we further assume that the derivative of acceleration is a white noise along all axes. So the reflection model [1] can be used for the Dutch roll flight test process. For motion along each axis, define system state as $s = [p, v, a]^T$ in which p, v, a are position, velocity and acceleration along the axis. Denote the natural frequency of the damp-free vibration as k, The damp-free vibration motion along x axis could be modeled as in (6).

$$\dot{s}_x = \begin{bmatrix} 0 & 1 & 0 \\ -k & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} s_x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w_x \tag{6}$$

And the constant acceleration motion along y and z axes could be further modeled as in (7).

$$\dot{s}_{y,z} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} s_{y,z} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w_{y,z} \tag{7}$$

Discretization of System Equations

According to the reflection model, the system equations for motions along y and z axes are as follows.

$$s_{y,z}^{n+1} = F_{y,z} s_{y,z}^n + w_{y,z}^n, F_{y,z} = \begin{bmatrix} 1 & T & T^2/2 \\ & 1 & T \\ & & 1 \end{bmatrix} \tag{8}$$

The covariance matrix for system noise along all three axes would be:

$$Q_{x,y,z} = \text{cov}(w_{x,y,z}^n) = S_{x,y,z} Q_0 \tag{9}$$

In which Q_0 is defined as:

$$Q_0 = \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix} \tag{10}$$

The system equation for motion along x axis could be discretized as follows. Integrate along one sample interval [0, T], we have:

$$s_x^{n+1} = F_x s_x^n + w_x^n \tag{11}$$

The system matrix F_x is defined as follows:

$$F_x = \begin{bmatrix} \cos(\sqrt{k}T) & \sin(\sqrt{k}T)/\sqrt{k} & 0 \\ -\sqrt{k} \sin(\sqrt{k}T) & \cos(\sqrt{k}T) & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{12}$$

The parameter k, which is the natural frequency of the motion, in (9) could be estimated through extending the system state and taking in parameter k as an extra parameter for estimation, which is further explained in later section.

3.2 Second Order EKF for Track Filtering

Due to the intrinsic non-linearity of the observation process, the second-order EKF was used for track filtering. In order to improve the adaptiveness of the algorithm, the system state was expanded and the natural frequency of the Dutch roll motion was taken in so that it could be estimated during the filtering process.

The system equations of the second order EKF could be derived as follows. Define the system state as:

$$x^n = \left[s_x^{nT} \quad s_y^{nT} \quad s_z^{nT} \quad \sqrt{k^n} \right]^T \tag{13}$$

The system matrix would be:

$$A = \text{diag}(F_x \quad F_y \quad F_z \quad 1) \tag{14}$$

The system noise would be:

$$w^n = \left[w_x^{nT}, w_y^{nT}, w_z^{nT}, w_{\sqrt{k}}^n \right]^T \tag{15}$$

The covariance matrix for system noise would be:

$$Q = \text{diag}(Q_x, Q_y, Q_z, Q_{\sqrt{k}}) \tag{16}$$

Based on the above-mentioned definitions, we could have the system equations for the EKF as:

$$x^{n+1} = Ax^n + w^n \tag{17}$$

The observation equations for the EKF could be derived from the radar observation process. Since the output of the radar system is given using polar coordinates, the observation equations could be attained from the transformation of standard coordinates into polar coordinates. Define system output as:

$$y^n = [r^n, \theta^n, \varphi^n]^T \tag{18}$$

The observation equations of the EKF would be:

$$y^n = h(x^n) + v^n \tag{19}$$

In which $h(x)$ is the transform between standard coordinates and polar coordinates and could be defined using system state as:

$$h(x) = \begin{bmatrix} \sqrt{x_1^2 + x_4^2 + x_6^2} \\ \tan^{-1}\left(\frac{x_4}{x_1}\right) \\ \tan^{-1}\left(\frac{x_7}{\sqrt{x_1^2 + x_4^2}}\right) \end{bmatrix} \quad (20)$$

The covariance matrix for observation noise is

$$R = \text{cov}(v^n) \quad (21)$$

The above-mentioned matrix R could be attained from the parameters of the radar system.

Last but not least, the initial estimation for system state and its covariance matrix could be attained from the clustered tracks of Sect. 2.

Based on the above definitions, the standard second-order EKF could be applied for track filtering process.

4 Track Association

The estimations of the EKF, namely the state mean and state covariance, could be used as the defining parameters for true state distribution if we approximate the true state distribution as multivariate Gaussian. Based on this approximation, we chose to build a QDA based minimum error rate classifier for track association using the estimations of the EKF as the basis of algorithm parameters. In order to meet the requirements of the QDA algorithm, a Kalman predictor was first applied to derive the posterior distribution of state distribution based on current state estimation provided by the EKF, then the QDA classifier was applied to accomplish the track association process.

4.1 Kalman Predictor for Posterior Approximation

The one-step forward Kalman predictor was used to approximate the posterior distribution needed by the QDA classifier. Since we cannot know the class label of the newly arrived observation, here Kalman predictors were applied to both track's last observation. Define $x^{n|n}$ and $P^{n|n}$ as the state mean and covariance matrix estimations provided by the EKF, $x^{n+1|n}$ and $P^{n+1|n}$ as the state mean and covariance matrix predictions provided by the Kalman predictor for state distribution approximation, The Kalman predictor could be defined as follows [2]:

$$x^{n+1|n} = Ax^{n|n} \tag{22}$$

$$P^{n+1|n} = AP^{n|n}A^T + Q \tag{23}$$

The system matrix A in Eqs. (21) and (22) is the same as in Eq. (14). The system noise matrix in the above two equations is the same as in Eq. (16).

4.2 QDA-Based Classifier for Track Association

According to the above-mentioned approximation, the distribution of current system state is:

$$x^{n+1} \sim N(x^{n+1|n}, P^{n+1|n}) \tag{24}$$

In which:

$$N(x^{n+1}|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x^{n+1} - \mu)^T \Sigma^{-1}(x^{n+1} - \mu)\right] \tag{25}$$

The minimum error rate classifier using QDA is defined as:

$$\delta_k(x) = -\frac{1}{2} \ln|\Sigma| - \frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu) + \ln \pi_k, k = 1, 2 \tag{26}$$

The π_k in the above equation is the prior probability of the kth track being observed. For flight test process we chose $\pi_1 = \pi_2 = 0.5$. Define the true class of the newly arrived observation as $k^*|x_n + 1$. The decision rule of the classifier for track association would be [3]:

$$k^*|x^{n+1} = \max_k \arg\{\delta_k(x^{n+1})\} \tag{27}$$

For every newly arrived observation, first apply Kalman predictor to approximate the required posterior distributions, then apply QDA-based classifier to finish track association process. Through the interactive use of filtering and association algorithms, the track of flight test could be filtered and associated in real-time. The whole algorithm is summarized in the following section.

5 Algorithm Sum-Up

Based on the discussions of Sects. 2, 3 and 4, the whole algorithm for on-line track filtering and association could be summarized as follows:

- (1) Track initialization. Apply the variation of K-means algorithm discussed in Sect. 2 until the stopping condition is met for track initialization;
- (2) Second-order EKF initialization. Apply the standard maximum likelihood estimators for multivariate Gaussian mean and covariance matrix to estimate the initial state and its covariance matrix for the EKF used in Sect. 3 for track filtering;
- (3) Real-time track filtering and association. After initialization of track and the second-order EKF, apply the following procedure to the newly arrived observation:
 - (a) Apply Kalman predictor to each associated tracks to predict the one-step forward state mean and covariance matrix so as to approximate the true distributions the newly-arrived observation;
 - (b) Calculate the decision functions for each potential track using the parameters provided by step 3a and classify the newly arrived observation using minimum-error rate principle as in (27) so as to associate the track;
 - (c) Update the newly associated track using the second-order EKF discussed in Sect. 3 so as to estimate the state mean and covariance matrix of the given track at current moment.
- (4) Repeat the procedure described in step 3 till the end of the estimation.

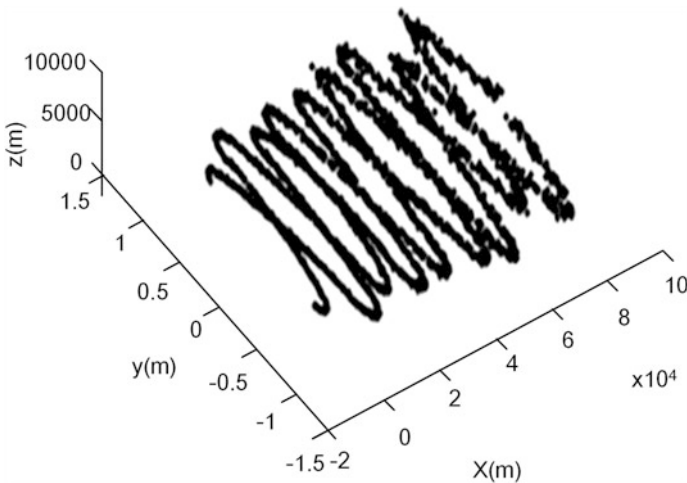


Fig. 1 Raw tracks for flight test with two flights

6 Experiments

The proposed algorithm was tested on the data collected during one flight test with two flights. The raw tracks are as follows (Fig. 1).

6.1 Track Initialization Test

The K-means based scheme initialized the two tracks as follows. Define N as the number of observation contained in each track, the temporary tracks for $N = 30$,

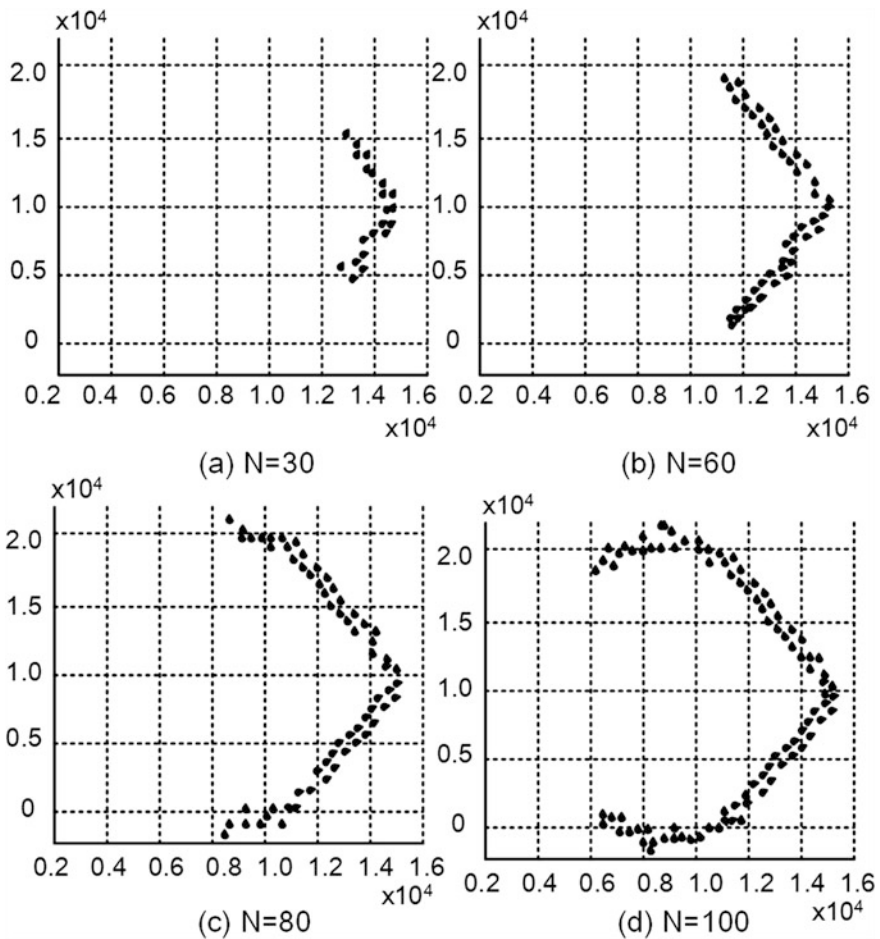


Fig. 2 Track initialization test

60, 80 and 100 are shown in Fig. 2. One track is marked with triangles and another is marked with crosses. The stopping condition was met when $N = 112$. The experimental results showed that the proposed algorithm could effectively initialize the tracks.

6.2 Track Filtering and Association Test

The filtered and associated tracks are showed in Fig. 3. Compared with traditional track filtering and association methods, the proposed method can stably handle the case when there is only one track observation and associate the track correctly under this circumstances.

6.3 Natural Frequency Estimation Test

The results for natural frequency estimation of the two Dutch roll motions are as follows. The two curves showed that the estimation could converge to the true value, which was 0.028 Hz for testing data, after 800 iterations. The estimation of the proposed algorithm has low variance, which is suitable for flight test application. Experimental results on flight test data showed that the proposed algorithm could initialize the tracks stably and accomplish the on-line filtering and association precisely (Fig. 4).

Fig. 3 Real-time track filtering and association test

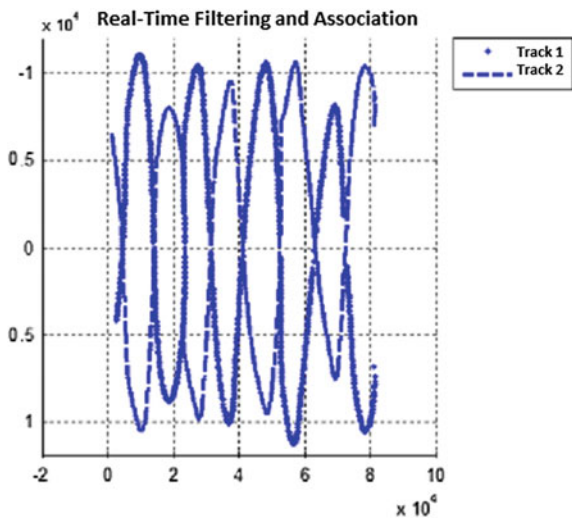
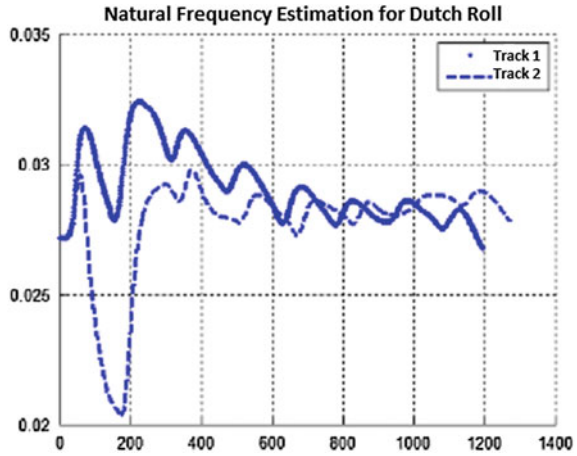


Fig. 4 Natural frequency estimation test



7 Conclusion

In this paper the design and implementation of an on-line track filtering and association algorithm for flight test was discussed. The presentation focuses on the variation of K-means algorithm for track initialization, the second-order EKF for track filtering and the Kalman predictor based QDA classifier for track association.

References

1. Li, X.R., Jilkov, V.P.: Survey of maneuvering target tracking, Part I: dynamic models. *IEEE Trans. Aerosp. Electron. Syst.* **39**, 1333–1364 (2003)
2. Deng, Z.L.: *Modeling and Estimation*. Science Press, Beijing (2008)
3. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*. Springer, New York (2013)

Simulation of Time Delay Compensation Controller for a Mobile Robot Using the SMC and Smith Predictor

Dong-Hyuk Lee, Jae-Hun Jung, Ha-Neul Yoon, Young-Sik Park
and Jang-Myung Lee

Abstract This paper uses a Smith predictor or time delay compensation in remote control system of mobile robot. Time delay causes a few errors, and the error in turns influence the control of the target. Smith predictor requires an accurate model, the type of a virtual model with a time delay in the feedback part, to compensate for the error value for the time delay. In the experiment of this paper, the error of Smith predictor is compensated by using a sliding mode controller. The mobile robot is implemented through the simulation that imaginary modeling is composed of two drive-wheels and one fixed-wheel. The mobile robot is controlled by sliding mode controller.

Keywords SMC · Smith predictor · Mobile robot · Time delay compensation controller

1 Introduction

Industrial manufacturing robots, which are the parts of an erstwhile robot industry, are developed mainly in order to improve the life quality of people. However, researches of robot industry are progressed not only for manufacturing robot filed

D.-H. Lee · J.-H. Jung · H.-N. Yoon · Y.-S. Park · J.-M. Lee (✉)
Pusan National University, 2, Busandaehak-ro 63beon-gil, Geumjeong-gu,
Busan 46241, Republic of Korea
e-mail: jmlee@pusan.ac.kr

D.-H. Lee
e-mail: ldh0917@pusan.ac.kr

J.-H. Jung
e-mail: jaehoon1696@pusan.ac.kr

H.-N. Yoon
e-mail: haneul1696@pusan.ac.kr

Y.-S. Park
e-mail: youngsik1696@pusan.ac.kr

but also for various categories such as service robot and medical robot in modern society. Special field application, such as discovery robot, a mobile robot which operates in poor environment or a rescue robot does researches. However these robots need not only quick and accuracy data provision, but also accurate remote control. Thus, development of remote control field which transfer surrounding environment and situation of robot to perform correct acting of mobile robot using data from sensors which equipped in mobile robot is becoming more and more necessary [1].

The target controlled mobile robot in this paper is commonly used in industry field or discovery. Although mobile robots are controlled by using robot-loaded sensors in real time by a distant operator, a variety of complement are required.

Time delay is a one part of problems in remote control system. While communication, time-delay is logically occurred, problems which come from time-delay transfer error data in controlling a robot. If time-delay is not compensated, system will be unstable as a little communication-delay in any remote environment [2].

In this paper, we use Smith predictor to compensate for time delay. Smith predictor compensates a time-delay elements between controller and plant to obtain stability, so we use Smith predictor to compensate a time-delay problem which is obligatorily shown in time-delay controller [3].

Therefore, in this paper we compensate a randomly given time-delay using a Smith predictor, and verify a remote control of mobile robot which has time-delay, through the simulation using sliding mode controller.

2 Designing of Mobile Robot

2.1 WMR (*Wheeled Mobile Robot*)

The diagram of mobile robot which has two wheels and one stabilizer is shown in Fig. 1.

To realize a mobile robot in simulation such as Fig. 1, we use initial value and wrote in Table 1.

In this paper, the control of WMR uses a v , θ (each variable means velocity and angle value), so parameters of mobile robot are represented shown as Table 1.

Also, WMR has 3 DOF.

$$q = [x \quad y \quad \theta]^T \quad (1)$$

This Steering system of WMR is same as below.

Fig. 1 Outline of mobile robot

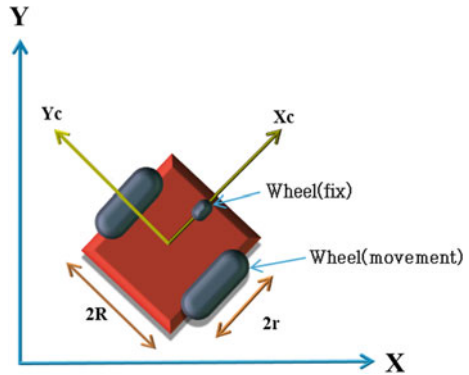


Table 1 WMR parameter

m	Mass of WMR
I	Inertia moment of WMR
R	Distance between two-wheels
r	Radius of wheel
O, X, Y	Cartesian coordinate
C, X_C, Y_C	(WMR criterion) Cartesian coordinate
x, y	Center of WMR in Cartesian coordinate
θ_m	Angle between X and X_C (direction of mobile robot progress)

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = J(\theta)v \tag{2}$$

Therefore modeling of WMR which referred in this paper can control a straight-line motion and a turning movement of mobile robot by using a velocity and angle rate of mobile robot.

2.2 Sliding Mode Control

The sliding mode controller is a variable structure control system, which change the structure of the controller to allow initial value of system state variable access to the sliding plane which lies the target state variable.

Designing sliding surface means to move the state variables of the system in a desired plane, and the plane is depend on the error between the reference input and the output, the plane can be designed suitably by designer [4].

Sliding mode controller is constructed by Ref. [5] and using tracking error of WMR with motion equation and Lyapunov equation, Eq. 3 can be identified when $\epsilon > 0$.

$$\tau = \tau_{eq} + \tau_s = E^{-1} \left[\dot{v}_c(t) + \beta e_c(t) + k \frac{S}{|S| + \epsilon} \right] \tag{3}$$

2.3 Smith Predictor

The Smith predictor is used to solve the problem which caused by time-delay between controller and plant, and compensate an instability which is generated by time delay factor, in such a way that feedback is got by combining an estimation equation of plant transfer function G with a time delay value [6, 7].

If the basic model of Fig. 2 to be mathematical formula, the modeling equation is shown as below.

$$\left(X - e^{-sT} Y - \left(\hat{G}w - e^{-sT} \hat{G}w \right) \right) e^{-s\frac{T}{2}} CG = Y \tag{4}$$

$$e^{-s\frac{T}{2}} CGX - e^{-sT} CGY - e^{-s\frac{T}{2}} CG\hat{G}w + e^{-s\frac{3T}{2}} CG\hat{G}w = Y \tag{5}$$

To sum up the expression for an output w

$$w = \frac{Y}{e^{-s\frac{T}{2}} G} \tag{6}$$

When an Eq. 6 substitute to Eq. 5

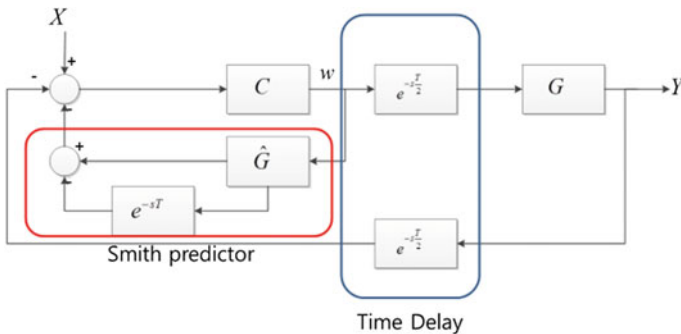


Fig. 2 Basic model of Smith predictor

$$\frac{Y}{X} = \frac{e^{-s\frac{T}{2}}CG}{1 + e^{-sT}CG - CG + e^{-sT}CG} = \frac{CG}{1 + CG} \cdot e^{-s\frac{T}{2}} \tag{7}$$

Uncertainty factors of control are cancelled out and delay-time factors are leaving, due to the Smith predictor as shown in Eq. 7.

3 Experiment

We generate a time delay and compensate through the smith predictor to progress the experiment. In Fig. 3, we can see the results of the controller that designed in this paper for the original input.

Ideally, it does not matter that plant modeling and smith compensator modeling are same. So we experiment by adjusting the time delay.

Figure 3 shows the simulation model. The blue border region is the part that randomly inserted time delay and red border region is the part that estimates the time delay using smith compensator. Each part of the experimental results is input, the case that removes the smith predictor and only gave the time delay, the case that adds the smith predictor.

Figure 4 shows the case that random time delay and time delay going through the smith predictor are each given 0.01 s and 0.02 s.

Figure 5 shows the resulting graph when random time delay and time delay going through the smith predictor are each given 0.05 s and 0.1 s.

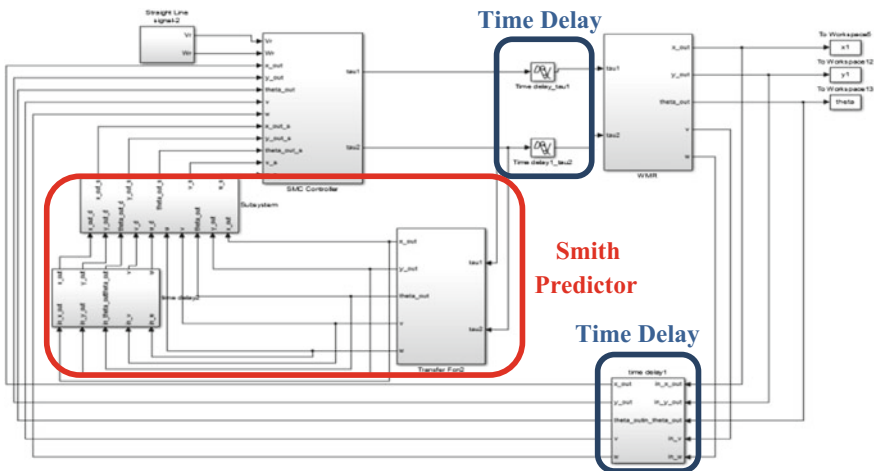


Fig. 3 The actual implementation of the simulation model

Fig. 4 Experimental results

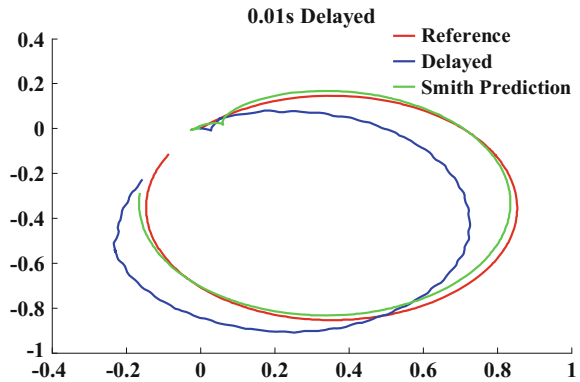
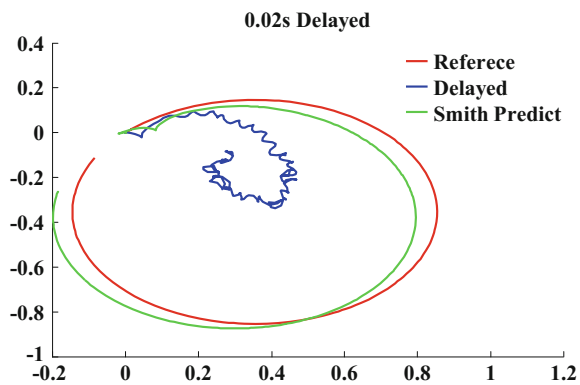


Fig. 5 Experimental results

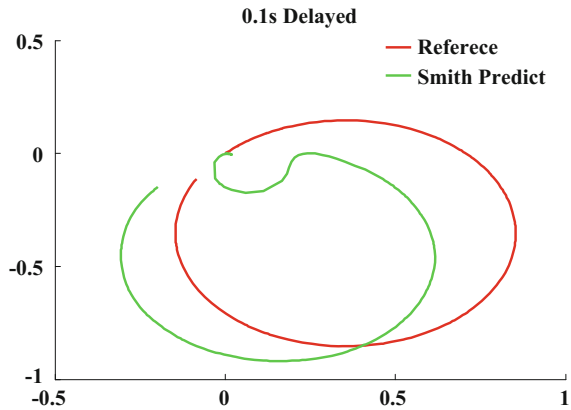


According to Fig. 5, because of the data from controller to plant is being delayed, the data error of the feedback system is increasing. If the delay value runs over 0.02 s, it will be much harder to compare the delay value.

Therefore, like Fig. 6, we checked the performance of the compensator while increasing the delay value except the delayed value. Figure 6 shows the result of the performance test of the smith compensator for the 0.1 s delay.

Figure 6 shows the experimental results. At the first few seconds data represents an invalid value and after that it will follows close to the input through the compensator.

By using a Smith predictor as shown in Figs. 4, 5 and 6, it can be seen that correcting the error caused the time delay, it is possible to make stably control with only time delay value.

Fig. 6 Experimental results

4 Conclusion

This paper shows a compensation for time-delay of WMR remote control through the simulation that the sliding mode controller with Smith predictor is applied. The more time delaying, the higher error rates. So, the researches which complement many kind situation and act in real experiment are needed. As the compensation does not consider several conditions in real system. It can be seen that a method for re handling is required for the signal after the Smith predictor, through the other controller, instead of using only Smith predictor.

Acknowledgements This research was financially supported by the Ministry of trade, Industry and Energy(MOTIE), Korea Institute for Advancement of Technology(KIAT) through the Robot Business Belt Development Project(A012000009).

This research was supported by the MOTIE (Ministry of Trade, Industry & Energy), Korea, under the Industry Convergence Liaison Robotics Creative Graduates Education Program supervised by the KIAT (N0001126).

References

1. Kweon, Y.-T., Kim, M.-K., Kang, H.-J., Roh, Y.-S.: Remote control of a mobile robot using haptic device. In: Korean Society of Precision Engineering Autumn Conference, pp. 120–124 (2004)
2. Yokokohji, Y., Imaida, T., Yoshikawa, T.: Bilateral control with energy balance monitoring under time-varying communication delay. In: IEEE Conference on Robotics and Automations, pp. 2684–2689 (2000)
3. Jung, S.: Study on smith predictor design for a system with a time-delayed controller. *J. Control Autom. Syst.* 146–148 (2014)
4. Camacho, O.E., Smith, C.A.: Sliding mode control: an approach to regulate nonlinear chemical process. *ISA Trans.* **39**, 205–218 (2000)

5. Yang, J.M., Kim, J.H.: Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *IEEE Trans. Robot. Autom.* **15**(3) (1999)
6. Choi, H.J., Jung, S.: Design of a time-delay compensator using neural network in a tele-operation system. *J. Control Autom. Syst.* **21**(4), 449–455 (2011)
7. Smith, A.C., Hashtrudi-Zaad, K.: Smith predictor type control architectures for time delayed teleoperation. *Int. J. Robot. Res.* **25**(8), 797–818 (2006)

Differential Wheeled Robot Navigation Based on the Smoothing A* Algorithm and Dead-Reckoning

Daowei Jiang and Liang Yuan

Abstract This paper aims to solve synthesis problems of two-wheel difference speed mobile robot in navigation and localization. A new method is proposed by combining the A* algorithm, motion control, and dead reckoning before the integrated treatment to address practical problems of short-term navigation and orientation during robot movement. First, the A* algorithm is used to plan out robot's trajectory from a known point to the target, Floyd algorithm and circular arc algorithm are chosen for the optimization of the A* algorithm to make it in accordance with the trajectory of two-wheel differential motion robot. Then, the robot is controlled to move along the scheduled trajectory. Finally, based on dead-reckoning, robot's position in the environment during locomotion is calculated to realize real-time positioning and tracking of its position and trajectory.

Keywords Positioning and navigation • The A* algorithm • Floyd algorithm • Motion control • Combination calculation of dead-reckoning

1 Introduction

In the research of autonomous mobile robots, the core lies in navigation technology which means that mobile robots perceive the surroundings and their own states through sensors to achieve self-movement in the environment with roadblocks. Navigation is mainly used to solve the following four problems: (1) Treatment of the information obtained and establishing the model of environment by a certain algorithm; (2) Finding an optimal or approximate optimal collision-free path which costs robot the minimum energy, time or distance from the starting point to the

D. Jiang (✉)

College of Mechanical Engineering, Xinxiang University, 1230, Ürümqi, China
e-mail: 6021847542@qq.com

L. Yuan

Xinxiang University, 1230, Ürümqi, China
e-mail: 123514045@qq.com

target; (3) Making robot move from the starting point to the target in accordance with the scheduled path; (4) Localization of robot by a certain means of detection to obtain its position and direction in space and the information in environment.

Path planning in the navigation of mobile robot is a critical problem. In recent years, Rapidly-Exploring random trees (RRT) algorithm [1] is more widely used, which is a single query path-planning method based on sampling. It guides the searching area to empty places through random sampling, so that modeling of space is avoided, which has unique advantages compared with other methods. But after the search of RRT for the connecting path, the path generated may be longer or with some apparent corners, which results in non-smooth path due to its randomness.

Voronoi diagram method, method of silhouette, and PRM (Probabilistic Roadmap Method) all belong to roadmap method. The PRM method is in more common use at present [2] Based on probability theory, map is established through random sampling, before the effective searching of map and feasible path of mobile robot. However, sample nodes are entirely based on randomly sampling technology on the construction of roadmap, making path search tend to be more random in nature, which easily results in the fact that searching path is often not the most optimal.

A* (A-Star) algorithm is a direct searching method with the shortest path and the most effectiveness in static grid map [3]. D* algorithm, as the improved version of A* algorithm, is the shortest path searching method in a locally known or unknown environment [4]. However, D* and A* algorithm are all on the basis of grid map, which is not equal to the shortest path in a real environment. Moreover, the shortest path based on grid map does not conform to the moving route of robot.

As the shortest path problem cannot be solved by the A* and D* algorithms in actual conditions, Anthony Stentz proposed the Field D* method based on interpolation method [5], which could bring more smooth path. However, as interpolation method is applied, it is equivalent to the rising number of the grid units, which leads to greatly increasing of calculation.

Another key problem in mobile robot navigation is positioning, which can be divided into two categories according to the type of sensors used, namely, absolute positioning and relative positioning. Absolute positioning mainly adopts navigation beacon, map matching, or satellite navigation technology while relative positioning is to measure relevant position and direction with respect to the original location to determine the current position. Dead-reckoning method based on encoder is commonly used in all relative positioning methods [6]. Dead reckoning based on encoder has the characteristics of realizing high-precision positioning within a local range in a short period of time.

Dead-reckoning method based on double differential drive is to establish its kinematics differential equation, before calculating numerical iterative calculation equation (hereinafter referred to as integral calculation) [7, 8], for the purpose of realizing the tracking and orientation of robot. The essence of the integral calculation is to use a short straight line instead of a small trajectory of robot. There are significant errors in integral calculation with limited frequency and precision of sampling. And integral calculation is used to estimate the next position with the

given current speed and position, which just answer the question of “where am I”, without solving the question of “how should I reach the goal”.

In line with the principle that robot could walk and the path is the shortest, a smooth schedule path is obtained through the Floyd algorithm and smooth optimization when the environment information is known. With a smooth scheduled path, the driving speeds of the left and right wheels are designed, so that the two-wheel robot would run according to the planned route. For the known smooth path planning problems, the dead-reckoning method is improved to the combination calculation through arc and linear, and the calculation of the speed and position at the next time is based on the current speed and position, for the purpose of realizing real-time localization and tracking of robot.

2 Smooth A* Algorithm

2.1 A* Algorithm

The expression of A* algorithm:

$$f'(n) = g'(n) + h'(n)$$

$f'(n)$, $g'(n)$, and $h'(n)$ are all estimates, representing estimation from the test point to the target, the cost from starting point to the current point and the cost valuation from the current point to the target, also called the heuristic value. $f(n)$, $g(n)$, and $h(n)$ are the actual value, and the optimal path could be found only when the $h'(n) \leq h(n)$.

If the A* algorithm is obtained by two kinds of heuristic function: A1 and A2, when $h'_1(n) < h'_2(n)$ in any target control points, it could be said that the A2 algorithm is more informed, that is the extension points of A2 are less than those of A1. As a result, the search of A2 is more efficient. The heuristic function of an optimal algorithm is $h' \equiv h$, which is used to search the minimum area and the optimal solution. However, the cost of f' is higher, the number of exploration is needed more, and thus the greater amount of calculation. The choosing of the heuristic function is the middle value between the exact function and the computational cost.

In this article, the heuristic function adopted is a diagonal function. The cost of a grid with horizontal or vertical movement is D , then the cost of a grid with diagonal movement is $D_2 = \sqrt{2}D$. Thus

$$\begin{aligned} h_d(n) &= \min(\text{abs}(n \cdot x - \text{goal} \cdot x), \text{abs}(n \cdot y - \text{goal} \cdot y)) \\ h_s(n) &= (\text{abs}(n \cdot x - \text{goal} \cdot x) + \text{abs}(n \cdot y - \text{goal} \cdot y)) \\ h'(n) &= D * h_d(n) + D * (h_s(n) - 2 * h_d(n)) \end{aligned}$$

Fig. 1 The diagonal function

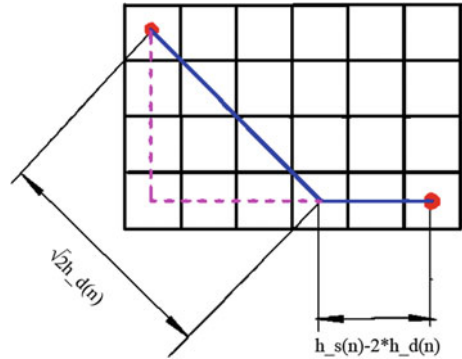
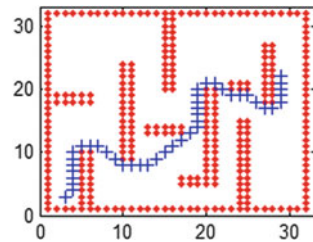


Fig. 2 The trajectory of the A* algorithm



As shown in Fig. 1.

In the case of known raster map, the planning path by A* algorithm as shown in Fig. 2, in which red represents obstacles, while blue shows the walking route.

2.2 Optimization of the A* Algorithm by the Floyd Algorithm

Because the A* algorithm is a representation method based on grid map, inflection points are more in the route. Optimization is adopted by the Floyd algorithm to remove excess points and inflexion points, making it more in line with the concept of minimum path.

Combination of Nodes with collinear. There are three points, namely, A (1, 1), B (2, 2), and C (3, 3). If the difference values of abscissa and vertical scale between A and B is equal to those between B and C, then it could be said that A, B, and C are with three points collinear, thus the point, B is deleted. After data processing, the route in Fig. 2 is shown in Fig. 3.

Floyd algorithm optimization. Floyd algorithm is used to find the shortest path between the source points in the given weighted graph. The purpose is to remove excess inflection points, whose principle is shown in Fig. 4.

Fig. 3 The roadmap after the removal of linear image

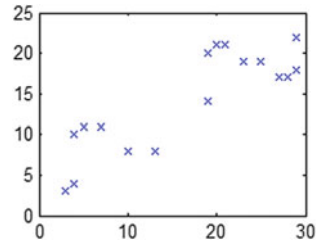


Fig. 4 Floyd algorithm diagram

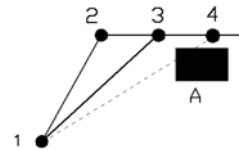
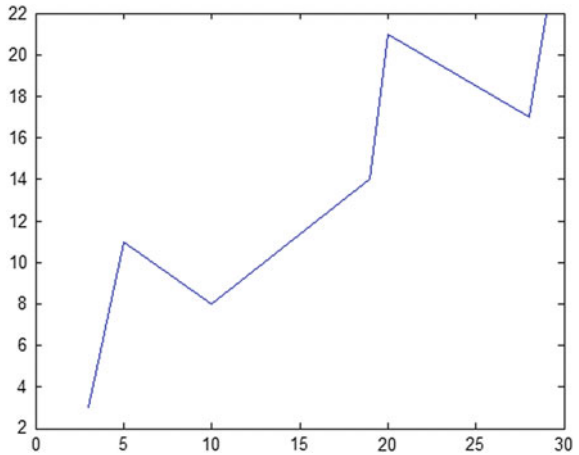


Fig. 5 The route after the optimization by Floyd



$D(1, 4)$ is the shortest distance from Point 1 to Point 4. Figure 5 shows that the initial value of $D(1, 4) = +\infty$, Route $R(1, 4) = 1 \rightarrow 4$

Insert Point 2 between Points 1 and 4

Insert Point 3 between Points 1 and 4

If $D(1,2) + D(2,4) < D(1,4)$

Then $D(1,4) = D(1,2) + D(2,4)$

$R(1,4) = 1 \rightarrow 2 \rightarrow 4$

Insert the point 3 between points 1 and 4

if $D(1,3) + D(3,4) < D(1,2) + D(2,4)$

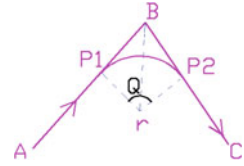
then $D(1,4) = D(1,3) + D(3,4)$

$R(1,4) = 1 \rightarrow 3 \rightarrow 4$

Finally, Point 2 is deleted.

After the optimization of Fig. 4 by Floyd algorithm, the results are shown in Fig. 5.

Fig. 6 Arc smooth schematic diagram



The route after the smooth removal of linear image. Sharp turning point is not in conformity with the two-wheel mobile robot motion, thus sharp turning points is removed by using a small segment of circular arc to replace the intersection of straight line on both ends. Its principle is shown in Fig. 6

A, B, and C are three known points, the angle bisector of $\angle ABC$ is done through Point B. Lines AB and CD stand for \overrightarrow{AB} and \overrightarrow{CD} respectively, P1 and P2 are intersections of the arc and two straight line, r is the center of the circle arc, and θ is the angle between the two straight lines.

The equation through straight line AB

$$y = \frac{B \cdot y - A \cdot y}{B \cdot x - A \cdot x} * x + A \cdot y - \frac{B \cdot y - A \cdot y}{B \cdot x - A \cdot x} * A \cdot x$$

The equation through straight line CD

$$y = \frac{B \cdot y - C \cdot y}{B \cdot x - C \cdot x} * x + C \cdot y - \frac{B \cdot y - C \cdot y}{B \cdot x - C \cdot x} * C \cdot x$$

$$\theta = \arccos\left(\frac{\overrightarrow{AB} * \overrightarrow{CD}}{|\overrightarrow{AB}| * |\overrightarrow{CD}|}\right)$$

The arc radius of movement:

$$|p1r| = |p2r| = R, \tag{1}$$

thus

$$\begin{aligned} |P1B| &= |P2B| = R * \cot(\theta/2), \\ \delta1 &= \frac{|P1B|}{|AB|}, \quad \delta2 = \frac{|P2B|}{|BC|} \\ P1 \cdot x &= \delta1 * A \cdot x + (1 - \delta1) * B \cdot x \\ P1 \cdot y &= \delta1 * A \cdot y + (1 - \delta1) * B \cdot y \\ P2 \cdot x &= (1 - \delta2) * B \cdot x + \delta2 * C \cdot x \\ P2 \cdot y &= (1 - \delta2) * B \cdot y + \delta2 * C \cdot y \end{aligned}$$

Lines perpendicular to the lines AB and BC are done through P1 and P2 respectively, with the intersection of r. The circular arc equation is:

$$(x - r \cdot x)^2 + (y - r \cdot y)^2 = R^2$$

Then the equation of trajectory A -> P1 -> P2 -> C is:

$$y1 = \begin{cases} \frac{B \cdot y - A \cdot y}{B \cdot x - A \cdot x} * x + A \cdot y - \frac{B \cdot y - A \cdot y}{B \cdot x - A \cdot x} * A \cdot x & x < p1 \cdot x \\ \sqrt{R^2 - (x - r \cdot x)^2} + r \cdot y & p1 \cdot x \leq x \leq p2 \cdot x \\ \frac{B \cdot y - C \cdot y}{B \cdot x - C \cdot x} * x + C \cdot y - \frac{B \cdot y - C \cdot y}{B \cdot x - C \cdot x} * C \cdot x & x > p2 \cdot x \end{cases} \quad (2)$$

The general equation can be expressed as

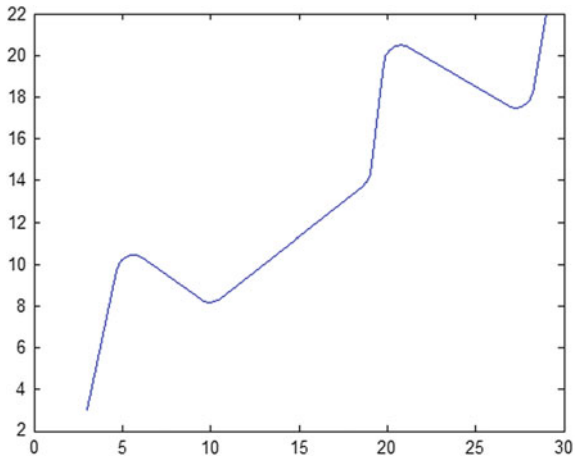
$$y = \begin{cases} y1 \\ y2 \\ y3 \\ \dots \end{cases}$$

The total length of the curve is

$$L = \left| \overrightarrow{\text{StartP1}} \right| + Q(1) * R + \left| \overrightarrow{\text{P2P3}} \right| + Q(2) * R + \left| \overrightarrow{\text{P4P5}} \right| + \dots + \left| \overrightarrow{\text{EndP}_{\text{end}}} \right| \quad (3)$$

After all the points on Fig. 5 has been handled by the above method, the curve shown in Fig. 7 is as follows.

Fig. 7 The optimized route



3 Control of the Robot with Double Differential Speed

From Chap. 2, it could be seen that the route from a start point to the target is scheduled for two-wheel robot movement within the shortest distance with the conditions of a known map. Then the following question is how to make the robot move along this route.

Formula 2 shows that the route planned out is composed of line, arc, line and arc.... In order to make the movement more stable and reliable, the robot should be made to move uniformly as a whole. The double differential model of the robot is shown in Fig. 8.

Assuming that the wheelbase between left and right wheel of robot is L , and the movement speed is V . When robots do rectilinear motion, $V_L = V_R = V$. When robots do circular motion, arc radius is R by Formula 1. From the motion model of the robot:

$$\frac{V_L + V_R}{2} = V \tag{4}$$

If the center position of arc

$$r \cdot x \leq \frac{p1 \cdot x + p2 \cdot x}{2} \parallel r \cdot y \leq \frac{p1 \cdot y + p2 \cdot y}{2}$$

Then

$$\frac{V_L}{V_R} = \frac{R + \frac{L}{2}}{R - \frac{L}{2}} \tag{5}$$

The solution of simultaneous equation 4 and 5 is

$$V_L = V * \frac{2R + L}{2R} \quad V_R = V * \frac{2R - L}{2R}$$

Fig. 8 Two-wheel differential model of the robot

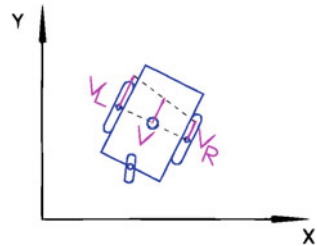


Fig. 9 The right wheel speed

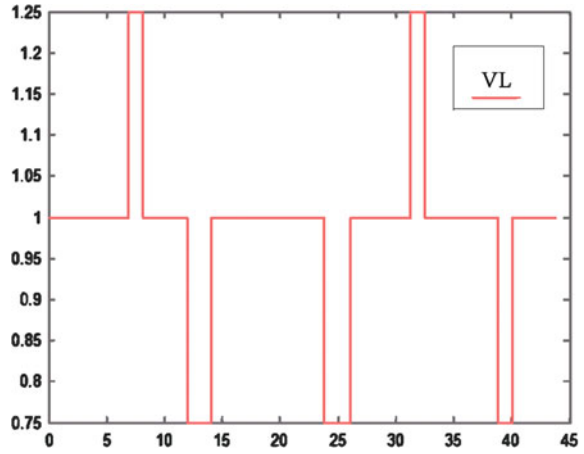
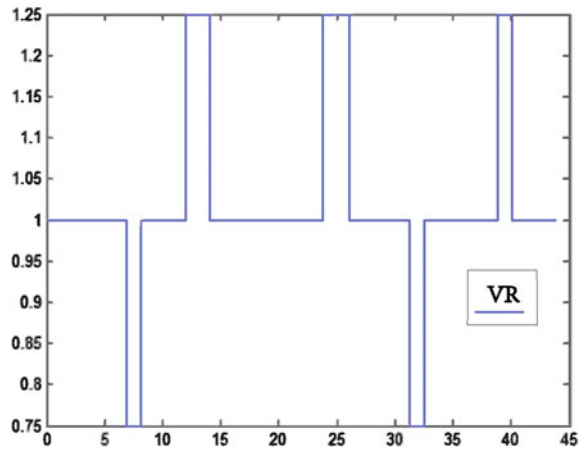


Fig. 10 The right wheel speed



If

$$r \cdot x \geq \frac{p1 \cdot x + p2 \cdot x}{2} \quad || \quad r \cdot y \geq \frac{p1 \cdot y + p2 \cdot y}{2}$$

Then

$$\frac{V_L}{V_R} = \frac{R - \frac{L}{2}}{R + \frac{L}{2}} \tag{6}$$

The solution of simultaneous equation 4 and 6 is

$$V_L = V * \frac{2R - L}{2R} \quad V_R = V * \frac{2R + L}{2R}.$$

If the route for driving is like Fig. 7, the speed of left and right wheel would be shown as Figs. 9 and 10.

4 Dead Reckoning of Two-Wheel Robot

With the running speed of robot, we would pass the speed information to the robot. Then, the robot's left and right wheels would move as we set, so that the track wanted could be produced.

In the process of the robot movement, if the movement condition of the robot needs to be examined, the robot location would be done to observe whether it will break away from the set condition. Thus the calculation process will be divided into two kinds, namely, linear prediction and arc calculated.

From Formula 3, L is the total length of path. According to Formula 4, the running speed of robot is V, thus the total time is $T = \frac{L}{V}$. If the number of sampling points is N, then the time interval $\Delta t = \frac{T}{N}$. From the planned route, the included angle between robot's posture in the beginning and the x axis is $\phi(1)$, and V_{Left} and V_{Right} are the real-time speed of robot's left and right wheels.

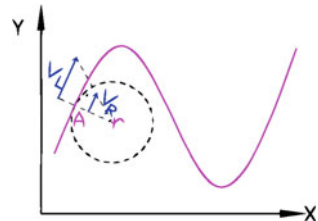
The basic idea of circular arc calculation is shown in Fig. 11.

At point A, the speed of robot's left and right wheels is $V_{Left} \neq V_{Right}$, thus at this time, the robot can be seen as doing short circular motion,

And since

$$\frac{V_{Left}}{V_{Right}} = \frac{R_t - \frac{L}{2}}{R_t + \frac{L}{2}}$$

Fig. 11 Arc calculation



thus

$$R_t = \frac{L}{2} * \frac{V_{Right} + V_{Left}}{\|V_{Right} - V_{Left}\|}$$

$$W_t = \frac{\|V_{Right} - V_{Left}\|}{L}$$

If the known direction angle of robot is in A_t position, the position of the center of the circle r_t at this time could be calculated, followed by the steps below to find out the location of the A_{t+1} position

- Step 1: moving the world coordinate system to r_t , to get coordinate system m;
- Step 2: finding out the location x'_a and y'_a of point A_t relative to the coordinate system m, and calculating the angle of vector $\vec{A_t r_t}$ relative to the x axis in coordinate system m;
- Step 3: making the rotation angle of coordinate system m coincide with vector $\vec{A_t r_t}$, then coordinates m1 is got, and the location x''_a and y''_a of point A_t is obtained based on the relative coordinates system m1;
- Step 4: finding out the location of point A_{t+1} relative to the coordinate system m1, then working out the position of A_{t+1} point relative to the world coordinate system

The deduced formula is as follows:

$$\begin{bmatrix} X_{t+1} \\ Y_{t+1} \end{bmatrix} = \begin{bmatrix} \cos(a) & -\sin(a) \\ \sin(a) & \cos(a) \end{bmatrix} * \begin{bmatrix} \cos(W_t * \Delta t) & 0 \\ \sin(W_t * \Delta t) & 0 \end{bmatrix} * \begin{bmatrix} \cos(a) & \sin(a) \\ -\sin(a) & \cos(a) \end{bmatrix} * \begin{bmatrix} X_t - r_{t,x} \\ Y_t - r_{t,y} \end{bmatrix} + \begin{bmatrix} r_{t,x} \\ r_{t,y} \end{bmatrix}$$

Its location in the sine curve $y = 100 * \sin(\pi * x/200)$ is shown in Fig. 12:

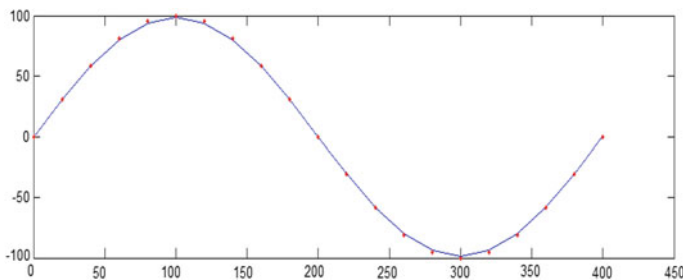


Fig. 12 Calculations on the sine curve

Therefore, the combination calculation of linear and circular arc could be used in the whole process, the basic idea of which is as follows:

Step1: getting the total number of straight line and arc, and the corresponding order i ;

Step2: if $|V_{Left} - V_{Right}| < \delta$ (The δ is a small range)

Calculation in the direction of straight line

$$X(j+1) = X(j) + \frac{V_{Left} + V_{Right}}{2} * \cos(\varphi(i)) * \Delta t$$

$$Y(j+1) = Y(j) + \frac{V_{Left} + V_{Right}}{2} * \sin(\varphi(i)) * \Delta t$$

Step3: if $|V_{Left} - V_{Right}| \geq \delta$

Calculation in the direction of the circular arc

$$\Psi = (W_{t1} + W_{t2} + W_{t3} + \dots) * \Delta t$$

If $V_{Left} > V_{Right}$
 $\varphi(i+1) = \varphi(i) - \Psi$

Else
 $\varphi(i+1) = \varphi(i) + \Psi$

Goto step2

End

Through combination calculation, when $N = 50, 200, 100$, the location maps of which could be seen in Figs. 13, 14 and 15. Figure 16 shows the comparison between the integral calculation and the combination calculation.

Fig. 13 $N = 50$ location map

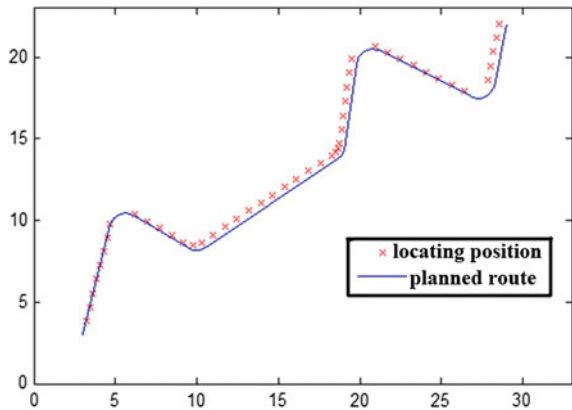


Fig. 14 N = 200 location map

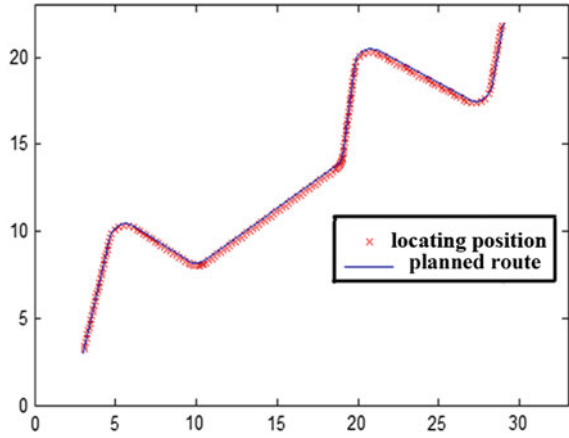


Fig. 15 N = 100 location map

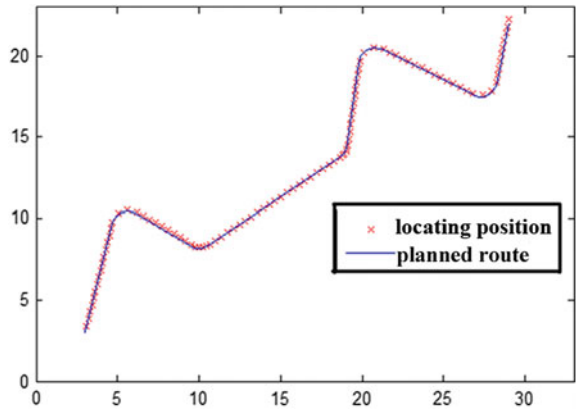
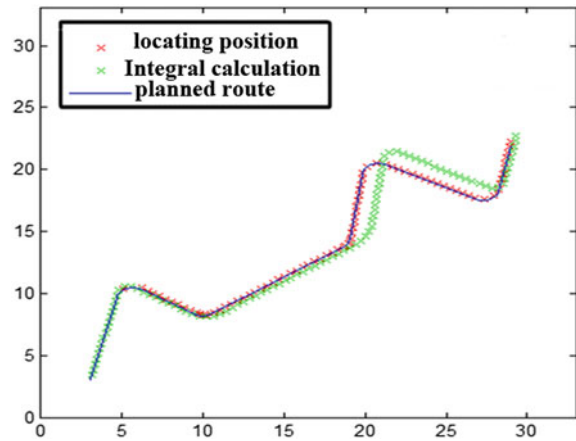


Fig. 16 N = 100 Comparison chart



5 The Experiment

The experimental platform used is the homemade mobile robot named Robot 01, a computer console, and a router for data transmission.

The work process is shown in Fig. 17.

The data in PC is mainly processed by matlab, including path planning and positioning of the robot.

In the environment shown in Fig. 18a, if the smooth processing path of $R = 1$ is chosen, the path by A* algorithm is shown in Fig. 18b. The path by smooth A* algorithm is shown in Fig. 18c. If the acquisition precision $N = 100$ in positioning, combination prediction positioning images is shown in Fig. 18d, and the comparison of combination models and integral calculation are shown in Fig. 18e.

In the environment shown in Fig. 19a, if the smooth processing path of $R = 1$ is taken, the path by the A* algorithm is shown in Fig. 19b, and the path by the

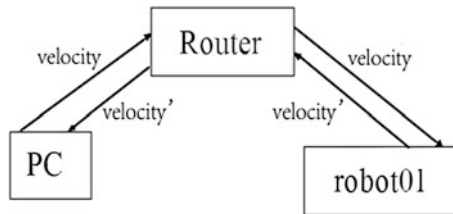


Fig. 17 The flow chart of working process

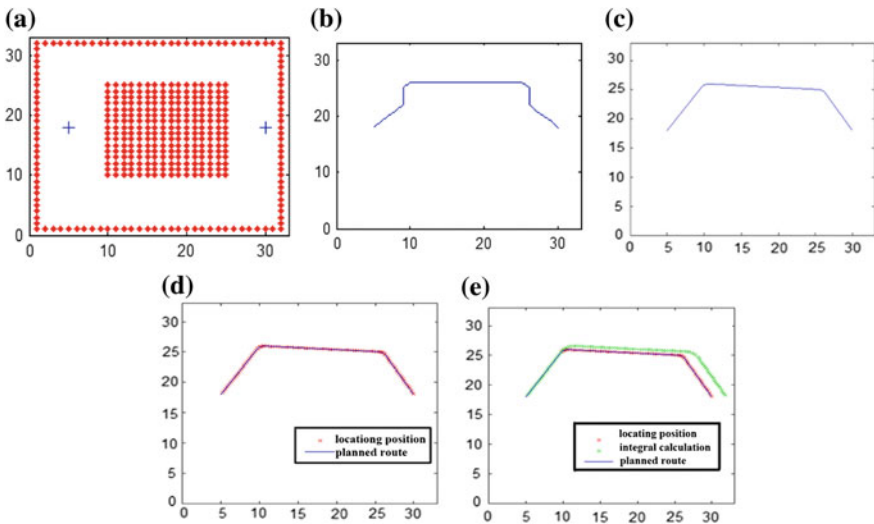


Fig. 18 Path planning and location in experiment 2

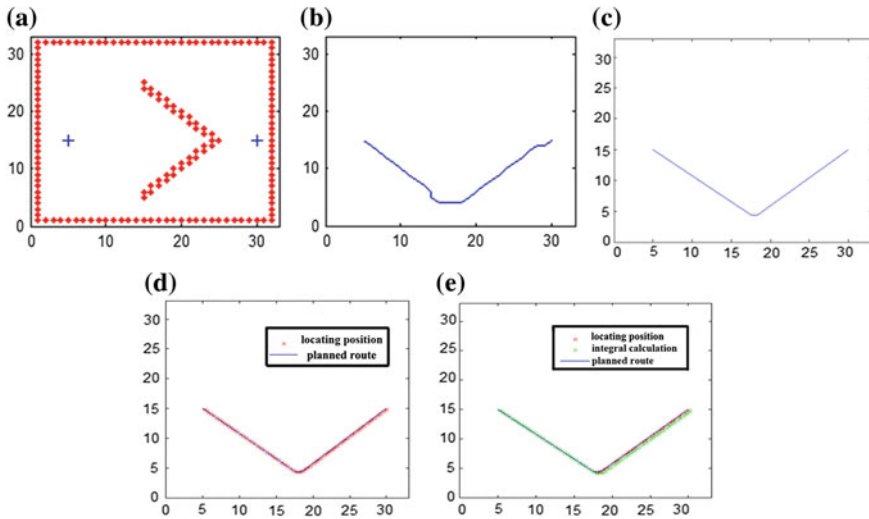


Fig. 19 Path planning and location in experiment 3

smooth A* algorithm is shown in Fig. 19c. If the acquisition precision $N = 100$ in positioning is set, combination prediction positioning images is shown in Fig. 19d, and comparison of combination models and integral calculation is shown in Fig. 19e.

In Table 1, it is shown from the three experimental results that in terms of performance parameters, the route planned out by the smooth A* algorithm is superior to that of the A* algorithm for mobile robot under the condition of different environment and starting point. The reduced rate of the smooth A* algorithm for the path length is about 4–10 %, a total of 66–80 % for the cumulative transition times, and the total turning angle is about 66–77 %. From the data above, it could be concluded that the smooth A* algorithm effectively reduces the length, the number of turns and the turning angle of the mobile robot path planning.

Table 1 The parameters using the smooth A* algorithm before and after comparison

		Length (m)	Reduced rate in length (%)	Twist number	Reduced rate in twist num (%)	Turned angle	Reduced rate in angle (%)
Text 1	A*	49.04	10.48	15	66.7	14.13	59.52
	smooth A*	43.90		5		5.72	
Text 2	A*	35.14	6.5	8	75	4.71	54.78
	smooth A*	32.85		2		2.13	
Text 3	A*	34.70	4.0	5	80	6.28	77.07
	smooth A*	33.3		1		1.44	

Table 2 The performance parameters in different grid environment using track mobile robot combination prediction before and after comparison

Name	Start point end point	Method	Error in x direction (m)	Error in y direction (m)	Sum error $\sqrt{\Delta x^2 + \Delta y^2}$
Experiment 1	(3, 3) (29, 22)	Integral calculation	2.8026	3.7254	4.6619
		Combination prediction	0.0883	0.1691	0.1908
Experiment 2	(5, 18) (30, 18)	Integral calculation	1.8026	0.2746	1.8234
		Combination prediction	0.0157	0.0892	0.0906
Experiment 3	(5, 15) (30, 15)	Integral calculation	0.2229	0.3785	0.4393
		Combination prediction	0.0296	0.1625	0.1652

In Table 2, the ideal path length 43.90 is calculated in Experiment 1 of Table 1. The error of combination models relative to the integral is reduced to $\frac{4.6619-0.1908}{43.90} = 10.18\%$. Similarly, the error in Experiments 2 and 3 is reduced to 5.27 and 5.27 %. Meanwhile, the turning number of path 1, 2, and 3 are 5 times, 2 times and 1 time respectively. Therefore, it can be seen that the path with the increasing number of turning times, the combination prediction is significantly higher compared with the integral calculation, which also proves that the smooth A* algorithm is more suitable for the positioning of robot.

6 Induction and Summary

In researching autonomous walking of intelligent mobile robot, navigation is a basic and very important studying field. This paper explores the combined smooth path planning and dead-reckoning based on the calculation of arc and straight line, to realize automatic navigation and positioning of real robots in a known environment.

However, in a strange environment, we also need map building. This is a new field for us, in which visual information processing will be used. At the same time, the position is based on data from encoder, and the positioning information is not much accurate. If it is combined with visual information and kalman filter for prediction and updating better results will be achieved. Thus, the next step for us is to add visual information, which would enable robot to realize actual automatic navigation and walking.

Acknowledgements Project supported by the National Nature Science Foundation of China (No. 61262059 and No. 31460248).

References

1. LaValle, S.M.: Rapidly-Exploring Random Trees: A New Tool For Path Planning. Department of Computer Science in Iowa State University Ames, IA 50011, USA, pp. 1–4 (1998)
2. Kavrak, L.E., Svestka, P., Latombe, J.C., Overmars, M.: Probabilistic roadmaps for path planning in high-dimensional configuration Spaces. *IEEE Trans. Robot. Autom.* **12**, 566–580 (1996)
3. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **7**(2), 100–107 (1968)
4. Stentz, A.: Optimal and efficient path planning for partially-known environment. In: *IEEE International Conference on Robotics and Automation*, vol. 5, pp. 3310–3317 (1994)
5. Ferguson, Dave, Stentz, Anthony: Using interpolation to improve path planning: the field D* algorithm. *J. Field Robot.* **23**(2), 79–101 (2006)
6. Han, S.M., Lee, K.W.: Mobile robot navigation using circularpath planning algorithm. In: *International Conference on Control, Automation and Systems*, pp. 2082–2086 (2008)
7. Xu, D., Tan, M., Chen, G.: An Improved dead reckoning method for mobile robot with redundant odometry information. In: *International Conference on Control, Automation, Robotics and Vision*, vol. 2, pp. 631–636 (2003)
8. Chitsaz, H., Lavalle, S.M., Balkcom, D.J., et al.: Minimum wheel-rotation paths for differential-drive mobile robots. *Int. J. Robot. Res.* **28**(1), 66–80 (2009)

Adaptive Impedance Control for Docking of Space Robotic Arm Based on Its End Force/Torque Sensor

Gangfeng Liu, Changle Li, Caiwei Song, Liyi Li and Jie Zhao

Abstract Aiming at space transposition using Space Robotic Arm (SRA), flexible docking between SRA's end effector (EE) and grapple fixture (GF) is the most important for space tasks. To avoid position errors leading to large contact force between EE and GF in the docking process, an adaptive impedance control method is proposed in this paper. PID feedforward with adaptive parameters is added into the impedance controller, and the force error function is used to deduce the adaptive parameters according to Lyapunov stability theory, which makes the force error decrease automatically during the connection process. Simulation proves that the adaptive impedance strategy gets better force control effect than the traditional impedance algorithm. Finally the SRA EE/GF connection experiments were conducted respectively based on traditional and adaptive impedance control strategy. The results showed that the adaptive impedance control strategy can achieve better control effect than the traditional strategy.

Keywords Space robotic arm · Impedance control · Adaptive impedance control · Space docking

G. Liu (✉) · C. Li (✉) · C. Song · J. Zhao
State Key Laboratory of Robotic and Systems, Harbin Institute of Technology,
Harbin, Heilongjiang, China
e-mail: liugangfeng@hit.edu.cn

C. Li
e-mail: lichangle@hit.edu.cn

L. Li
School of Electrical Engineering and Automation, Harbin Institute of Technology,
Harbin, Heilongjiang, China

1 Introduction

Dynamics and control of space redundancy manipulator has its particularities [1, 2]. Closure force is formed when robotic arm makes contact operation, and it needs compliant force to solve it [3, 4]. It is necessary to make the dynamic and control simulation because of the complication of docking process [5, 6]. The Ref. [7] applied a Reaction Null/Jacobian transpose control strategy to control and deploy the robotic arms to grasp an orbiting flexible spacecraft. The Ref. [8] built a dynamic model of robotic arm considering the flexible joint and friction characteristics. The Ref. [9] developed a coupling simulation of rigid-flexible dynamics and complex control system by modeling the spacial large-scale robotic arm transposition assembling experiment module based on the virtual prototype. The Ref. [10] presented a novel space station robot with high coverage rail for the space station modules, and the mechanical design and the typical tasks simulation have been studied.

The force control method can decrease interaction force effectively, and help to accomplish the docking mission. Pathak et al. [11, 12] discussed impedance control of space robots using passive DOFs in the controller domain. The Ref. [13] presented an impedance control algorithm based on velocity for capturing two low impact docking mechanisms (LIDMs). The Ref. [14] proposed a 6-D force signals compensation algorithm to guarantee the correctness of impedance control calculation. The Ref. [15] presented a methodology for the impedance control at the interaction point between the robot tips and floating body.

2 Philosophy of the Design

The Space Robotic Arm System built in this paper consists of a 7-DOF symmetrical robotic manipulator and an antigravity suspension system which provides suspension against gravity to simulate micro-gravity environment in space. The Space Robotic Arm (SRA) is able to move around on the specially designed grapple fixtures (GF) by alternatively detaching and connecting its two end effectors (EE) with those grapple fixtures (Fig. 1).

Connection between the SRA's EE and the GF can be divided into three dominant processes: capturing, pulling and locking. During the capturing process, the SRA moves its EE right above the GF and captures its guide bar by straining up three metal strings inside EE. Contact forces parallel to the ground are created in this process. During the pulling process the strings begin to pull the guide bar up at a constant velocity, forcing EE to move down toward GF until they are finally connected. Contact force along the guide bar is created which is vertical to the ground. Finally when the gears on EE are perfectly connected to the gears on GF, the three locks on EE will be locked up. Since the unavoidable position errors of EE could possibly lead to large contact forces, especially during pulling process, an

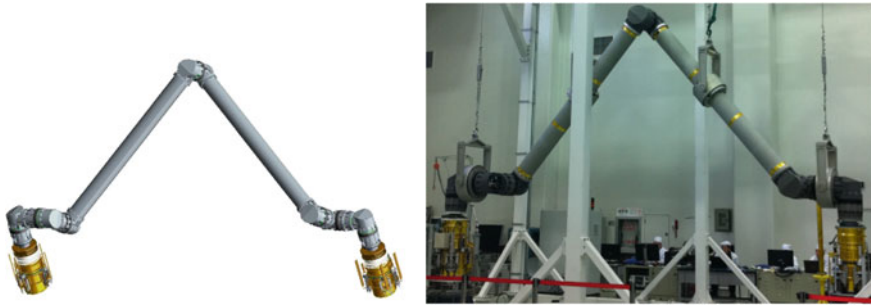


Fig. 1 The space robotic arm and its ground simulator

active compliance control strategy is needed to limit the maximum contact force in tolerance. Feedback of a 6-dimensional force/torque sensor which has been mounted on EE will be used to perform the control strategy.

In this paper, an impedance force control strategy based on feedback from the end 6-DOF force torque sensor is presented. Model of the EE/GF connection is built up and the simulation focuses on the pulling process. But the lack of adaptive ability is more sensitive to the change of environmental parameters in need of improving its robustness. In order to make the impedance controller continuously revise its impedance parameters in the follow-up process, we proposed an adaptive impedance control scheme to make the contact force and position deviation reduce as much as possible. A PID feedforward with adaptive parameters is added into the impedance controller. The force error function is used to deduce the adaptive parameter according to Lyapunov stability theory, which makes the force error decrease automatically during connection process. Simulation is conducted and it proves that the adaptive impedance strategy has better force control ability than the traditional impedance algorithm. Finally the experimental results proved the effect of the adaptive impedance control.

3 Traditional Impedance Control Strategy

The target impedance dynamics could be described as the following differential equation:

$$M_d(\ddot{X} - \ddot{X}_r) + B_d(\dot{X} - \dot{X}_r) + K_d(X - X_r) = E_f \quad (1)$$

where M_d , B_d , K_d are symmetric definite-positive matrices which define mass, stiffness and damping values of the impedance model in different directions of the task space. X_r is the required trajectory while X is the actual position. F is the contact force between the manipulator and the environment.

To implement Eq. 1 into the impedance control of SRA, we use δX to describe the difference between the required trajectory and the desired trajectory based on the target impedance dynamics, i.e. $X_d = X_r - \delta X$. Hence we could rewrite Eq. 1 by means of Laplace transform as:

$$\delta X(s) = \frac{E_f(s)}{M_d s^2 + B_d s + K_d} \tag{2}$$

Since the end force-torque sensor is mounted at the wrist with a distance from the metal strings where the real contact forces are located, a transformation between these two ordinates is needed.

$$F = T(t)F_c \tag{3}$$

where F and F_c are contact forces described in the contact frame $\{O - xyz\}$ and sensor frame $\{O_c - x_c y_c z_c\}$. $T(t)$ is the transform matrix between $\{O - xyz\}$ and $\{O_c - x_c y_c z_c\}$.

Based on this transformation we designed the impedance control strategy. Its control scheme is shown in Fig. 2.

Where F_d is the desired contact force, F is the actual contact force between strings and end effector, and F_c is the contact force in sensor coordinate. F_c is derived from the following equation in which F_a is the feedback from the end force torque sensor and F_b is bias:

$$F_c = F_a - F_b \tag{4}$$

Model of Simulink is built to simulate the proposed impedance control scheme. The pulling process which is most likely of the three processes to have large contact force have been studied in this simulation. The pulling starts at 250 mm above the origin with a velocity of 5 mm/s along Z axis towards the origin. The transport delay between force detection and motion reaction is 4 s. The environment stiffness is 10 N/mm, and the desired contact force is 0 N. Along the Z axis direction, M_d, B_d, K_d is 0.5 kg, 55 Ns/m, 0 N/m respectively. The diagram and simulation results are displayed in Fig. 3.

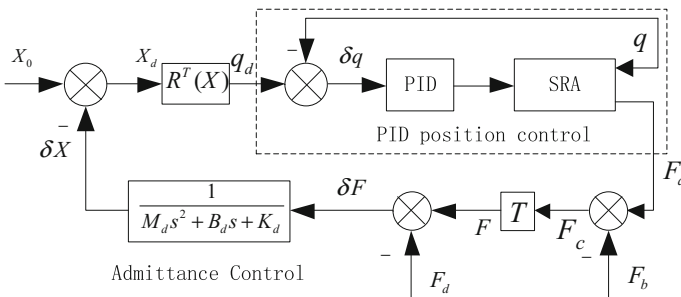


Fig. 2 Diagram of traditional impedance control based on end force/torque sensor

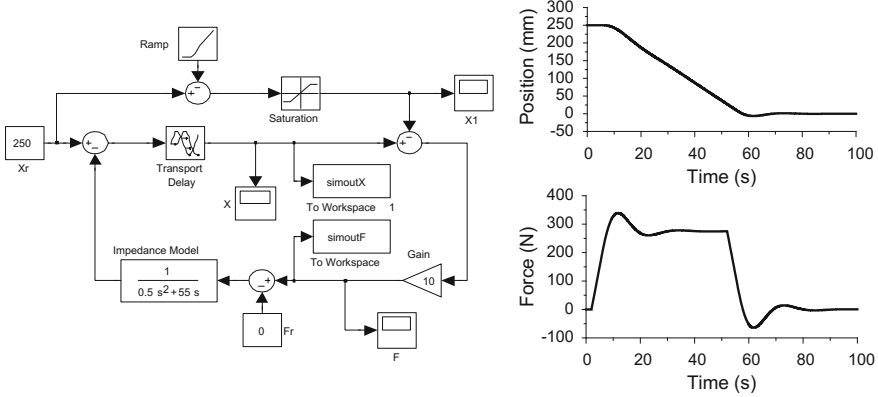


Fig. 3 Simulation of traditional impedance control

4 Adaptive Impedance Control Strategy

In the previous section, impedance parameters based on the environment parameters won't revise along with the change of dynamic contact force error, which leads to poor ability to adapt. In order to make the impedance controller continuously revise its impedance parameters, we use MRAC (Model Reference Adaptive Control) to adjust impedance parameters, which makes the contact force and position deviation reduce as much as possible.

The correction value of original trajectory is Δx_f with modification of the impedance parameters by adaptive controller, and we can get the relationship between trajectory control variable x_c and the original trajectory x_f :

$$x_c = x_f + \Delta x_f \tag{5}$$

Assuming that Δx_f is variable parameter PID feedforward of force control error $e(t)$, and the integral item is equal to a variable parameter $g(t)$.

$$\Delta x_f = g(t) + p(t)e(t) + d(t)\dot{e}(t) \tag{6}$$

The formula 6 is substituted into the formula which is the differential equation of position error:

$$m_d \ddot{e}_f + b_d \dot{e}_f + (k_d + k_e)e_f = k_e(b_d v + k_d vt) \tag{7}$$

We can get:

$$\ddot{e}_f \frac{b_d + k_d k_e d(t)}{m_d} \dot{e}_f = \frac{k_d + k_e + k_d k_e p(t)}{m_d} e_f = \frac{k_d k_e}{m_d} \left[\frac{b_d}{k_d} V + Vt - g(t) \right] \tag{8}$$

Make substitution:

$$\begin{cases} a_p(t) = \frac{b_d + k_d k_e d(t)}{m_d} \\ b_p(t) = \frac{k_d + k_e + k_d k_e p(t)}{m_d} \\ w_p(t) = \frac{k_d k_e}{m_d} \left[\frac{b_d}{k_d} V + V(t) - g(t) \right] \end{cases} \quad (9)$$

We can get contact force transient equations after the variable parameter PID feedforward added as:

$$\ddot{e}_f = a_p(t)\dot{e}_f + b_p(t)e_f = w_p(t) \quad (10)$$

Make $E_p = \begin{pmatrix} e_f \\ \dot{e}_f \end{pmatrix}$, and the transient equation is simplified to the dynamic equation:

$$\dot{E}_p = \begin{pmatrix} 0 & 1 \\ -b_p(t) & -a_p(t) \end{pmatrix} E_p + \begin{pmatrix} 0 \\ w_p(t) \end{pmatrix} = A_p(t)E_p + \begin{pmatrix} 0 \\ w_p(t) \end{pmatrix} \quad (11)$$

Formula 11 is an adaptive impedance control state equation based on variable parameter PID feedforward, where $a_p(t)$, $b_p(t)$, $w_p(t)$ are adjustable control parameters. According to the control principle of MRAC, ideal model can be established, and system stability condition can be obtained by Lyapunov second method. Ideal model of the system is as follows:

$$\ddot{e}_m + a_m \dot{e}_m + b_m e_m = 0 \quad (12)$$

Make $E_m = \begin{pmatrix} e_m \\ \dot{e}_m \end{pmatrix}$, and the transient equation is simplified to the dynamic equation:

$$\dot{E}_m = \begin{pmatrix} 0 & 1 \\ -b_m & -a_m \end{pmatrix} E_m = A_m(t)E_m \quad (13)$$

Formula (13) minus formula (11) is the deviation between actual output and theoretic output, and dynamic equation $E_e = \begin{pmatrix} e_m - e_f \\ \dot{e}_m - \dot{e}_f \end{pmatrix}$:

$$\begin{aligned} \dot{E}_e &= A_m E_e + (A_m - A_p(t))E_p - \begin{pmatrix} 0 \\ w_p(t) \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 \\ -b_m & -a_m \end{pmatrix} E_e + \begin{pmatrix} 0 & 1 \\ b_p(t) - b_m & a_p(t) - a_m \end{pmatrix} \begin{pmatrix} e_f \\ \dot{e}_f \end{pmatrix} + \begin{pmatrix} 0 \\ -w_p(t) \end{pmatrix} \end{aligned} \quad (14)$$

Thus, we construct the quadratic energy function according to Lyapunov second method:

$$\begin{aligned} V(E_e, t) &= \frac{1}{2} E_e^T P E_e + \frac{1}{3} Z^T R Z \\ &= \frac{1}{2} E_e^T P E_e + \frac{1}{2} \beta_0 (b_p(t) - b_m)^2 + \frac{1}{2} \beta_1 (a_p(t) - a_m)^2 + \frac{1}{2} \beta_2 w_p^2(t) \end{aligned} \quad (15)$$

$$\text{Where } Z = \begin{pmatrix} b_p(t) - b_m \\ a_p(t) - a_m \\ w_p(t) \end{pmatrix}, R = \begin{pmatrix} \beta_0 & 0 & 0 \\ 0 & \beta_1 & 0 \\ 0 & 0 & \beta_2 \end{pmatrix}, P = \begin{pmatrix} p_1 & p_2 \\ p_2 & p_3 \end{pmatrix}.$$

Each parameter of R and P is positive, so it is obvious that the energy function is positive. Take the derivative of the energy function and define the positive real symmetric matrix $Q = -(A_m^T P + P A_m)$. Derivative of energy function can be written as:

$$\begin{aligned} \dot{V}(E_e, t) &= -\frac{1}{2} E_e^T Q E_e + (b_p(t) - b_m)(\chi e + \beta_0 \dot{b}_p(t)) \\ &\quad + (a_p(t) - a_m)(\chi \dot{e} + \beta_0 \dot{a}_p(t)) + w_p(t)(\beta_2 \dot{w}_p(t) - \chi) \end{aligned} \quad (16)$$

We can make the latter three items zero to meet the condition that derivative of lyapunov function $\dot{V}(E_e, t) < 0$:

$$\dot{a}_p(t) = \frac{\chi}{\beta_1} \dot{e}, \quad \dot{b}_p(t) = -\frac{\chi}{\beta_0} e, \quad \dot{w}_p(t) = \frac{\chi}{\beta_2} \quad (17)$$

This is parameters' adjustment rule of the adaptive impedance control a. Make $e_m = 0$, and $\dot{e}_m = 0$, Adjustment rules of $g(t)$, $p(t)$, $d(t)$ obtained by Formula (9) are as follows:

$$\left\{ \begin{array}{l} \chi(t) = -\lambda_p e(t) + \lambda_d \dot{x}(t) \\ g(t) = g_0 - \mu_1 \int_0^t \chi dt + \frac{b_d}{k_d} \dot{x}_e + x_e - \sigma_1 \int_0^t g(t) dt \\ p(t) = p_0 - \mu_2 \int_0^t \chi e(t) dt - \sigma_2 \int_0^t p(t) dt \\ d(t) = d_0 + \mu_3 \int_0^t \chi \dot{x}(t) dt - \sigma_3 \int_0^t d(t) dt \end{array} \right. \quad (18)$$

where d_0, p_0, g_0 are the initial value of PID parameters, $\sigma_1, \sigma_2, \sigma_3$ are correction factors, and $\lambda_p, \lambda_d, \mu_1, \mu_2, \mu_3$ are small positive numbers. Thus we obtain the output of the adaptive PID controller:

$$\Delta x_f = g(t) + p(t)e(t) - d(t)\dot{x}(t) \quad (19)$$

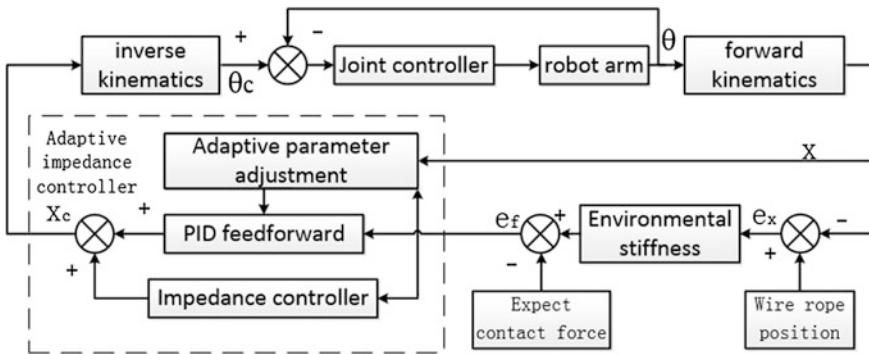


Fig. 4 The principle diagram of adaptive impedance control system

The adaptive impedance controller is added to SRA’s compliant control system, and control system framework is shown in Fig. 4. Adaptive impedance controller consists of three parts: the impedance controller of fixed parameters with genetic algorithm optimization, variable parameter PID feedforward and adaptive parameter adjustment module constantly adjusting PID parameter according to the control error (Figs. 5 and 6).

Model of Simulink is built to simulate the proposed impedance control scheme. Set up the adaptive control parameter $\lambda_p = 0.1$, $\lambda_v = 0.1$, $\mu_0 = 0.1$, $\mu_1 = 0.1$, $\mu_2 = 0.1$. Initial values of PID parameter is zero, and the correction coefficient of σ —

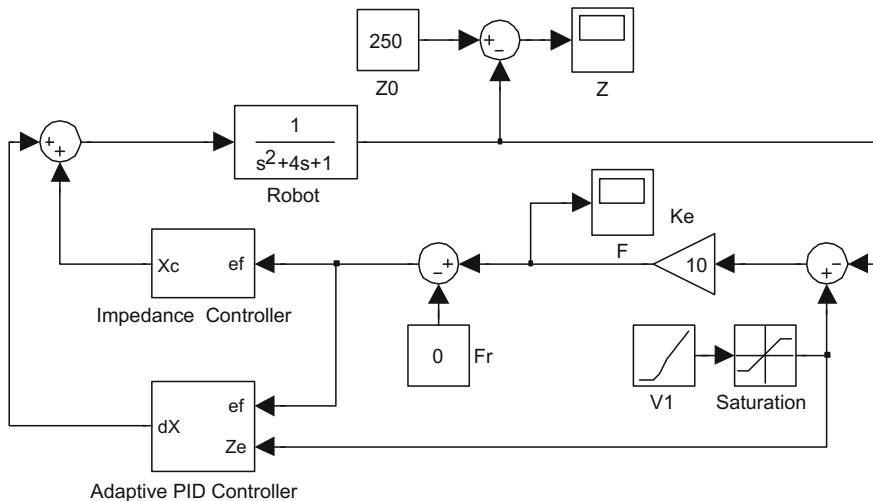


Fig. 5 Diagram of adaptive impedance control

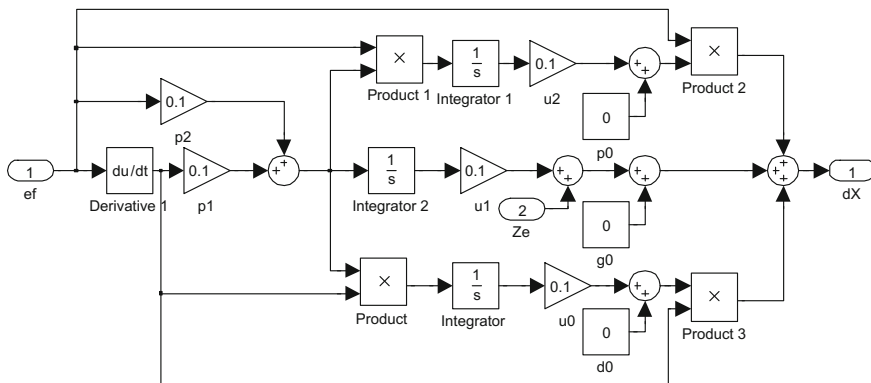


Fig. 6 Diagram of adaptive PID feedforward control

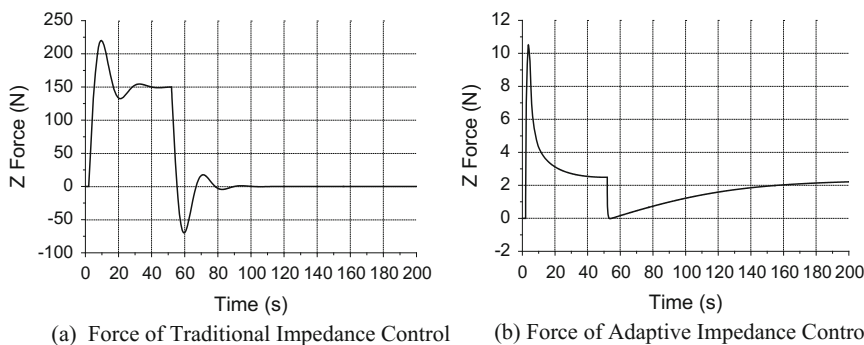


Fig. 7 Contrast of contact force during pulling simulation. **a** Force of traditional impedance control. **b** Force of adaptive impedance control

correction method is set as $\sigma_1 = 0.01$, $\sigma_2 = 0.01$, $\sigma_3 = 0.01$. The pulling process is still studied in this simulation. Position control inner ring is equivalent to a second order system with 4 s delay to improve the simulation speed. The contrast of simulation results in the pulling process is displayed in Fig. 7.

5 Experiments and Analysis

After the impedance control schemes are programmed, experiments are conducted on SRA as shown in Fig. 8 and the results are presented in Fig. 9, where $\{X, Y, F_x, F_y\}$ represents position and contact force on XY plane during capturing process and $\{Z, F_z\}$ is position and contact force along Z axis during pulling process.

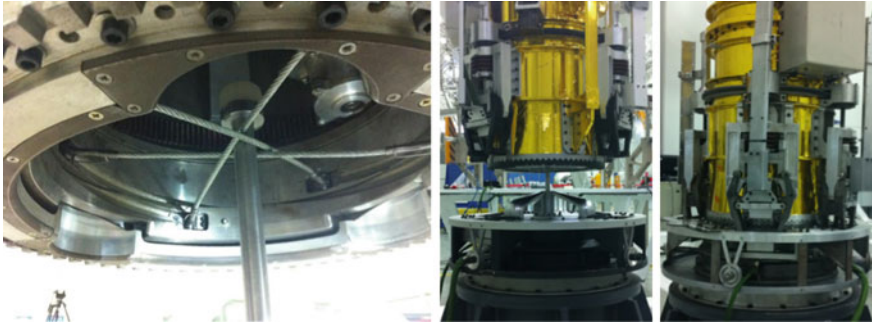


Fig. 8 Experiments of EE/GF connection: capturing (*left*), pulling (*central*) and locking (*right*)

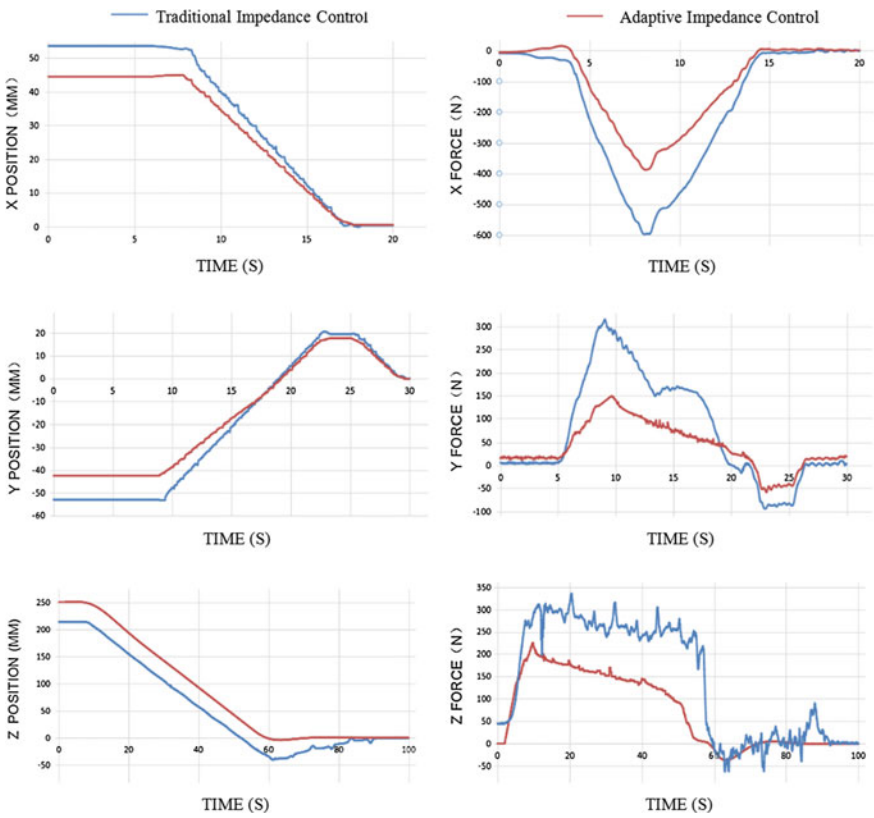


Fig. 9 Comparison of force/motion control effect by traditional and adaptive impedance control

The maximum contact force has been significantly reduced to 400 N from 600 N along X axis, and along Y axis it has been reduced to 150 N from 350 N. During pulling experiment the adaptive control has eliminated the steady state error and controlled the maximum contact force within 250 N. What's more, the contact force of adaptive impedance control along the Y axis isn't stable value as traditional impedance control in a period of time and actually, it constantly decreases. The similar phenomenon occurs along the Z axis in pulling process. Adaptive controller constantly modifies PID feedforward parameters to make the actual contact force decreasing. From the experiment results, we conclude that the proposed adaptive impedance control scheme has controlled the contact force within a limited range during the whole EE/GF connection process successfully.

6 Conclusions

An adaptive impedance control method was introduced based on 6D force-torque sensor mounted on the SRA's end effector. The simulation results showed that this improved method can greatly reduce the contact force than the traditional method. Space docking experiments were carried out under an antigravity suspension system which provided micro-gravity environment as in space. It was shown that the contact forces were controlled under 300 N during capturing process, and the steady-state error in pulling process was small enough to be eliminated.

Acknowledgments This work was supported by National High-tech R&D Program of China (No. 2014AA041601, No. 2015BAF09B02) and China Postdoctoral Science Foundation (No. 2014M561338).

References

1. Ismail, S.Z., Al-Qaisia, A.A., Al-Bedoor, B.O.: Dynamic model of a rotating flexible arm-flexible root mechanism driven by a shaft flexible in torsion. *Shock Vib.* **13**(6), 577–593 (2006)
2. Al-Yahmadi, A.S., Abdo, J., Hsia, T.C.: Modeling and control of two manipulators handling a flexible object. *J. Franklin Inst.* **344**(5), 349–361 (2007)
3. Dubanchet, V., Saussié, D., Alazard, D., Bérard, C., Le Peuvédic, C.: Motion planning and control of a space robot to capture a tumbling debris. *Adv. Aerosp. Guidance Navig. Control*, 699–717 (2015)
4. Wu Jianwei, S.H.I., Shicai, L.H., Hegao, C.A.I.: spacecraft attitude disturbance optimization of space robot. *Robot* **33**(1), 16–27 (2011)
5. Merat, P., Aghili, F., Su, C.Y.: Modeling, control and Simulation of a 6-DOF reconfigurable space manipulator with lockable cylindrical joints. *Intell. Robot. Appl.* **8103**, 121–131 (2013)
6. Dubanchet, V., Saussié, D., Alazard, D., Bérard, C., Le Peuvédic, C.: Modeling and control of a space robot for active debris removal. *Ceas Space J.* **7**, 203–218 (2015)
7. Pisculli, A., Gasbarri, P.: A minimum state multibody/FEM approach for modeling flexible orbiting space systems. *Acta Astronaut.* **110**, 324–340 (2015)

8. Xu, W., Zhou, R., Meng, D.: A hybrid force /position control method of space robot performing. *J. Astronaut.* **34**(10), 1353–1361 (2013)
9. Huaiwu, Z., Hongxia, X., Xiaofeng, S., Meng, C.: Dynamic modeling and simulation research of space station assembly on orbit by large-scale robotic arm. In: 2015 IEEE International Conference on Information and Automation, Lijiang, China, pp. 2281–2286 (2015)
10. Chen, Y., Yang, Y., Qian, H., Xu, Y.: The design of robot for space station operation. In: 2012 IEEE International Conference on Robotics and Biomimetics, ROBIO 2012, Guangzhou, China, pp. 247–252 (2012)
11. Pathak, P.M., Mukherjee, A., Dasgupta, A.: Impedance control of space robots using passive degrees of freedom in controller domain. *J. Dyn. Syst. Meas. Control* **127**(4), 564–578 (2005)
12. Pathak, P.M., Mukherjee, A., Dasgupta, A.: Impedance control of space robot. *Int. J. Model. Simul.* **26**(4), 316 (2006)
13. Chen, C., Nie, H., Chen, J., Wang, X.: A velocity-based impedance control system for a lowimpact docking mechanism (LIDM). *Sensors (Switzerland)* **14**, 22998–23016 (2014)
14. Mo, Y., Gao, S., Jiang, Z., Li, H., Huang, Q.: Impedance control with force signal compensation on space manipulator-assisted docking mission. In: 2014 IEEE International Conference on Information and Automation, Hailar, China, pp. 1239–1243 (2014)
15. Patolia, H., Pathak, P.M., Jain, S.C.: Design of a virtual foundation for impedance control in a dual arm cooperative space robot. *Simulation* **88**, 731–745 (2012)

A Robotic Hardware-in-the-Loop Simulation System for Flying Objects Contact in Space

Chenkun Qi, Xianchao Zhao, Feng Gao, Anye Ren and Yan Hu

Abstract The hardware-in-the-loop (HIL) simulation (also called hybrid simulation) is a useful and flexible approach for the simulation of contact dynamics in space. In this study, a HIL contact simulation system including the motion simulator, control system and algorithm is introduced. The simulation divergence problem due to the time delay is studied. To compensate the time delay, the phase lead method is used to compensate the force measurement delay, and the response error based force compensation is used to compensate the dynamic response delay of the motion simulator. The compensation requires the force measurement delay value, but does not require the dynamic response model. The effectiveness of the HIL simulation system and the delay compensation approach are verified by the simulations and experiments.

Keywords Hybrid simulation · Space docking · Contact dynamics · Simulation divergence · Delay compensation

1 Introduction

In recent years, the space technologies got a lot of achievements in many countries. Space servicing techniques, e.g., rendezvous and docking are very important for the space transporting, upgrading and repairing missions. In these missions, two spacecraft often dock with each other for subsequent servicing. Therefore, the docking process is a very important process. The docking process involves the contact of two spacecraft through the docking hardware. The contact process of

C. Qi (✉) · X. Zhao · F. Gao · Y. Hu
State Key Laboratory of Mechanical System and Vibration,
School of Mechanical Engineering, Shanghai Jiao Tong University,
Shanghai 200240, China
e-mail: chenqi@sjtu.edu.cn

A. Ren
Institute of Aerospace System Engineering Shanghai, Shanghai 201108, China

the spacecraft with the docking hardware should be tested or simulated on the ground. In space, the contact happens in the zero-gravity (0-g) environment. However, on the ground the experimental equipment is in the one gravity (1-g) environment. This is the main difference between the contact in space and the contact on the ground [1, 2].

The implementation of the zero-gravity environment on the ground is very complicated and expensive. To avoid a physical 0-g environment, the hybrid simulation or hardware-in-the-loop (HIL) simulation (also called hybrid simulation) is often used. The computer software is used to calculate the motion of spacecraft in a 0-g environment and the real spacecraft is not required. A robotic simulator on the ground is used to implement the motion of spacecraft. The docking hardware installed on the robotic simulator is used to perform the real contact process and generate the real contact force.

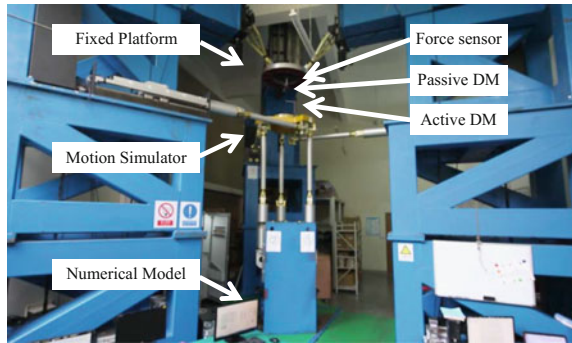
As we know, the time delay in the HIL simulation system makes the simulation divergent. A divergence criterion called the coefficient of restitution (CoR) will exceed one. The main ideas of the delay compensation are the phase lead and feedforward filter. There are several delay compensation methods for the HIL contact simulation system, e.g., phase lead force compensation [3, 4], virtual force compensation [5], virtual damping compensation [4, 6] and feed-forward position compensator [7, 8]. The authors also developed several delay compensation methods, e.g., Smith predictor [9] and response error based force compensation [10]. This study is a partial work of [9, 10].

In this paper, a HIL contact simulation system is introduced. In the delay compensation, the phase lead method is used to compensate the force measurement delay. A response error based force compensation approach is used to compensate the delay of the dynamic response of the motion simulator. The method requires a force measurement delay value and does not require a dynamic response model. From the simulations and experiments, it can be seen that the proposed approach can compensate the simulation divergence of the HIL simulation of contact dynamics.

2 Robotic HIL Simulation System

Figure 1 is the developed HIL simulation system. It includes an upper platform, a force/torque sensor, docking mechanism, a lower motion simulator and a numerical model. The active part of the docking mechanism is installed on the lower motion simulator. The lower motion simulator is used to implement the relative motion of two spacecraft. The force/torque sensor is used to measure the contact force of the docking mechanism. The force/torque sensor is installed between the upper platform and the passive part of the docking mechanism. The force measurement, dynamics model computation and motion control are performed in real-time computers. A six-degree-of-freedom 3-3 perpendicular parallel mechanism [11] is used to design the motion simulator to guarantee high stiffness and high precision

Fig. 1 Robotic HIL simulation system



requirements. Dual-motor actuators [12] are used to implement the high power and high velocity of the motion simulator. The upper platform is designed by using a three-degree-of-freedom parallel mechanism. In the docking simulation, the upper platform does not move.

3 HIL Simulation Divergence

The modeling of each subsystem for the HIL contact simulation system is shown in Fig. 2.

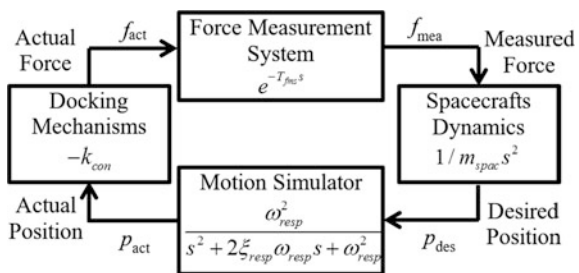
For simplicity, we only consider the one-dimensional contact. Moreover, we assume that the interaction between passive and active docking mechanisms can be described by a mass-spring system. The effects of the damping and friction are ignorable. The contact model is given by

$$f_{act}(t) = -k_{con}p_{act}(t) \tag{1}$$

where $f_{act}(t)$ is the actual contact force, $p_{act}(t)$ is the actual relative position of two spacecraft, k_{con} is the contact stiffness.

For the force measurement delay T_{fms} between the measured force $f_{mea}(t)$ and the actual force $f_{act}(t)$, we assume that it can be described as

Fig. 2 HIL simulation system modeling



$$f_{mea}(t) = e^{-T_{fms}s} f_{act}(t) \tag{2}$$

The motion of spacecraft with respect to the contact force can be described using the Newton’s second law

$$m_{spac} \ddot{p}_{num}(t) = f_{num}(t) \tag{3}$$

where $f_{num}(t)$ is the contact force, and m_{spac} is the equivalent mass of two spacecraft. The computed position $p_{num}(t)$ is also the desired position $p_{des}(t)$ of the motion simulator.

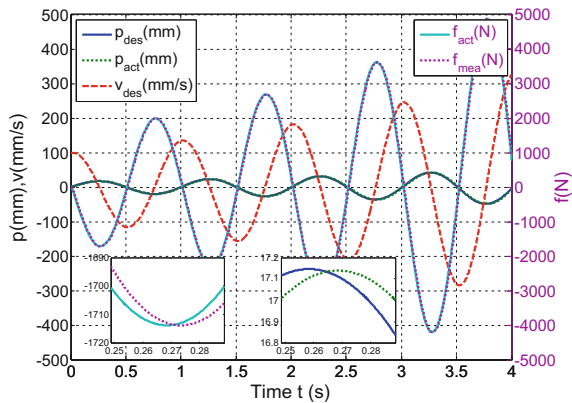
For the motion simulator, the dynamic response is approximated by a simple second-order model

$$p_{act}(t) = H(s)p_{des}(t) = \frac{\omega_{resp}^2}{s^2 + 2\xi_{resp}\omega_{resp}s + \omega_{resp}^2} p_{des}(t) \tag{4}$$

where ω_{resp} is the undamped natural frequency (rad/s), ξ_{resp} is the damping ratio. The response frequency in Hz is denoted as $f_{resp} = \omega_{resp} / 2\pi$. Of course, high-order models can also be used. Here, a second-model is used only for an example. In the compensation algorithm, this model is not required.

To show the HIL simulation divergence, the simulation using the Matlab software is performed. As an example the parameters are set as follows. The contact frequency is 1 Hz, the contact stiffness is 100 N/mm and the initial approach velocity is 100 mm/s. The frequency response of the motion simulator is 25 Hz and the time delay of the force measurement is 5 ms. As shown in Fig. 3, the measured force is delayed to the actual force, and the actual position is also delayed to the desired position. A small window of figure is zoomed in to clearly show the time delay.

Fig. 3 Position, velocity and force of HIL simulation



4 Delay Compensation

The proposed force compensation approach is provided in Fig. 4. It consists two parts: the compensation of force measurement system delay (CFMS) and the compensation of dynamic response of the motion simulator (CDRS). The compensation of the dynamic response delay includes the contact stiffness identification and the dynamic response error of the simulator.

The HIL simulation is a closed-loop real-time simulation. The whole HIL simulation process is provided in the following.

- (1) Set the simulation conditions (i.e., initial position, velocity and acceleration), the motion simulator moves and arrives at the docking conditions.
- (2) The contact mechanism generates the actual contact force $f_{act}(t)$.
- (3) The contact force $f_{mea}(t)$ is measured by using the force/torque sensor.
- (4) The compensated force $\hat{f}_{act}(t)$ for the force measurement delay is calculated.
- (5) The contact stiffness \hat{k}_{con} is estimated by using the contact force $\hat{f}_{act}(t)$ and the actual position $p_{act}(t)$.
- (6) The response compensation force Δf_{resp} is obtained from the response position error $\Delta p_{resp}(t) = p_{des}(t) - p_{act}(t)$ and the estimated contact stiffness \hat{k}_{con} .
- (7) The compensated contact force $f_{comp}(t)$ is obtained.

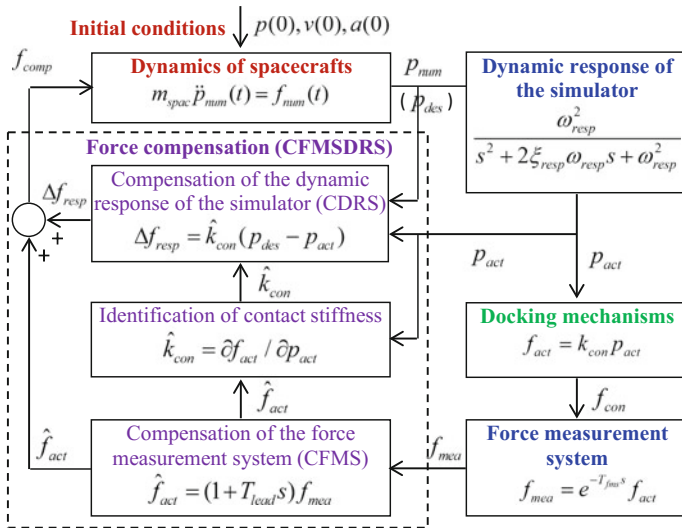


Fig. 4 Delay compensation of HIL simulation system

- (8) The desired position $p_{des}(t)$ is calculated by the dynamics model of the spacecraft.
- (9) The motion simulator realizes the desired position and arrives an actual position $p_{act}(t)$.
- (10) Set $t = t + \Delta t$ and repeat steps (2)–(9).

5 Verifications

5.1 Simulation Verifications

The performance of the HIL simulation system is first verified by using the Matlab software. The simulation conditions include the contact stiffness, the contact frequency and the initial velocity. Using the simulation, the compensation of both the force measurement system and the dynamic response of the simulator (CFMSDRS) is compared with only the compensation of the force measurement system (CFMS).

As an example, the simulation initial conditions are set as follows: the contact stiffness 100 N/mm, the contact frequency 1 Hz, and the initial velocity 100 mm/s. The HIL simulation system parameters are the response frequency 25 Hz and the time delay 5 ms. From Fig. 5, it can be seen that the HIL simulation with the CFMS still has a large divergence; the HIL simulation divergence with the CDRS decreases; and the HIL simulation with the CFMSDRS is similar to the ideal case. From Fig. 6, it can be seen that the CoR with the CFMS is larger than 1; the CoR with the CDRS decreases; and the CoR with the CFMSDRS is similar to 1. It means that the compensation of both time delays is effective and satisfactory. From the contact force in Fig. 7, the compensated force with the CFMSDRS is close to the

Fig. 5 Position of HIL simulation

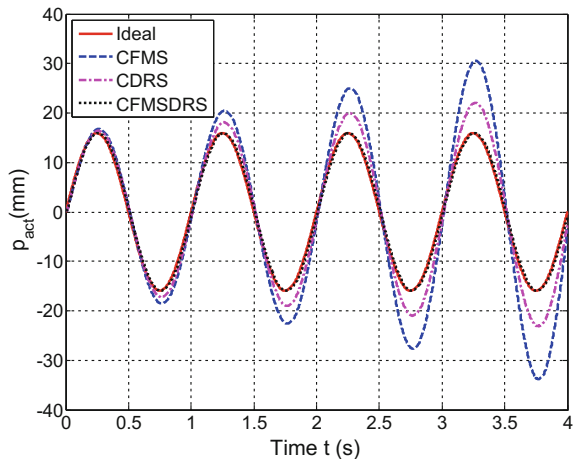


Fig. 6 CoR of HIL simulation

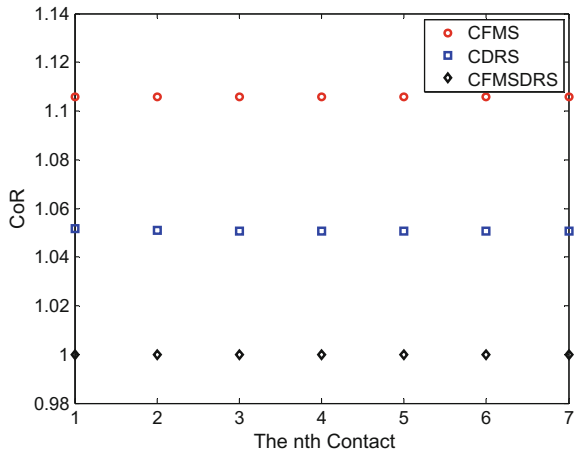
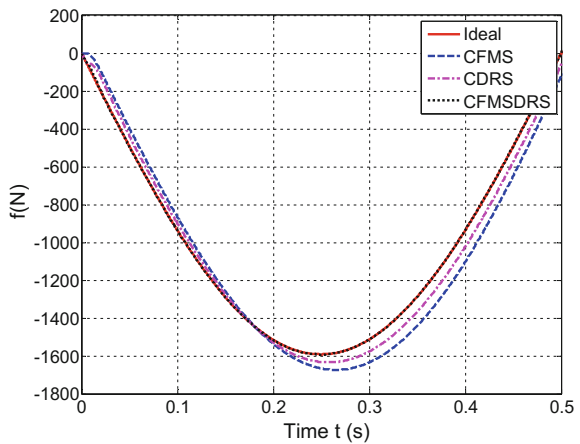


Fig. 7 Force of HIL simulation



ideal force. Because the time delay are made small enough, the proposed method can prevent the simulation divergence.

5.2 Experimental Verifications

The experimental verifications on an undamped elastic rod are performed. The stiffness of the elastic rod is calibrated and known, and the damping is ignorable. In the experiments, we set the contact stiffness 100 N/mm, the contact frequency 3 Hz and the initial velocity 50 mm/s.

Two pictures in the experiments are shown in Fig. 8. From (a) to (b) they denote the initial position and the contact with the lower frame. Because the velocity is not

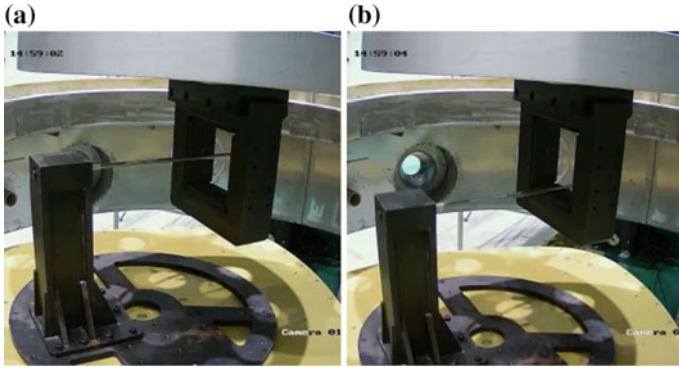


Fig. 8 Pictures of contact experiments

changed between the lower and upper frames, the data between frames are deleted to clearly show the following figures.

Figures 9, 10 and 11 show the desired relative velocity of spacecraft in the HIL simulation under three cases: without the compensation, with the CFMS, and with the CFMSDRS. The CoR under three cases is also shown in Fig. 12, which is about 1.2, 1.1 and 0.98 respectively. The simulation diverges significantly when there is no force compensation. The HIL simulation divergence with the CFMS decreases. However, the divergence is still a little bit large. The HIL simulation divergence with the CFMSDRS disappears with a little convergence. In fact, a little convergence is satisfactory because the HIL simulation system should have a certain stability margin.

Fig. 9 Velocity without the delay compensation

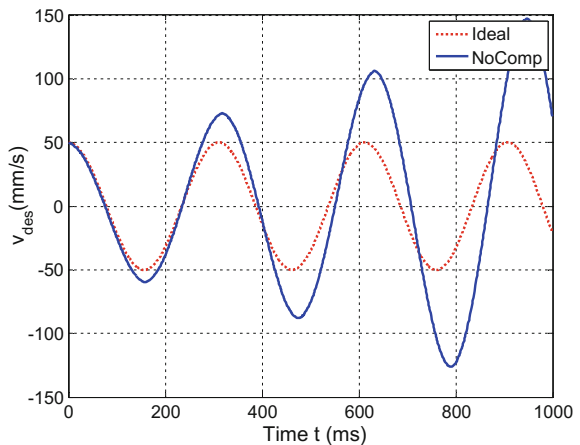


Fig. 10 Velocity with the CFMS

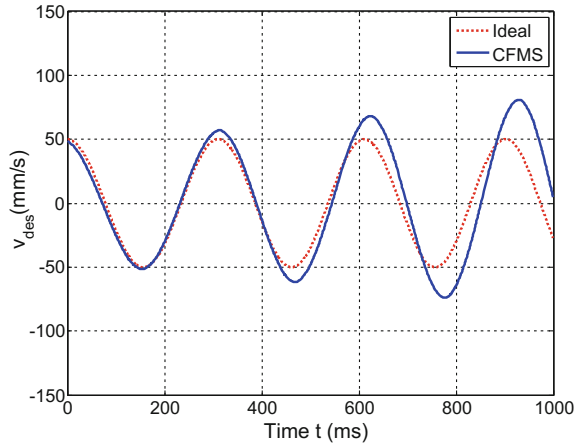


Fig. 11 Velocity with the CFMSDRS

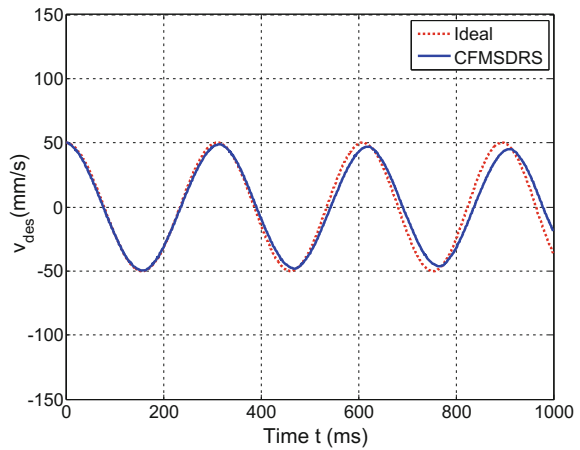
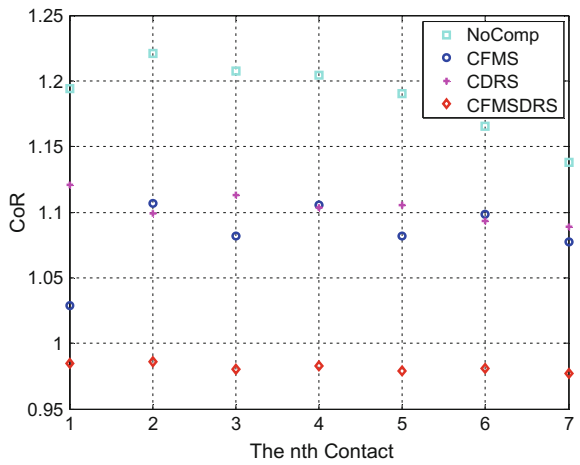


Fig. 12 CoR in the experiment



6 Conclusions

In this study a robotic HIL contact simulation system is introduced. The main purpose is to test the contact process of two flying objects in the 0-g environment in space. The delay compensation problem is also presented in this study. The phase lead method is used to compensate the time delay of the force measurement. A response error based force compensation method is introduced to compensate the dynamic response delay. The delay compensation method in this study requires the force measurement delay value and does not require the motion simulator dynamic response model. From the simulations and experiments, it can be concluded that the HIL contact simulation system can work well and the proposed delay compensate methods are effective.

Acknowledgements The work is partially supported by grants from National Basic Research Program 973 of China (Grant No. 2013CB035501), National Natural Science Foundation of China (Grant No. 51335007, 61473187, 51675328) and Shanghai Natural Science Foundation (Grant No. 14ZR1422600).

References

1. Xu, W.F., Liang, B., Xu, Y.S.: Survey of modeling, planning, and ground verification of space robotic systems. *Acta Astronaut.* **68**, 1629–1649 (2011)
2. Flores-Abad, A., Ma, O., Pham, K., Ulrich, S.: A review of space robotics technologies for on-orbit servicing. *Prog. Aerosp. Sci.* **68**, 1–26 (2014)
3. Shimoji, H., Inoue, M., Tsuchiya, K., Niomiya, K., Nakatani, I., Kawaguchi, J.: Simulation system for a space robot using six-axis servos. *Adv. Robot.* **6**(2), 179–196 (1991)
4. Osaki, K., Konno, A., Uchiyama, M.: Time delay compensation for a hybrid simulator. *Adv. Robot.* **24**(8–9), 1081–1098 (2010)
5. Zebenay, M., Boge, T., Krenn, R., Choukroun, D.: Analytical and experimental stability investigation of a hardware-in-the-loop satellite docking simulator. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **229**(4), 666–681 (2015)
6. Abiko, S., Satake, Y., Jiang, X., Tsujita, T., Uchiyama, M.: Time delay compensation based on coefficient of restitution for collision hybrid motion simulator. *Adv. Robot.* **28**(17), 1177–1188 (2014)
7. Ananthakrishnan, S., Teders, R., Alder, K.: Role of estimation in real-time contact dynamics enhancement of space station engineering facility. *IEEE Robot. Autom. Mag.* **3**(3), 20–28 (1996)
8. Chang, T., Cong, D., Ye, Z., Han, J.: Time problems in HIL simulation for on-orbit docking and compensation. In: 2nd IEEE Conference on Industrial Electronics and Applications, Harbin, China (2007)
9. Qi, C.K., Gao, F., Zhao, X.C., Ren, A.Y., Wang, Q., Sun, Q., Hu, Y., Qiao, L.: Smith predictor based delay compensation for a hardware-in-the-loop docking simulator. *Mechatronics* **36**, 63–76 (2016)
10. Qi, C.K., Ren, A.Y., Gao, F., Zhao, X.C., Wang, Q., Sun, Q.: Compensation of velocity divergence caused by dynamic response for hardware-in-the-loop docking simulator. *IEEE/ASME Trans. Mechatron.* (2016). doi:[10.1109/TMECH.2016.2601219](https://doi.org/10.1109/TMECH.2016.2601219)

11. Cao, R., Gao, F., Zhang, Y., Pan, D.: A key point dimensional design method of a 6-DOF parallel manipulator for a given workspace. *Mech. Mach. Theory* **85**, 1–13 (2015)
12. Zhang, J.Z., Gao, F., Yu, H.N., Zhao, X.C.: Use of an orthogonal parallel robot with redundant actuation as an earthquake simulator and its experiments. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **226**(1), 257–272 (2012)

Particle Filter on Episode for Learning Decision Making Rule

Ryuichi Ueda, Kotaro Mizuta, Hiroshi Yamakawa and Hiroyuki Okada

Abstract We propose a novel method, a particle filter on episode, for decision makings of agents in the real world. This method is used for simulating behavioral experiments of rodents as a workable model, and for decision making of actual robots. Recent studies on neuroscience suggest that hippocampus and its surroundings in brains of mammals are related to solve navigation problems, which are also essential in robotics. The hippocampus also handle memories and some parts of a brain utilize them for decision. The particle filter gives a calculation model of decision making based on memories. In this paper, we have verified that this method learns two kinds of tasks that have been frequently examined in behavioral experiments of rodents. Though the tasks have been different in character from each other, the algorithm has been able to make an actual robot take appropriate behavior in the both tasks with an identical parameter set.

Keywords Particle filter · Decision making · Learning · Episodic memory

R. Ueda (✉)

Chiba Institute of Technology, 2-17-1 Tsudanuma, Narashino, Chiba, Japan
e-mail: ryuichi.ueda@p.chibakoudai.jp

K. Mizuta

Riken Brain Science Institute, 2-1 Hirosawa, Saitama, Wako, Japan

H. Yamakawa

Dowango Artificial Intelligence Laboratory, Kabukiza Tower,
4-12-15 Ginza, Chuo-ku, Tokyo, Japan

H. Okada

Graduate School of Brain Sciences, Tamagawa University,
6-1-1 Tamagawa-gakuen, Machida, Tokyo, Japan

© Springer International Publishing AG 2017

W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_54

1 Introduction

Whether it is an animal or an autonomous robot, every agent in the real world must solve navigation problems so as to fulfill some purposes. Though they have different types of calculators in their bodies, the transformation from sensor information to appropriate sequences of actions is essential and should be solved for finishing a task.

Recent studies on neuroscience suggest that the hippocampal formation in brains of mammals play an important role in solving navigation problems [2]. From brains of rodents, some kinds of cells that have special roles in processing of spatial information have found from 70s to the present [8, 10, 15, 16, 22]. Grid cells that react to displacement of the owner of the brain [15], and place cells that form maps of familiar environments [16] are famous in particular.

These kinds of knowledge have been utilized in robotics. Especially, Milford et al. have proposed RatSLAM [12]. They have simulated some functions of cells in the hippocampal formation and have successfully created a map of streets with a camera on a car. Not just RatSLAM but dozens of applications inspired by brains exist in robotics as reviewed in [5, 11].

It is a challenging problem if we aim to implement brain inspired algorithms on robots not only as applications but also as workable models that explain functions of brains. Though such models will not easily go much beyond hypotheses, to simulate behavior of rodents with actual sensors and actuators will be worth trying for considering behavioral principle of agents in the real world. In this paper, we implement a simple but workable decision making method on a micromouse robot as the first step of this challenge.

For the implementation, we take much notice of the findings that the hippocampus handles memories of events [9, 30] though the ability of map building has been mainly emphasized in discussions of navigation. In a brain of rodents, the hippocampus receives information of motion, landmarks, and the shape of an environment from the entorhinal cortex. When we name the sets of these primary information events, maps built from them are secondary information from a sequence of events. This conversion process will not be required if the brain can memorize events infinitely. It can decide actions directly from the sequence of events.

Therefore, we choose a decision making method that only uses events as the first one for the workable model. This model will be useful for studies of navigation with maps since we can examine various algorithms that build or use maps through addition or modification to it.

When such a method is implemented on a robot, events can be stored in its computer virtuality to infinity. However, the time complexity for decision making will increase in proportion as the amount of events if the implementation is naive. We utilize a particle filter for this problem. Calculations for decision are done on the sequence with a particle filter in real-time. Operations of the particle filter have some similarities to the retrospective and prospective activities of place cells [18].

The structure of this paper is as follows. In Sect. 2, we refer to some related works. The problem solved by the proposed method is explained in Sect. 3, and the method is given in Sect. 4. We have experiments with an actual robot in Sect. 5, and conclude this paper in Sect. 6.

2 Related Works

This study is incorporated in *the whole brain architecture project*, which is launched by Dwango artificial intelligence laboratory recently. Its early concept has been shown in [29]. In the project, they try creating a general-purpose artificial intelligence software through continuous modifications of hypotheses and software models of a brain. Experiments with actual robots as performed in this paper are vital for studying artificial intelligence that performs in the actual world.

Reinforcement learning methods [24] relate to the learning mechanism of brains, and some of them store past events. In Kernel-based reinforcement learning proposed by Ormonite and Sen [17], past events are utilized for finding appropriate approximation of value functions. Reinforcement-learning on episodes has been proposed by Unemi and Saitoh [28], and it has been applied to tasks on a simulator. This method utilizes past events for direct estimation of appropriate actions and has some similarities with our method. They have tackled a problem with perceptual aliasing [26], which also occurs in a task examined in this paper. Though there are some heuristics in their method, we can simplify it with a particle filter.

Particle filters are frequently used for self-localization [4], SLAM (simultaneous localization and mapping) [14], and decision making [27] for real-time calculations. The proposed method is also executed in real-time on an actual robot.

Artificial neural networks (ANNs) are also established tools inspired by brains [13, 25]. They are used for simulating brains in most cases and our project also utilizes it. Though we do not use ANNs for the implementation of the proposed method, representation of the method on ANNs will be an interesting study.

There are a lot of studies that take the biomimetic approach for generating behavior of robots toward navigation tasks [5, 11]. Though the method proposed in this paper is categorized in the simplest one in the methods, we pay attention to keep its expandability for future studies. Since our method works in the actual world at the present moment, we can compare various extensions that are inspired by functions of a brain in the actual world based on it.

3 Problem Definition

3.1 A Decision Making Problem

We assume a discrete-time system. An agent belongs to a state space \mathcal{X} in the system. The agent cannot observe the state $x \in \mathcal{X}$ at any time step directly, while it receives

a set of sensor measurements. The set is called *an observation* and we represent it a symbol $o(t)$, where $t \in \{0, 1, 2, \dots\}$ denotes a time step.

The agent is given a task. To finish this task successfully, the agent must choose an action $a(t) \in \mathcal{A}$ at each time step. The agent tries the task repeatedly and learns an appropriate decision making rule for the task. One trial of the task is simply called *a trial* in this paper.

At the end of a trial, a reward $r(t) \in \mathfrak{R}$ is given for evaluation of the agent's behavior in the trial. In any other time step, the reward is zero. The reward is given based on a hidden rule.

The agent memorizes the sequences of sensor values, rewards, and actions continuously in trials. The sequences are combined as *an episode*:

$$\begin{aligned} \mathcal{E}(T) &= \{o(0), r(0), a(0), o(1), r(1), a(1), \dots, o(T-1), r(T-1), a(T-1)\} \\ &= \{e(t) | t = 0, 1, \dots, T-1\} \end{aligned} \quad (1)$$

where T is the current time, and a set $e(t) = [o(t), r(t), a(t)]$ is named *an event* at time t . Note that $t = 0$ is not the start time of every trial but that of the first trial. It means that events of multiple trials are recorded in this episode. T always points the current time, which is incremented after the action $a(T)$ is chosen and done.

We handle a problem to make the agent find an action $a(T)$ that will maximize the evaluation:

$$J = \frac{1}{t_f - t_0} r(t_f), \quad (2)$$

where t_0 and t_f represent the time steps at which a trial starts and is finished respectively. When $r(t_f)$ is 1 or -1 , for example, the agent should obtain the positive reward as quickly as possible while avoiding the negative reward. The robot can use \mathcal{E} and a pair of the latest information and reward $[o(T), r(T)]$ for decision making.

3.2 Relation Between the Formulation of the Episode and Functions of Brains

The hippocampus receives sensor inputs $o(t)$ and information of motion, which is $a(t)$ in the context of the above problem, mainly from the entorhinal cortex (EC).

We assume that the sensor input $o(t)$ is related to visual cues that are sent from the lateral entorhinal cortex (LEC) to the hippocampus [7], or boundary information of the environment that is generated by *border cells*. Border cells are found by Solstad et al. in the medial entorhinal cortex (MEC) [21]. They are activated when a mouse or rat exists near a wall. This information is sent to the hippocampus. We use the values of range sensors as a substitute for this information from the border cells in the later experiment.

The action $a(t)$ is related to and simplified from the information of displacement that is generated by grid cells, which are discovered from MEC by Hafting et al. [6]. Grid cells give odometry or dead reckoning information in terms of robotics to the hippocampus. In the above formulation of the problem, the sequence of actions $a(t)$ ($t = 1, 2, \dots$) is substitution of the odometry information though the sequence is too abstracted to be the substitution. We should generalize the problem with odometry in future.

The existence of rewards $r(t)$ ($t = 1, 2, \dots$) is required for formulating a task in the context of the optimal control theory for robots. In the cases of mammals, the decision making theory is discussed based on rewards. Decision making is not the function of the hippocampus, but handled by the prefrontal cortex (PFC) [8, 22] and other parts of a brain.

We have assumed that the length of the episode can be infinite in the problem definition. In the case of mammals, however, old events are forgot or abstracted to other forms. The information of events is reduced into a map of the environment as observed through the activity of place cells in the hippocampus [1, 16].

The reason why we assume unforgotten sequences is that a transformation from events into a map can be regarded as an approximation since this transformation erases the time axis. Though the mechanism of transformation is also an important issue in this study, we want to fix a reference model of decision making without the lossy transformation at the beginning.

4 Particle Filter on Episode

We propose a novel particle filter that is operated on the episode. This particle filter represents the agent's internal state of which past events are recalled at the time.

4.1 Probability Distribution on an Episode

We assume that an agent recalls some past events when it decides its action. Let us define a weight $w[\mathbf{e}(t)]$ ($0 \leq t < T$) for each time step, where $\sum_{t=0}^{T-1} w[\mathbf{e}(t)] = 1$. This value quantifies how much the event at a time step t is recalled by the agent at the current time T . The function $w : \mathcal{E} \rightarrow \mathfrak{R}$ becomes a probability distribution on the episode.

This function w is regarded as an internal state of the agent. We name this state a *memory recall state* in the track of *the belief state*, which is regarded as an internal state in the studies of mobile robot localization [26].

4.2 Definition of Particles

For real-time calculation, we approximate the memory recall state w by using a particle filter. Particle filters are frequently used for mobile robot localization [4].

The particle filter for the memory recall state has the following particle set:

$$\{\xi^{(i)} = (t^{(i)}, w^{(i)}) \mid i = 1, 2, \dots, N\}, \quad (3)$$

where $\xi^{(i)}$ is i th particle. N is the number of particles in the particle set.

4.3 Operation of Particles

The particle filter operates particles with the following procedures when the agent receives information.

4.3.1 Sensor Update

In a particle filter, sensor information is reflected to the distribution of particles by Bayes theorem. This procedure is called a sensor update. This procedure requires likelihood functions, which are given to the agent beforehand.

The particle filter on episode uses a likelihood function that is formulated as $L[o(T), r(T)|e(t)]$. This function quantifies the likelihood of the observation and the reward at T with the assumption that the event $e(t)$ is appeared again at T .

With this likelihood function, the particle filter operates particles as:

$$w^{(i)} \leftarrow w^{(i)} L[o(T), r(T)|e(t^{(i)})] \quad (i = 1, 2, \dots, N) \quad (4)$$

$$\alpha \leftarrow \sum_{i=1}^N w^{(i)} \quad (5)$$

$$w^{(i)} \leftarrow w^{(i)} / \alpha, \quad (6)$$

where the arrow denotes a substitution. The values of likelihood are reflected in the particles at the procedure in Eq. (4), and the weights are normalized by Eqs. (5) and (6).

4.3.2 Motion Update

The time steps of particles are incremented from $t^{(i)}$ to $t^{(i)} + 1$ after an action $a(T)$ with an increment of the actual time step. This procedure is called a motion update in this paper though this word means positional changes of particles in the context of self-localization.

In this process, weights of particles such that $a(t^{(i)}) \neq a(T)$ are reduced to zero because the state transition from $t^{(i)}$ to $t^{(i)} + 1$ becomes different from the current state transition.

After that, a new particle set is resampled from the old set. Some particles that have heavy weights are split into more than one particles whose weights are $1/N$, while some particles with small weights are erased.

4.3.3 Decision Making

The action $a(T)$ is chosen by a voting algorithm. Each particle evaluates each action that can be chosen at time T with

$$v(\xi^{(i)}, a) = \begin{cases} \frac{r(t')}{t' - t^{(i)}} & (a = a(t^{(i)})) \\ 0 & (a \neq a(t^{(i)}) \text{ or } t' > T), \end{cases} \quad (7)$$

where t' is the first time step when the agent obtained a nonzero reward after time $t^{(i)}$. t' can be obtained by a forward-looking of the sequence from $t^{(i)}$. Though we should not relate this algorithm to activities in a brain without careful consideration, a similar forward-looking procedure has been observed from activities of the hippocampal place cells in several studies [19].

The action whose weighted sum of v is larger than any other action is chosen as the action that should be taken by the agent. This calculation is formulated as

$$a(T) = \underset{a}{\operatorname{argmax}} \sum_{i=1}^N w^{(i)} v(\xi^{(i)}, a). \quad (8)$$

4.3.4 Retrospective Resetting

When the memory recall state is inconsistent with current information, a replacement of particles is required. This procedure is called resetting, which is essential for using a particle filter in the actual world.

We propose a novel resetting procedure that is named *the retrospective resetting*, which is inspired by the retrospective activity of place cells [18]. This reset occurs when the value of α in Eq. (5) is smaller than a threshold α_{th} . It also occurs at the motion update by the comparison of $a(T)$ and $a(t^{(i)})$.

The retrospective resetting simulates all procedures of the particle filter from a time step $T - \lambda$ to the current time step T . At first, particles are placed randomly on the episode \mathcal{E} . Next the events from $T - \lambda$ to T are used for sensor updates and motion updates. After these procedures, we can expect that the particles converge to several time steps whose preceding event sequences are similar to the sequence of events from $T - \lambda$ to T .

5 Experiment

The proposed method is evaluated with a micromouse type robot and a T-shaped maze. We examine two kinds of tasks that have different characters.

5.1 Environment

T-shaped mazes, which are called T-mazes in neuroscience, are frequently used for behavior experiments of rodents [3, 23, 30]. We have build a T-maze for the micromouse with a standard maze kit for micromouse competitions as shown in Fig. 1. The walls are placed on $180 \times 180 \text{ mm}^2$. The thickness and height of every wall are 12 and 50 mm respectively. We give names to the three arms of this maze as shown in the figure.

5.2 Robot

We use the “Raspberry Pi Mouse” made by RT Corporation. This robot is shown in Fig. 2a. As indicated by the name, this robot is a micromouse that has a Raspberry Pi (the Raspberry Pi 2 Model B) as its computer.

Fig. 1 T-maze for experiments

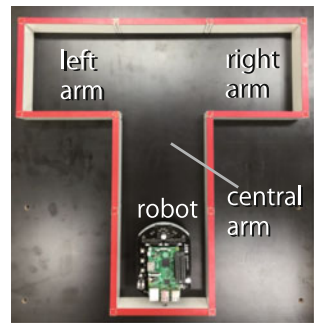
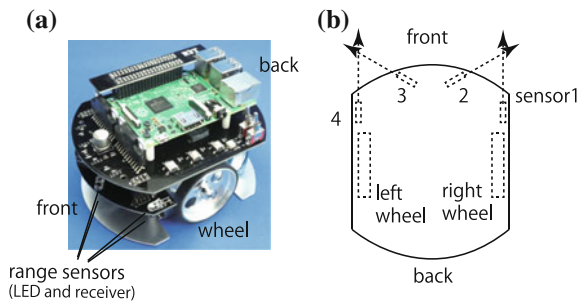


Fig. 2 a Raspberry Pi mouse (a photograph by courtesy of RT Corporation). b Positions of sensors and wheels (top view)



This robot has four range sensors shown in Fig. 2b. The value of the sensor i is denoted by $\ell_i(t)$ ($i = 1, 2, 3, 4$). Each sensor receives a value near zero when there is no wall in its front. When a wall exists in front of the sensor, it receives a value near 4000. We have roughly measured the relation between distances and sensor readings in the environment. The result is shown in Appendix A.1.

5.3 Actions and Observation

The robot can choose an action from $\mathcal{A} = \{a_{\text{fw}}, a_{\text{right}}, a_{\text{left}}, a_{\text{stay}}\}$. The action a_{fw} makes the robot go forward until a wall exists on the front of the robot. When the sum of values from all of the sensors exceeds a threshold, the robot senses the existence of a wall. The actions a_{right} and a_{left} makes the robot change its direction around the midpoint of the wheel axis. The turn is finished when the value of $\ell_1(T) + \ell_4(T)$ falls below a threshold (T : the current time step). a_{stay} is chosen between the end of a trial to the start of the next trial.

Before each action $a(T)$ is chosen and done, values of the range sensors are recorded as an observation $o(T) = [\ell_1(T), \ell_2(T), \ell_3(T), \ell_4(T)]$. Moreover, the robot receives and records the reward $r(T)$ at this moment.

5.4 A Trial

Before each trial, a marker is placed at an arm, which is called the reward arm. The mouse starts from the central arm and turns one of the arm. When the robot goes to the reward arm, the robot obtains a reward, whose value is one. When the robot goes to the other side, -1 is given as a penalty. Since the robot cannot detect the marker, the reward is given to the robot through the buttons at the top board of the robot. Figure 3 shows a successful trial and a failed trial as examples.

Through dozens of trials, the robot learns the choice of a_{right} or a_{left} at the T-junction. The robot makes only one decision in each trial. Though this method has a possibility to make a robot learn a whole sequence of actions, we leave this generalization as a future work. In each trial, the robot is placed at the initial position shown in Fig. 1. At the start, the robot can only chooses a_{fw} , and goes to the T-junction. Then the robot chooses a_{right} or a_{left} . After the robot changes its direction to right or left, it takes a_{fw} again and does a_{stay} at the end of one of the left or right arm.

The robot obtains four events:

$$\begin{aligned} \mathbf{e}(t) &= [o(t), 0, a_{\text{fw}}], & \mathbf{e}(t+1) &= [o(t+1), 0, a_{\text{right}} \text{ or } a_{\text{left}}], \\ \mathbf{e}(t+2) &= [o(t+2), 0, a_{\text{fw}}], \text{ and} & \mathbf{e}(t+3) &= [o(t+3), 0 \text{ or } 1, a_{\text{stay}}] \end{aligned}$$

marker of reward (used for the instructor)

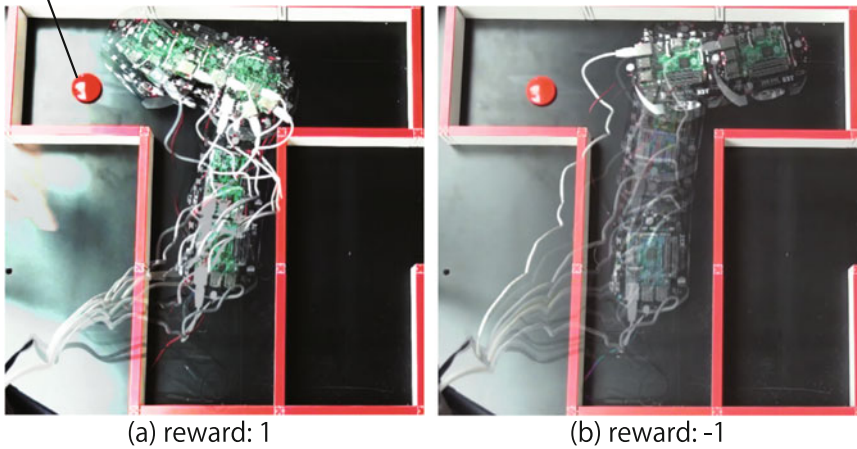


Fig. 3 a A successful trial. b A failed trial

in each trial. They are recorded as a part of the episode. We give actual values of these events in an experiment in Table 4.

5.5 Implementation of Likelihood Function

The likelihood function used for this experiment is not derived from any statistical data but from some subjectively-determined rules:

- when the reward is not given at the current time T despite $r(t) = 1$ at the particle's time t , the value of likelihood is zero, and
- when the values of a sensor at time T and time t are totally opposite, the value of likelihood is reduced to half.

This way of definition is based on Bayesianism.

We implement the following likelihood function:

$$L[\mathbf{e}(T)|\mathbf{e}(t)] = \delta[r(T), r(t)] \cdot \left(\frac{1}{2}\right)^{\sum_{i=1}^4 \Delta_i}, \quad (9)$$

$$\text{where } \Delta_i = \frac{|\ell_i(T) - \ell_i(t)|}{4000} \text{ and } \delta(a, b) = \begin{cases} 0 & (a = b) \\ 1 & (\text{otherwise}) \end{cases}$$

based on the above rules. When $\ell_i(T) = 4000$, $\ell_i(t) = 0$ ($i = 1, 2, 3, 4$), and $r(T) = r(t)$, for example, the value of $L[\mathbf{e}(T)|\mathbf{e}(t)]$ becomes $(1/2)^4 = 1/16$.

The weight of each particle is changed based on this function. The number of particles is fixed to 1000 in the tasks.

5.6 Tasks

We have prepared two kinds of tasks. They have different rewarding rules from each other.

5.6.1 Task 1: Learning a Periodic Task

In this task, we examine whether the proposed method can learn a simple periodic task or not. As shown in Fig. 4, the reward is set at the right arm or the left arm alternately. When the reward is set at the left/right arm at a trial, it is set at the right/left at the next trial.

The robot becomes able to obtain the reward every four state transitions after all of the particles are converged to the event at $t = T - 8$. However, particles are easily replaced by a reset. In the early part of the episode, moreover, mistakes are contained. We check whether the particle filter becomes stable or not after the initial confusion.

This task is an modification of the delayed alternation task [3, 23, 30]. A mouse or a rat has a sample trial in which it can only go to the reward arm before a test trial in this task. Moreover, the mouse/rat spends enough time to build a map of the environment in its hippocampus. This test examines whether the mouse/rat keeps the memory of the latest sample trial at the test trial. In Task 1, the robot can remember past events perfectly though it does not create a map. We can assume various algorithms for this or similar tasks in between the way of decision making of rodents

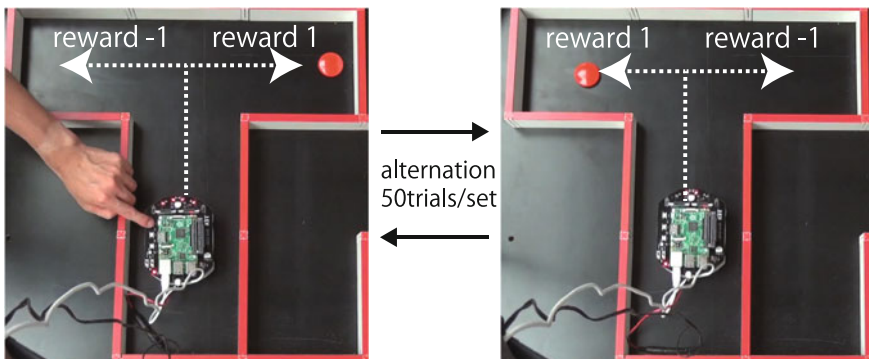


Fig. 4 Placement of the reward at Task 1

(a map and poor short-term memory) and the proposed method (no map but perfect memory). We will examine modifications of the proposed method as the series of algorithms in future.

5.6.2 Task 2: Learning a Relation Between the Initial State to the Reward Position

In this task, the initial position of the robot at the central arm is shifted to the side of the reward arm. As shown in Fig. 5, the robot is placed at the left side when the reward is placed on the left arm. When the reward is put at the right arm, the robot is set at the right. The arm on which the reward is placed is chosen randomly.

We show the statistics of sensor values on each initial position in Table 1. The sensor values are obtained in the experiment to be described in Sect. 5.7.2. Though the side on which the robot is placed can be distinguishable according to the sensor values, the robot must find the relation between the initial position and the reward arm without previous knowledge.

This task is an imitation of a visual discrimination task in a T-maze [20]. In the task, visual information is usually used for cues of discrimination. Since the robot does not have a camera, we use the values of sensors as substitute for visual cues.

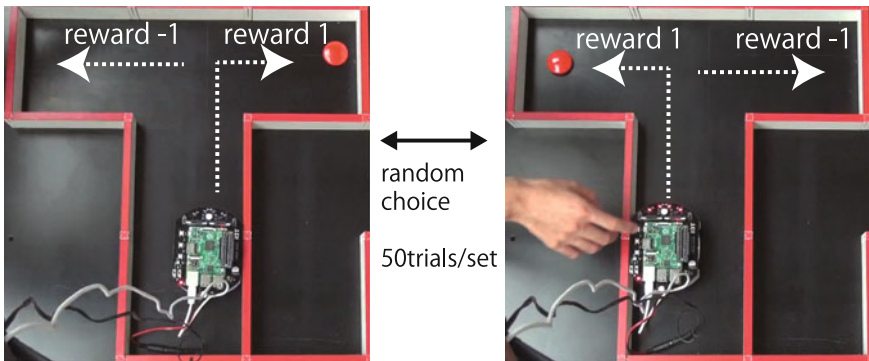


Fig. 5 Relation between the initial position and the place of the reward at Task 2

Table 1 Average sensor values and their standard deviations

Initial position at each trial	Average (standard deviation)			
	ℓ_1	ℓ_2	ℓ_3	ℓ_4
Left-side	125 (13)	399 (26)	3085 (61)	321 (14)
Right-side	247 (19)	2397 (206)	403 (37)	146 (18)

5.7 Result and Discussion

5.7.1 Effect of Retrospective Resetting

At first, we have investigated the effect of the retrospective resetting with Task 1. We have used 0.2 as the threshold α_{th} of a resetting. We have tried five sets of 50 consecutive trials with the retrospective resetting. Then we have done the other five sets with random resetting. The episode has been initialized at the beginning of each set. It means that the robot starts each set with an empty episode, while it can decide its action based on the experience of 49 trials at the 50th trial of each set. The parameter λ for the retrospective resetting is fixed to 4.

The result is shown in Fig. 6. We have calculated the success rates of every 10 trials in each set. The averages of the rates in the five sets are drawn in the figure with their standard deviation.

As shown in this figure, the particle filter with retrospective resetting bring the robot to the reward arm after 21st trials in every set. We can confirm that the robot can choose the reward arm with random resetting with 69.3 % after 21th trials. However, the result is much worse than that with retrospective resetting.

To quantify the difference, we have investigated the positions of particles on the episode through the experiment. In this task, similar sensor values are obtained in 4[step] cycle (n : a natural number), while the cycle of the task is 8[step]. Though the particles have a tendency to converge on $T - 4n$ [step] by the sensor updates, the particles that are not on $T - 8n$ [step] should be removed for consecutive successes. This problem is a kind of perceptual aliasing [26].

As shown in Table 2, 87 % of particles are in the eight step cycle with the retrospective resetting. This value is reduced to 64 % when random resetting are used. The particle filter requires several steps to converge particles on the eight step cycle from a random distribution of particles. However, a retrospective resetting can reduce this time for convergence by using the execution of sensor and motion updates without the actual motion of the robot.

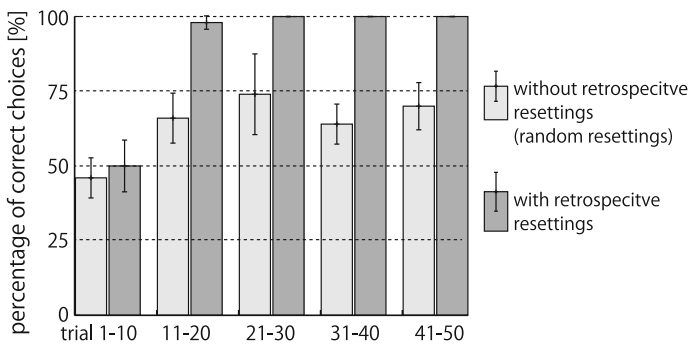


Fig. 6 Effect of retrospective resetting on Task 1

Table 2 Average weights of particles at specific cycles

Resetting algorithm	Average weights of particles at		
	$T - 8n[\text{step}]$	$T - (8n - 4)[\text{step}]$	Others
Random	0.64	0.20	0.15
Retrospective	0.87	0.07	0.05

5.7.2 Learning of Task 2 with the Identical Algorithm

Next, we have investigated the ability of the proposed method toward a task that does not have the cyclic pattern by using Task 2. We have examined Task 2 with the two values of the threshold $\alpha_{th} = 0.2$ and $\alpha_{th} = 0.5$ for the retrospective reset. Five sets of 50 trials are examined with each threshold.

The result is shown in Fig. 7. The success rate through the experiments is 70 % with $\alpha_{th} = 0.5$, while this value is 55 % with $\alpha_{th} = 0.2$.

In the case of this task, information of the last trial does not directly tell the place of the reward. To reflect the first observation of a trial, a reset at the observation works better for good decision since it removes bias of the distribution of particles. As shown in Table 3, a retrospective resetting occurs at 119 trials in the 250 trials when $\alpha_{th} = 0.5$. The difference of the success rates arises from the difference of frequencies of resetting.

Fig. 7 Results on Task 2

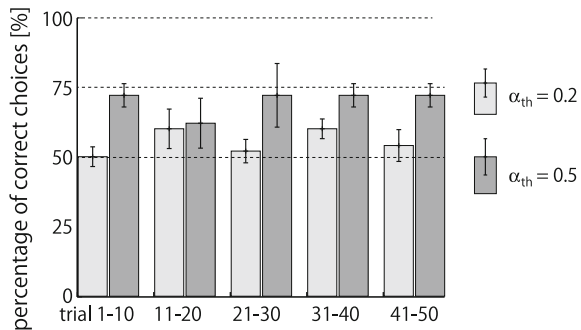
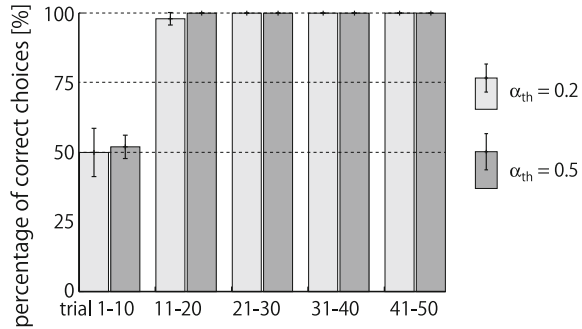


Table 3 Frequency of retrospective resetting at the first observation of a trial

α_{th}	# of trials	# of resetting
0.2	250	2 (0.8 %)
0.5	250	119 (47.6 %)

Fig. 8 Results on Task 1 with two thresholds

5.7.3 Task 1 with High Frequency of Retrospective Resetting

Finally, we investigate whether the frequent retrospective resetting give ill-effects in Task 1. We examine Task 1 again with the threshold $\alpha_{th} = 0.5$, by which the frequency of retrospective resetting increases. In the case of self-localization, frequent resetting make the convergence of particles slow.

The result is shown in Fig. 8 with the result obtained in Sect. 5.7.1 with $\alpha_{th} = 0.2$ and retrospective resetting. The result is not worse than the result with $\alpha_{th} = 0.2$. Since the distribution of particles after a retrospective resetting is made from several sensor updates and motion updates, the robot can choose appropriate actions just after a reset.

6 Conclusion

We have proposed a particle filter on episode. Following knowledge is obtained.

- The particle filter on episode has learned the rule of reward placement in Task 1 and 2.
- The retrospective reset exerts an important role on the particle filter. It enhances the success ratios from 69.3 to 100 % after 21st trial in Task 1, and from 55 to 70 % in Task 2.
- Frequent retrospective resetting do not give ill-effects for decision making according to the experimental results.

There are a number of studies that should be tackled. The next subjects will be as follows: (1) to examine the method with tasks that have longer cycles than the cycle of Task 1; and (2) to consider the case where the number of time steps of every trial vary due to odometry error or mischoice of actions.

A Appendix

A.1 The Character of the Range Sensor

We have measured the relation between sensor readings and distances from a sensor to a wall in the environment. The result is shown in Fig. 9, Note that sensor readings are easily shifted by some differences of conditions.

A.2 Actual Episode on the Experiment

Table 4 shows the first eight events of an experimental set, which is the first set of the experiment in Sect. 5.7.1. In the first trial, we set the reward one at the left arm and the robot obtained it. In the second trial, the robot did not obtain the reward placed at the right arm since it chose action “left” at $t = 5$.

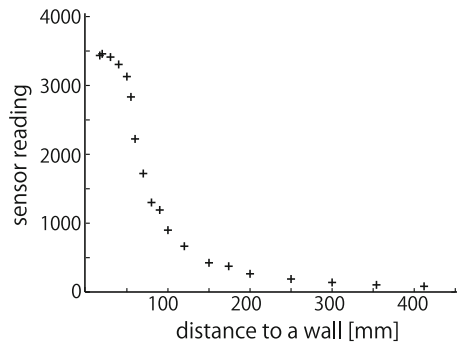


Fig. 9 Relation between distances and sensor readings

Table 4 A part of an episode

Trial	t	$o(t)$				$r(t)$	$a(t)$
		ℓ_1	ℓ_2	ℓ_3	ℓ_4		
1	0	111	769	1067	144	0	Fw
1	1	2617	965	3309	3023	0	Left
1	2	212	1405	524	208	0	Fw
1	3	496	3314	781	977	1	Stay
2	4	144	829	986	172	0	Fw
2	5	1956	534	2692	2580	0	Left
2	6	188	795	900	240	0	Fw
2	7	825	3222	1274	1413	-1	Stay

References

1. Barbieri, R., et al.: An analysis of hippocampal spatio-temporal representations using a Bayesian algorithm for neural spike train decoding. *IEEE Trans. Neural Syst. Rehabil. Eng.* **13**(2), 131–136 (2005)
2. Buzsáki, G., Moser, E.I.: Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nat. Neurosci.* **16**(2), 130–138 (2013)
3. Dudchenko, P.A.: An overview of the tasks used to test working memory in rodents. *Neurosci. Biobehav. Rev.* **28**(7), 699–709 (2004)
4. Fox, D., et al.: Monte Carlo localization: efficient position estimation for mobile robots. In: *Proceedings of AAAI*. pp. 343–349 (1999)
5. Franz, M.O., Mallot, H.A.: Biomimetic robot navigation. *Robot. Auton. Syst.* **30**, 133–153 (2000)
6. Hafting, T., et al.: Microstructure of a spatial map in the entorhinal cortex. *Nature* **436**, 801–806 (2005). Aug
7. Hargreaves, E.L., et al.: Major dissociation between medial and lateral entorhinal input to dorsal hippocampus. *Science* **308**, 1792–1794 (2005). June
8. Ito, H.T., et al.: A prefrontal-thalamo-hippocampal circuit for goal-directed spatial navigation. *Nature* **522**, 50–55 (2015)
9. Kitamura, T., et al.: Island cells control temporal association memory. *Science* **343**(6173), 896–901 (2014)
10. Lever, C., et al.: Boundary vector cells in the subiculum of the hippocampal formation. *J. Neurosci.* **29**(31), 9771–9777 (2009)
11. Milford, M., Schulz, R.: Principles of goal-directed spatial robot navigation in biomimetic models. *Philos. Trans. R. Soc. B* **369**(1665), 2013484 (2014)
12. Milford, M.J., Wyeth, G.F.: Mapping a suburb with a single camera using a biologically inspired SLAM system. *IEEE Trans. Robot. Autom.* **24**(5), 1038–1053 (2008)
13. Mnih, V., et al.: Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015)
14. Montemerlo, M.: FastSLAM: a factored solution to the simultaneous localization and mapping problem with unknown data association. Doctor Thesis, Carnegie Mellon University (2003)
15. Moser, E.I., Moser, M.B.: A metric for space. *Hippocampus* **18**(12), 1142–1156 (2008)
16. O’keefe, J., Dostrovsky, J.: The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* **34**(1), 171–175 (1971)
17. Ormonet, D., Sen, S.: Kernel-based reinforcement learning. *Mach. Learn.* **49**(2–3), 161–178 (2002)
18. Pastalkova, E., et al.: Internally generated cell assembly sequences in the rat Hippocampus. *Science* **321**(5894), 1322–1327 (2008)
19. Pfeiffer, B.E., Foster, D.J.: Hippocampal place-cell sequences depict future paths to remembered goals. *Nature* **497**, 74–79 (2013)
20. Shaw, C.L., et al.: The role of the medial prefrontal cortex in the acquisition, retention, and reversal of a tactile visuospatial conditional discrimination task. *Behav. Brain Res.* **236**, 94–101 (2013)
21. Solstad, T., et al.: Representation of geometric borders in the entorhinal cortex. *Science* **322**(5909), 1865–1868 (2008)
22. Spellman, T., et al.: Hippocampal-prefrontal input supports spatial encoding in working memory. *Nature* **522**, 309–314 (2015)
23. Suh, J.: Entorhinal cortex layer III input to the Hippocampus is crucial for temporal association memory. *Science* **334**(9), 1415–1420 (2011)
24. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge (1998)
25. Tesauro, G.: Temporal difference learning and TD-Gammon. *Commun. ACM* **38**(3), 58–68 (1995)

26. Thrun, S., et al.: Probabilistic Robotics. MIT Press (2005)
27. Ueda, R.: Generation of compensation behavior of autonomous robot for uncertainty of information with probabilistic flow control. *Adv. Robot.* **29**(11), 721–734 (2015)
28. Unemi, T., Saitoh, H.: Episode-based reinforcement learning—an instance-based approach for perceptual aliasing. In: *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 435–440 (1999)
29. Yamakawa, H.: Hippocampal formation mechanism will inspire frame generation for building an artificial general intelligence. In: *Artificial General Intelligence*, pp. 362–371 (2012)
30. Yamamoto, J., et al.: Successful execution of working memory linked to synchronized high-frequency gamma oscillations. *Cell* **157**, 845–857 (2014)

Part XII
Robot Design

Efficient Measurement of Fibre Orientation for Mapping Carbon Fibre Parts with a Robotic System

Morris Antonello, Matteo Munaro and Emanuele Menegatti

Abstract The strength of carbon fibre parts depends on the fibre arrangements all over them, but only manual and sparse checks are usually executed to assess their quality. Here, we present an automatic method for computing the fibre orientations in each part point and mapping them onto the 3D model of the part. This process is automated by a robot that moves the measurement sensor above the object to be scanned. Since this sensor needs to acquire multiple images of the same point with different illuminations for correctly estimating the fibre orientation, we developed algorithms for online image registration in presence of translational sensor motion. Moreover, we propose real-time methods for projection of the estimated orientation vectors to a 3D model. Experiments show that this software allows the accurate and fast mapping of carbon fibre parts by means of an industrial robot. Accuracy assessments report a measurement accuracy below 5°.

Keywords Carbon fibre parts · Quality control · Automatic inspection · Fibre orientation estimation · Image registration · Fibre mapping · 3D model · Draping simulation

M. Antonello (✉) · M. Munaro · E. Menegatti
Department of Information Engineering, University of Padova,
Via Ognissanti 72, 35129 Padova, Italy
e-mail: morris.antonello@dei.unipd.it
URL: <http://robotics.dei.unipd.it>

M. Munaro
e-mail: matteo.munaro@dei.unipd.it

E. Menegatti
e-mail: emg@dei.unipd.it

1 Introduction

In the automotive industry, there is a strong trend towards the use of carbon fibre parts. On the one hand, their use aims at reducing the weight of the car and on the other, those parts are an important design component. Carbon parts are not only relevant for lightweight sports cars, but also for the development of electrical vehicles that have a strong motivation to reduce weight in order to extend their operating range.

A typical production process is visualized in the top part of Fig. 1. Starting from a CAD drawing that also includes the single layers of material (ply book), the placing of the material on the part (draping) is simulated (draping simulation) to obtain process parameters. By executing this process a preform is produced that is then ready for further processing (e.g. injection moulding).

Draping simulation provides a prediction of the fibre angles on the 3D part, which is important, because the mechanical strength of structural parts depends on these fibre angles. In fact during the design of carbon fibre parts, much time is spent on optimizing the part for the expected loads that will later act on the part [10, 14]. An ideal process, illustrated in Fig. 1, would require to check the quality of the produced preform and provide direct feedback to the simulation for correcting the production process. However, this kind of loop is not feasible in current production systems, because

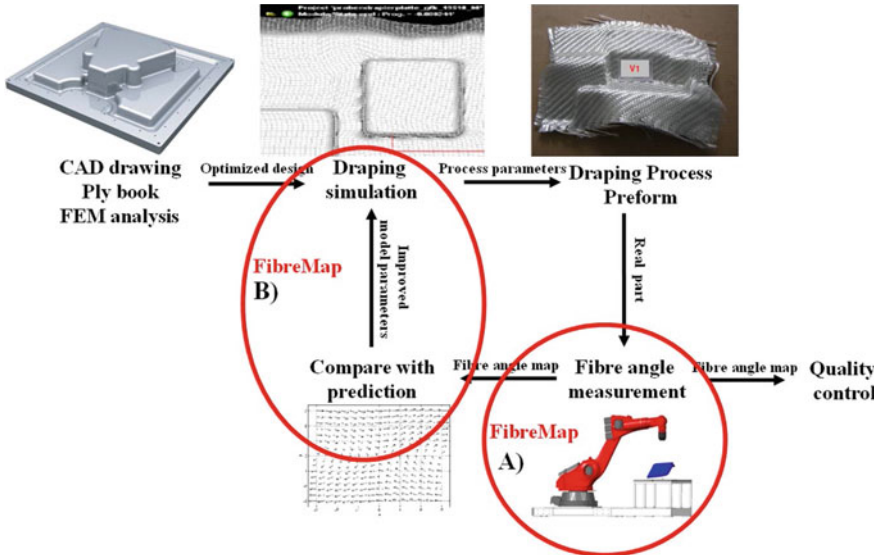


Fig. 1 Improved carbon parts production process as envisioned in the FibreMap project

- there is no method of acquiring dense fibre angle mappings for complex 3D parts. Measurements are often done manually, where accurate coordinate systems and reference lines are hard to establish and obtaining a dense mapping of the part is next to impossible.
- this lack of accurate fibre angle data is also hindering further development and improvement of draping simulation, because the models used inside the simulation tools would need to be tuned by comparing the predictions to reality.
- quality control is also limited, because manual measurements are insufficient and do not provide data and documentation that are suitable for high-quality part production as is required in the automotive industry.

The lack of technologies for accurate fibre angle measurement and prediction has several consequences that have a negative influence on the applicability of carbon fibre parts:

- Carbon fibre structural parts are designed to be light and strong. However, safety factors need to be considered to compensate for the lack of detailed knowledge about the material, and the mechanical properties, but also for the currently unknown differences between the part as designed and as produced. This leads to unnecessarily high safety factors (and thus weight), which could be reduced by 20 % if the fibre angles could be accurately predicted and measured.
- Aside from being lightweight, carbon fibre parts are often expected to have the typical carbon look, with the pattern of the carbon material still being visible. For these parts, it is also important to have the fibres aligned e.g. to the edges of the part, which currently can only be obtained through lengthy and time-consuming experimentation.

The FibreMap [16] project¹ aims at the development of an automatic quality control and feedback mechanism to improve draping of carbon fibres on complex parts. This mechanism could shorten process development times by 90 % and allow automatic 100 % quality control of fibre orientation.

1.1 Related Work

Most automated visual inspection systems for woven material and fabric are based on the analysis of their textural properties. This topic has been of interest in the past [7, 17] and is still of interest given the increasing use of such materials [20]. Nonetheless, the optical properties exhibited by this kind of material, like specular reflection and light absorption, make this analysis and, more specifically, the measurement of the fibre orientation relative to the part difficult [21]. Even if, on the one hand, diffuse lighting can relieve these issues, on the other, also the contrast of important textural features is reduced.

¹<http://www.fibremap.eu>—2013–2016.

The ongoing approaches are based on gradient vectors in a local neighborhood [13], directional evidence accumulation [5, 15], Gabor filters and steerable pyramids [4]. Even if they can estimate the local dominant orientations, they need the line-like structure to be visible and strongly rely on parameters like the size of the neighborhood, sub-window or kernel. In addition they are time-consuming. To address these issues more effectively and provide an orientation per pixel, the reflection behavior, more precisely the cone-shaped reflection model, has been used as a basis for the development of a sensor for the fast and accurate estimation of the dominant orientations in textures [24] without texture analysis. It can measure azimuth angles with a RMSE of 0.4° and polar angles with a RMSE of 1.0° . There exist also alternatives based on different technologies like thermography [8].

In our work, for the first time to the best of our knowledge, the sensor presented in [24] is mounted on an industrial robot, see Fig. 2a. This way, with the aims of inspecting parts larger than the field of view of the camera and automating the build-up process, the carbon fibre parts can be completely and quickly inspected. Before, only sparse and manual measurements were performed. Recently, also projects based on thermographic inspection like ThermoBot [1, 11] tackled similar problems by developing methods for automatic detection of defects in carbon fibre parts by means of a robotic system.

1.2 System Overview

The technology that is being developed in this project includes a new sensor system for robust detection of fibre orientation combined with a robotic system to scan complex parts. Using an industrial robotic arm, the sensor is moved relative to the inspected part and fibre angles are measured. After capturing sensor data of a part surface, fibre angles are calculated and virtually mapped back onto the surface of the

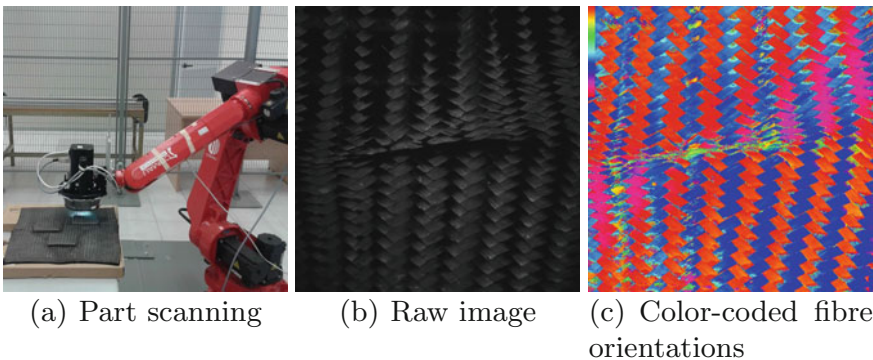


Fig. 2 Example of **a** scanning of a carbon fibre part, **b** a raw image acquired with the sensor and **c** the estimated fibre orientations coded with colors

CAD model. This back-projection leads to a fully 3D representation of the measured fibre angle data (Result of Part A in Fig. 1). The main contributions of this work are:

- an image registration method that allows to measure fibre angles while the sensor translates over planar surfaces;
- a mapping method that exploits the timestamped robot positions and allows to map the fibres to the 3D part model in real-time.

A flowchart of the main software modules of the mapping process is reported in Fig. 3a. From an architectural point of view, two multi-threaded applications, that exploit the high level of knowledge of the work-cell obtained as described in Sect. 2, have been developed. The former application implements the robot controller while the latter the fibre angle measurement system.

The former application continuously sends the current robot position and timestamp to the latter. In addition, it receives the trajectory from the path planner. To this end, a communication protocol has been defined. Each message contains the motion type, i.e. joint or linear interpolation, the robot position, i.e. in joint or Cartesian space. Then, it interpolates the intermediate positions and sends them to the robot, a Comau Smart5 SiX.

The latter application is synchronized in time with the former at startup and interfaces with the sensor. Likewise Photometric Stereo (PS), fibre orientations can be calculated by observing them when illuminated from different and known light sources. Thus, multiple images of the same point have to be captured in order to compute the orientation of that point. Experimentally, eight images proved to be enough [24]. Thus, this application continuously grabs raw images, as the one shown in Fig. 2b, and associates to each of them the corresponding robot and sensor position. Given that the robot is moving and the eight images must be referred to the same viewpoint for fibre angles to be properly computed, they are registered by exploiting the sensor positions, as described in Sect. 3. Then, for each pixel, an arrow, that is a 3D vector (u_x, u_y, u_z) representing the carbon fibre orientation, is calculated. Alternatively, the fibre orientations can be represented by azimuth and polar angles. An example of color-coded azimuth values for an image is shown in Fig. 2c. Finally, each image point and the corresponding arrow are projected in real time to a 3D model of the part, as detailed in Sect. 4.

In Sect. 5, we report experimental results and accuracy tests performed on one test case of the FibreMap project, the so-called *Three-Hills* visible in Fig. 2a. In Sect. 6, conclusions are drawn and future directions for research are identified.

2 System Calibration

The algorithms performing fibre angle computation and projection need to know the exact transformations between the reference systems listed here below and illustrated in Fig. 3b:

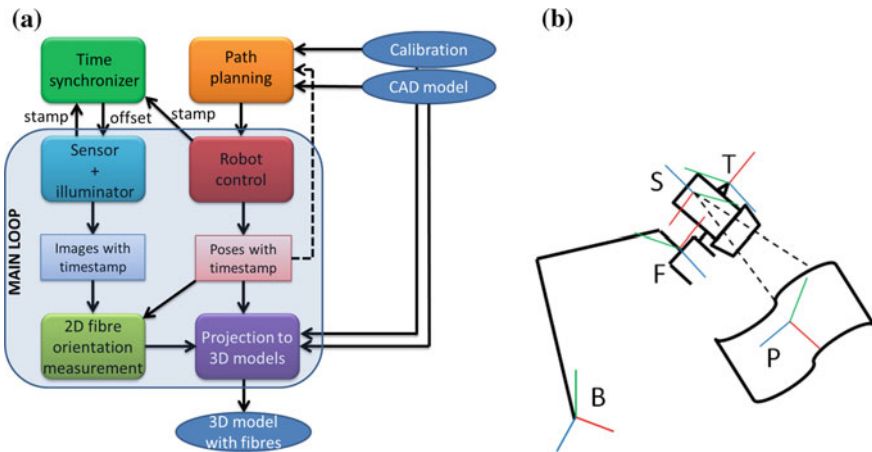


Fig. 3 System overview: **a** workflow of the fibre mapping software and **b** reference systems: calibration tip (T), part (P), robot base (B), robot flange (F) and sensor (S)

- the world triad (B): it coincides with the robot base;
- the user triad (P): it is defined on the work object, which could be the real part during the scanning, or the checkerboard when performing the hand-eye calibration;
- the flange triad (F): it is defined on the robot flange;
- the calibration tip triad (T): it is defined on the tip mounted on the robot flange;
- the sensor triad (S): it is defined on the sensor and the origin coincides with its optical center.

In particular, these algorithms need two pieces of information. The former is the position of the object to be scanned with respect to the robot, in other words the transformation between the part (P) and the robot base (B). The latter is the transformation between the robot flange (F) and the sensor (S). Both of them can be derived by means of three procedures: the calibration of the tip mounted on the robot flange, the calibration of the work object and the hand-eye calibration [23]. They are standard so their in-depth description can be omitted. The hand-eye calibration needs about 50 pictures of a checkerboard taken from different heights and angles. Low reprojection errors have been obtained by exploiting not only one but all of the eight images returned by the sensor at a time. If their average is considered, the reprojection error results of 1.30 px instead of 2.25 px for the intrinsics and of 1.96 px instead of 5.11 px for the hand-eye.

3 Image Registration for Accurate Fibre Orientation Measurement in Motion

Fibre angles are estimated from eight images, $I_i, i = 0 \dots 7$, taken under different illuminations. Given that their acquisition is performed in motion, each of them is taken from a different sensor pose, $P_i, i = 0 \dots 7$. Each pose P_i , derived from the procedures in Sect. 2, is relative to the Cartesian reference system of the part, triad P in Fig. 3. Fibre angles can be estimated only in the overlapping region O since it is in the field of view of the sensor from all the eight poses. For ease of understanding, an example with 4 images is reported in Fig. 4a. This overlap varies with the robot speed, the distance of the sensor to the part and the sensor field of view. To find O , the eight images must be registered with respect to a common reference pose, that can be one of the eight camera poses, e.g. P_3 . Feature-based algorithms, which find correspondences between image features such as points, lines and contours, are unreliable because of the repetitive and regular texture and the changes in illumination between images. Even if SIFT [6] is one of the best features, often more than 60% of the feature matches are wrong in this case making the alignment with RANSAC [9] risky. Instead, the proposed approach is based on the transformations between pairs of poses. Indeed, since each of them is known, the relative poses, $RT_i, i = 0 \dots 7$, describing the transformations between the poses $P_i, i \neq 3$ and the reference P_3 can be computed. Given that the robot moves in a zig-zag fashion, its movements could be approximated as subsequent translations. In particular, for each pair of poses P_i and P_3 , the translation vector in world coordinates (t_x^w, t_y^w, t_z^w) is the translational component of the roto-translation matrix RT_i . This vector is then converted to image coordinates (t_x^i, t_y^i) , i.e. pixels, so as to apply it to each image $I_i, i \neq 3$ to perform the registration.

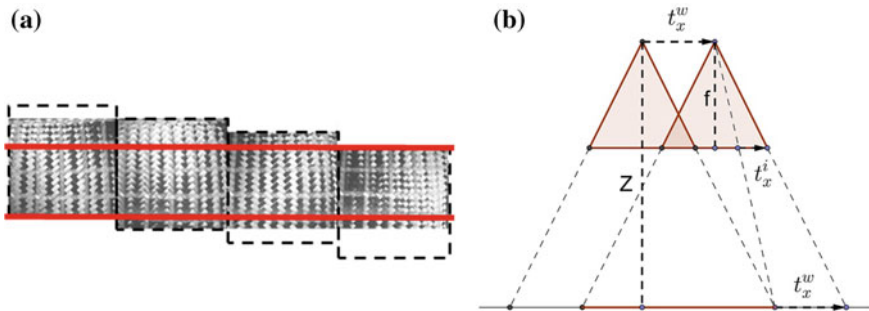


Fig. 4 Image registration. **a** Four images I_i acquired with different light sources and aligned to the reference frame of the second. The overlapping region O is bounded by the red box. **b** A sample translation along the x axis in world (t_x^w) and pixel (t_x^i) coordinates. Also the sensor height Z and focal length f are shown

The relationship between these translation vectors can be derived from the well-known pinhole camera model [3] describing the mapping from 3D points in the world to 2D points in the image plane of the camera:

$$x_i = f_x \frac{X}{Z} + c_x, y_i = f_y \frac{Y}{Z} + c_y, \quad (1)$$

where f_x and f_y are the focal lengths, c_x and c_y the optical center coordinates and Z the distance of the sensor to the part. The resulting proportions are:

$$t_x^i : t_x^w = f_x : Z, t_y^i : t_y^w = f_y : Z \quad (2)$$

An illustration of the relation between world and image translation is reported in Fig. 4b.

4 Efficient Rasterization for Real-Time Arrow Mapping to 3D Models

The carbon fibre mapping to a 3D model consists in the ray casting of each camera image point to that model. This process can be implemented with the z-buffering algorithm [22] which stores in the z-buffer two pieces of information for each camera image point. Let P_i be a generic camera image point and r_i the corresponding ray passing through P_i and the focal point. The former piece of information is the intersection point between P_i and the 3D model while the latter is the associated distance between P_i and the intersection point. The algorithm consists in the forward-projection of each 3D triangle to the image space. For each P_i belonging to the 2D triangle, the distance is derived and compared against the z-buffer value. If the distance is less than the previously recorded value, the location is overwritten. A speed-up is possible by optimizing the derivation of the list of points belonging to the 2D triangle, the so-called triangle rasterization. This can be done straightforwardly by approximating the 2D triangle with the smallest rectangle containing it [12].

This choice has three main advantages:

- efficiency: the complexity is $O(Pp)$ where P is the number of 3D triangles and p the average number of pixels in each 2D triangle. Since the product Pp is insensitive to the number of 3D triangles, the performance is not sensitive to the number of 3D triangles. Thus with decreasing memory costs and increasing number of 3D triangles it becomes more cost-effective;
- simplicity: the lack of additional data-structures implies decreasing memory costs;
- generality: given that it operates in image space and does not require the objects to be 3D triangles it handles both low and high-resolution meshes.

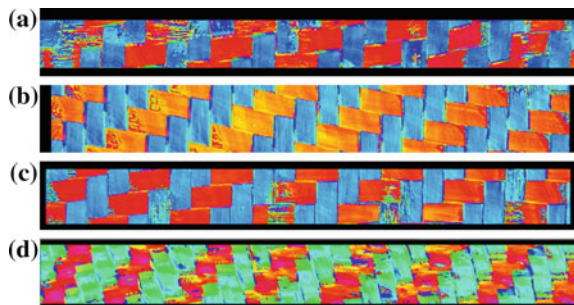
5 Experiments and Assessments

Ten *Three-Hills* preforms differing in the type of material have been fully scanned with the proposed system. For these samples, the material was woven or not, coated with EP-binder or not and differed in the filament count of each bundle, e.g. 6 K or 24 K, and in the fibre orientation, e.g. $\pm 45^\circ$ or $0/90^\circ$. The surface of some of them has been wiped with the chalk along some lines in order to make it possible to visually check the alignment of the registered raw images. The path followed by the robot was manually generated with a zig-zag pattern, thus consisting in a series of translations along the x-axis and y-axis. In Fig. 5a and b, we show sample results of fibre computation when the sensor was moving along the y and x image axis, respectively. These images represent the fibre azimuth angles and the black parts refer to pixels that do not belong to the overlap O . Translations across both axes at the same time have been tested too, as reported in Fig. 5c. Fibres are badly estimated only along the chalk lines since, of course, the reflection properties of the material are altered by the chalk. In Fig. 5d, the azimuth images obtained by inverting on purpose the translation signs are reported in order to show an example of bad registration.

To quantitatively evaluate performances, the accuracy of the fibre angle measurement system has been measured. The system accuracy depends on the accuracy of the robot, calibrations, sensor and registration. In the following, two of the tests we performed are reported. They measure the accuracy of the fibre angles both in a static and in a dynamic configuration and share a common setup: as shown in Fig. 6a, a bundle of fibres is marked on one of the preforms with some scotch tape around it. This way, as shown in Fig. 6b, the orientations measured by the sensor system can be easily retrieved and compared with the ground truth, i.e. the hand-measured orientation, 93.0° .

The former test provides a measure of the sensor accuracy in a static situation. It consists in calculating the fibre angles with respect to the part reference frame with the sensor placed in 10 different points of view. In Fig. 7a, the sensor poses framing the bundle are reported. The average measure over the 10 points of view is 97.2° with a standard deviation of 0.8° . The error is 4.2° .

Fig. 5 Azimuth images computed with the robot in motion: **a** along y image axis, **b** x, **c** both. **d** Example of bad registration, the translation signs have been deliberately inverted



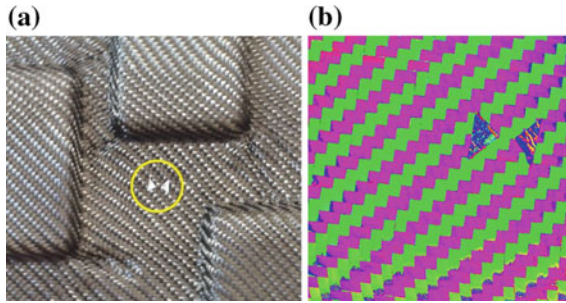


Fig. 6 **a** A bundle of fibres is marked on one of the preforms with some scotch tape. The hand-measured orientation in that point is 93.0° and it is used as ground truth. **b** This bundle of fibres can be easily retrieved in the images because the scotch tape is visible also in the azimuth images and in the 3D model obtained after fibre projection

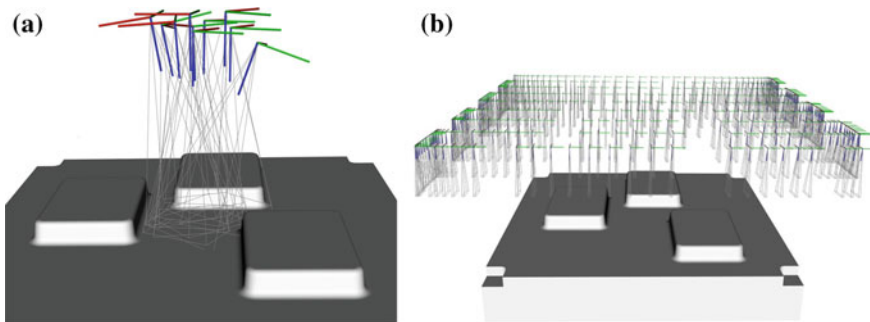


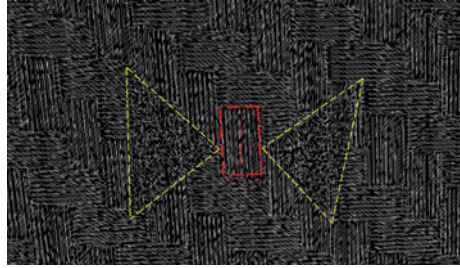
Fig. 7 **a** The 10 sensor poses used for framing the selected bundle of fibres. **b** A subset of the sensor poses used for a full scan of the part

The latter test provides a measure of the system accuracy in a dynamic situation, i.e. when the robot moves so as to fully cover the 3D model of the scanned preform. This test is different from the previous one because, on the one hand, the robot moves and, on the other, the ground truth is compared with the fibre angle after the registration and projection phases. In Fig. 7b, a subset of the sensor poses is reported. In Fig. 8, the fibres mapped to the part are shown. The average measure over the 10 points of view is 93.5° with a standard deviation of 1.2° . The error is 0.5° .

In both tests, the average errors are lower than what is usually required by the industrial standards (in the order of 5°) even if the images have been taken from very different points of view. The accuracy in the dynamic case is higher than in the static case because the sensor has been kept more perpendicular to the part. This constraint is easily satisfied when scanning planar parts.

The algorithms described in this paper have been implemented in C++ using the following open source libraries: PCL [19], OpenCV [2] and ROS [18]. The proposed method has proven to be efficient enough for real time scanning. In the following,

Fig. 8 Fibre arrows after the projection to a 3D model of the part. The scotch tape is highlighted in yellow and the selected bundle of fibres is highlighted in red



the runtime analysis on a desktop PC with Intel i7-4770K CPU @ 3.5 GHz, 16 GB RAM, 7200 rpm hard disk and Windows 7 is reported:

- image acquisition (800×100 px): 50 fps;
- image acquisition and fibre computation (800×100 px): 30 fps;
- image acquisition, fibre computation (800×100 px) and mapping (200×25 px): 20 fps.

These frame rates allow to perform a complete scan of a *Three-Hills* part (400×400 mm) in a zig-zag fashion with an average speed of 0.02 m/s in less than 5 min. The fibre-mapped model obtained from a complete scan is shown in Fig. 9a, while a detail of this model is reported in Fig. 9b, where the fibre pattern is clearly visible. The denser regions are due to overlaps generated from different passes of the robot over the part.

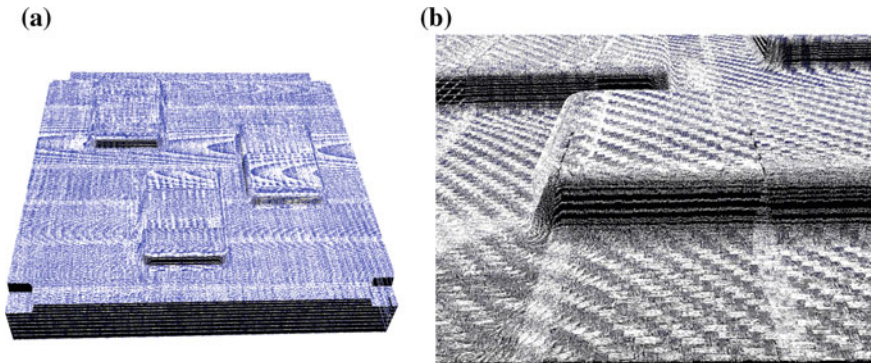


Fig. 9 **a** Fibre-mapped model obtained from a complete scan of a *Three-Hills* preform. **b** Zoom on the fibre-mapped model: the *denser regions* are due to overlaps generated from different passes of the robot over the part

6 Conclusions and Future Work

In this work, we presented a robotic system for the automatic inspection of carbon fibre parts. In particular, we described efficient methods for densely estimating fibre orientation and map this information to a 3D model of the part, that can be used for either quality inspection or for improving draping simulation. Unlike previous work, the proposed method allows to estimate fibre orientation even when the sensor is translating over the part by exploiting an efficient image registration algorithm based on robot pose information. Also the algorithm developed for projecting fibre arrows to a 3D model of the part proved to work in real time, so that the overall frame rate is of about 20 fps. The system accuracy has been evaluated with the robot in motion and an average error below 5° has been achieved. The inspected preforms are made by different fabrics typically used in serial production. As future work, we plan to integrate an automatic path planner for scanning complex parts. For this purpose, the registration will be extended so as to work with generic robot movements, other than translations. We will also develop algorithms for optimally merging arrows estimated from different camera poses but projected to the same portion of the surface by promoting views where the sensor is more perpendicular to the part.

Acknowledgements This research has been funded by the European Union's 7th Framework program ICT under grant agreement No. 608768, *FibreMap* project.

References

1. Antonello, M., Ghidoni, S., Menegatti, E.: Autonomous robotic system for thermographic detection of defects in upper layers of carbon fiber reinforced polymers. In: IEEE International Conference on Automation Science and Engineering (CASE) (2015)
2. Bradski, G.: The opencv library. *Doct. Dobbs J.* **25**(11), 120–126 (2000)
3. Bradski, G., Kaehler, A.: *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc. (2008)
4. Chang, J., Fisher, J.W.: Analysis of orientation and scale in smoothly varying textures. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 881–888. IEEE (2009)
5. Chaudhuri, B.B., Kundu, P., Sarkar, N.: Detection and gradation of oriented texture. *Pattern Recogn. Lett.* **14**(2), 147–153 (1993)
6. Cheung, W., Hamarneh, G.: Sift: dimensional scale invariant feature transform. *IEEE Trans. Image Process.* **18**(9), 2012–2021 (2009)
7. Cohen, F.S., Fan, Z., Attali, S.: Automated inspection of textile fabrics using textural models. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(8), 803–808 (1991)
8. Fernandes, H.-C., Maldague, X.: Fiber orientation assessment in complex shaped parts reinforced with carbon fiber using infrared thermography. *Quant. Infrared Thermogr. J.* **12**(1), 64–79 (2015)
9. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (1981)
10. Fuhr, J.-P., Baumann, J., Härtel, E., Middendorf, P., Feindler, N.: Effects of in-plane waviness on the properties of carbon composites—experimental and numerical analysis. *CompTest 2013-Book of Abstracts*, p. 61 (2013)

11. Ghidoni, S., Antonello, M., Nanni, L., Menegatti, E.: A knowledge-based approach to crack detection in thermographic images. In: 13th International Conference on Intelligent Autonomous Systems (IAS-13). In Press (2014)
12. Hughes, J.E., Foley, J.D.: *Computer Graphics: Principles and Practice*. Pearson Education (2013)
13. Kass, M., Witkin, A.: Analyzing oriented patterns. *Comput. Vis. Graph. Image Process.* **37**(3), 362–385 (1987)
14. Kim, J.-W., Lee, D.-G.: Effect of fiber orientation and fiber contents on the tensile strength in fiber-reinforced composites. *J. Nanosci. Nanotechnol.* **10**(5), 3650–3653 (2010)
15. Miene, A., Herrmann, A.S., Göttinger, M.: Quality assurance by digital image analysis for the preforming and draping process of dry carbon fiber material. In: SAMPE Europe Conference, Paris (2008)
16. Munaro, M., Antonello, M., Moro, M., Ferrari, C., Pagello, E., Menegatti, E.: Fibremap: automatic mapping of fibre orientation for draping of carbon fibre parts. In: IAS-13 Workshop on ROS-Industrial in European Research Projects, Padova, Italy, pp. 272–275, July 2014
17. Ozdemir, S., Baykut, A., Meylani, R., Ercil, A., Ertuzun, A.: Comparative evaluation of texture analysis algorithms for defect inspection of textile products. In: International Conference on Pattern Recognition, vol. 2, pp. 1738–1738. IEEE Computer Society (1998)
18. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA Workshop on Open Source Software, vol. 3, p. 5 (2009)
19. Rusu, R.B., Cousins, S.: 3d is here: point cloud library (pcl). In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 1–4. IEEE (2011)
20. Schmitt, R., Mersmann, C., Schoenberg, A.: Machine vision industrialising the textile-based frp production. In: 6th International Symposium on Image and Signal Processing and Analysis (ISPA), pp. 260–264. IEEE (2009)
21. Shi, L., Wu, S.: Automatic fiber orientation detection for sewed carbon fibers. *Tsinghua Sci. Technol.* **12**(4), 447–452 (2007)
22. Theoharis, T.: *Algorithms for Parallel Polygon Rendering*, vol. 373. Springer (1989)
23. Tsai, R.Y., Lenz, R.K.: A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Trans. Robot. Autom.* **5**(3), 345–358 (1989)
24. Zambal, S., Palfinger, W., Stger, M., Eitzinger, C.: Accurate fibre orientation measurement for carbon fibre surfaces. *Pattern Recogn.* (2014)

A Honeycomb Artifacts Removal and Super Resolution Method for Fiber-Optic Images

Zhong Zheng, Bin Cai, Jieting Kou, Wei Liu and Zengfu Wang

Abstract The special fiber bundle image has characteristics of anti-radiation and resistance to high temperature, thus can be used to observe nuclear environment. But there are honeycomb artifacts in fiber bundle image, and the resolution of image is low, which greatly affects the image quality. This paper propose a method to segment optical fiber cladding and fiber center. And an improved non local means (NLM) algorithm is used to denoise and repair the fiber bundle images. The algorithm can remove the honeycomb artifacts and greatly enhance the image resolution. The test results show that the algorithm is effective.

Keywords Fiber bundle image · Honeycomb artifacts · Super resolution

1 Introduction

Nuclear fusion chamber is an extreme environment, which with strong radiation, high temperature, strong magnetic field and high vacuum and so on. For instance, in the International Thermonuclear Experimental Reactor (ITER), the maximum radiation can reach 470 Gy/h and the temperature is as high as 200 °C, the intensity of the magnetic field can reach 10 T, and the vacuum degree reaches 10^{-6} Pa. Usually the camera equipment in strong radiation and high temperature environment will soon be invalid, it can't work stably for a long time. Therefore, it is very important to develop an imaging device with anti radiation and high temperature resistance, and can work in strong magnetic field, high vacuum and complex geometry structure. Fortunately, the optical fiber, which is made of a doped OH or fluorine quartz glass material, is not afraid of high temperature and radiation. When using high temperature resistant glue them into a fiber bundle, and wrapped the fiber bundle in a layer of Teflon, it can be resistant to high temperature of 300 °C,

Z. Zheng · B. Cai · J. Kou · W. Liu · Z. Wang (✉)
Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, China
e-mail: zfwang@ustc.edu.cn

and even being irradiated by 10^5 Gy, the imaging performance of the optical fiber is not significantly affected.

Specially made optical fiber imaging beams can solve the problem of radiation resistance and high temperature resistance, but it has also brought some other problems: in addition to the technical challenges, such as low coupling efficiency and optical crosstalk, the disadvantage of using fiber bundles, in terms of image quality, is that a non-continuous honeycomb-like artifact degrades the lateral image resolution in the 2-D image plane between individual fibers. This pixelation noise induces to decrease the resolution compared to the resolving power of its single core and prevent from analyzing the continuous structure of target sample.

In recent years, many methods are proposed to remove the bundle pattern while preserving image content. Gobel et al. [1], Dubaj et al. [2], Oh et al. [3], Winter et al. [4] and Rupp et al. [5] use a series of time-domain filter, include the mean, gaussian, round, star filter. Dickens et al. [6] and Suter et al. [7] eliminate the periodic cladding edge through a circular or rectangular band-reject filter in frequency-domain. Han et al. [8] firstly applied a histogram equalization process, and then performed a weighted-averaged Gaussian smoothing filter to the original image. A more recent study proposed by Han et al. [9] is to find the local peak in the frequency domain, and use the band-reject filter to remove the peak. Ford et al. [10] acquired two widefield images under uniform and structured illumination, and then numerically processed to reject out-of-focus background. However, this method is cumbersome, the camera parameters need to be calibrated every time the camera is used. Lee and Han [11, 12] proposed a method from the viewpoint of physical realization by removing the pixelation pattern from the image, where the fiber imaging system is caused to move laterally with high-precision linear translation stages. However, it is difficult to shift the imaging system by the radius of a core in practical application. Cheon et al. [13] introduced a spatial compound imaging method to overcome this pixelation artifact. In order to remove or interpolate the periodically missing regions between imaging fiber cores, it uses multiple frames taken with small deviation of position, and directly copies the useful information to the reference image after alignment. The above methods focus on the design of a filter or a certain precision mechanical mechanism to improve the quality of the optical fiber image. The existing filter method, which is used to remove the impact of the cladding, will bring some blurring effect. And using the method of precision mechanical mechanism, the use of the observation device is limited, and the cost is expensive. Therefore, in order to meet the requirements of nuclear environment observation mission, it is critical to have an efficient method to enhance the optical fiber image.

According to the previous analysis, this paper proposes a method for fiber optic image enhancement based on the image inpainting algorithm. Firstly, according to the characteristics of optical fiber bundle imaging, we use a simple method to find the fiber center pixels. Then the interpolation of the periodically missing regions between imaging fiber cores is completed based on the fiber center pixels by using bilinear interpolation algorithm, where the result is named as reference image. Finally, we enhance the reference image by using an improved non-local means

(NLM) denoising algorithm [14], which we previously proposed. Experimental results show that compared with other state-of-art ones, our proposed optical fiber image enhancement method has a greater improvement in the image resolution and visibility.

2 Methods

2.1 Determining Fiber Center Pixels

The typical optical fiber image is shown in Fig. 1, due to the existence of the optical fiber cladding around each fiber core, a non-continuous honeycomb-like artifact degrades the lateral image resolution in the 2-D image plane between individual fibers.

In general, the fiber core has a better light transmission ability than the cladding region, and the value of fiber center pixels are similar. So we make use of the characteristics of the fiber optical imaging to segment the fiber center pixels and the cladding region. Firstly, filtering the image by median filter method. Then, by comparing the pixel value between the original image and the filtering result, it is easy to divide the center pixels and the cladding ones, because the pixel value in the original image is larger than the filtering one during the fiber core region, while the cladding region is the opposite. That is, any pixels in the images of the optical fiber.

$I(x, y) \geq I_{\text{median}}(x, y)$, if (x, y) is the fiber center pixels; and $I(x, y) < I_{\text{median}}(x, y)$, if pixel (x, y) belongs to the cladding region, where $I(x, y)$ is the gray-scale value of pixel (x, y) in the original optical fiber image, and $I_{\text{median}}(x, y)$ is the value of pixel (x, y) in the median filtering result. If there is a black spot in the scene with a single fiber core scale, it may be misrecognition. However, this does not affect the final recovery result, because this situation is very rare in our application scenarios.

Fig. 1 Typical optical fiber image



Fig. 2 Segmentation results

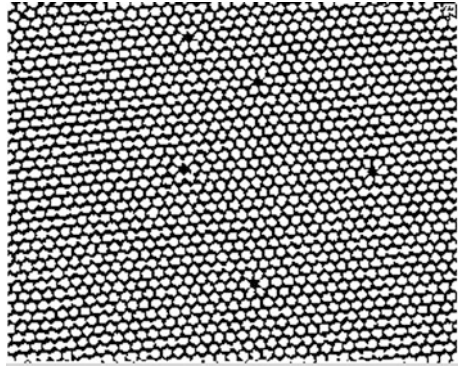


Fig. 3 The (R, P) neighborhood type used to $LBPM_{R,P}$ operator: central pixel and its $P = 8$ neighbors on circle of radius R

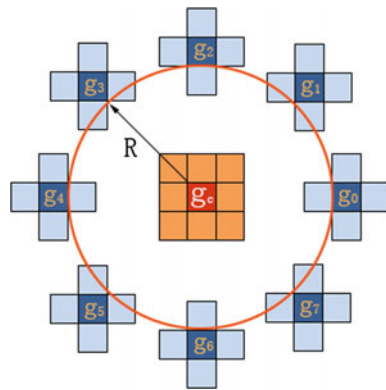


Figure 2 shows the segmentation results, where white area corresponding to the fiber center pixels, and black area corresponding to the cladding region or invalid pixels. Invalid pixels are caused by the broken fiber, so the method can extract the invalid pixels at the same time (Figs. 3 and 4).

2.2 Optical Fiber Image Inpainting

If we take the cladding region and the invalid pixels as the missing area, and inpaint the image using bilinear interpolation, we can remove the non-continuous honeycomb-like artifact. However, there is a lot of noise in the inpainting results, and image resolution is not enough. Hence, we use an improved NLM algorithm which we previously proposed [14] to enhance the inpainting results.

Non-local means (NLM) algorithm is a kind of image denoising and super-resolution algorithm for images with repetitive textures. It searches similar patches and uses them in a weighted average, which the weights depend on the

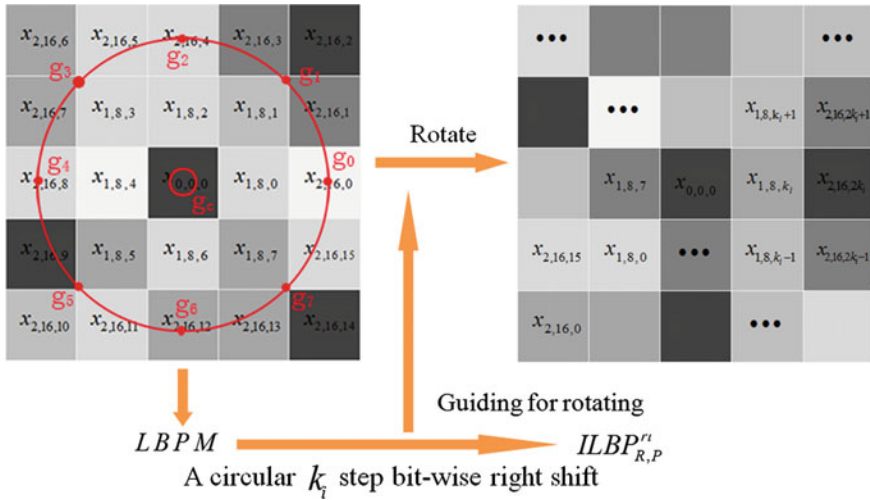


Fig. 4 Illustration of patch rotation

amount of similarity. Because of the missing regions between imaging fiber cores, it is difficult to realize the block matching in the original optical fiber image. So, our block matching is completed based on the bilinear interpolation results. Moreover, NLM cannot handle the large rotation. if we rotate the block several times before matching, and select the most similar one for the given block, there will be greatly increasing the amount of calculation. Therefore, we use an improved LBP operator to obtain the dominant orientation, and rotate the patch based on the dominant orientation before matching. Thanks to this, our similarity measure is rotation-invariant and noise-resistant. Here are the details:

Suppose $g = \{g(i)|i \in \Omega\}$ is the bilinear interpolation results, where Ω represents the image area. The enhanced image is represented as \hat{f} . $N(i)$ denotes a square neighborhood of fixed size and centered at pixel i , $N(j)$ centered at pixel j is the neighborhood of patch $N(i)$.

- (1) The LBP operator $ILBP_{R,P}^{ri}$ of pixel i can be computed as follows:

$$ILBP_{R,P}^{ri} = \min\{ROR(LBPM_{R,P}, k), k = 0, \dots, P - 1\} \tag{1}$$

$$LBPM_{R,P} = \sum_{p=0}^{P-1} s(\overline{g_p} - \overline{g_c})2^p \tag{2}$$

where $s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$, g_c is the gray-level value of pixel i . g_p is the neighbor of g_c . P is the sample number of g_p , R is the radius of the neighborhood. $\overline{g_c}$ is the means of g_c and its 8-connected. $\overline{g_p}$ is the means of g_p and its 4-connected

(see in Fig. 1). Due to this strategy, the $LBPM_{R,P}$ is robust to noise, it can also keep the difference between the neighborhood points even the radius is small ($R = 1$). $ROR(\bullet, k)$ performs a circular k -step bit-wise right shift on $LBPM_{R,P}$, so, the $ILBP_{R,P}^{ri}$ is rotation-invariant.

- (2) Obtain the rotated patch $N'(i)$ from $N(i)$
 Suppose the size of $N(i)$ and $N(j)$ is $(2R + 1) \times (2R + 1)$. We can obtain $ILBP_{R,P}^{ri}$ of pixel i and j with radius R , neighborhood sample points $P = 8$ using Eqs. (1) and (2). We can also obtain the circular step k_i bit-wise right shift from $LBPM_{R,P}$ to $ILBP_{R,P}^{ri}$ of pixel i , and the circular step k_j bit-wise right shift from $LBPM_{R,P}$ to $ILBP_{R,P}^{ri}$ of pixel j . So, the rotated patch $N'(i)$ from $N(i)$ is then computed by

$$\begin{aligned}
 N(i) &= \{ \mathbf{x}_{r,8r}, r = 0, \dots, R \} \\
 &= \{ [\mathbf{x}_{r,8r,0}, \dots, \mathbf{x}_{r,8r,k_i-1}, \mathbf{x}_{r,8r,k_i}, \dots, \mathbf{x}_{r,8r,8r-1}]^T \} \\
 N'(i) &= \{ \mathbf{x}'_{r,8r}, r = 0, \dots, R \} \\
 &= \{ [\mathbf{x}_{r,8r,k_i}, \dots, \mathbf{x}_{r,8r,8r-1}, \mathbf{x}_{r,8r,0}, \dots, \mathbf{x}_{r,8r,k_i-1}]^T \}
 \end{aligned}$$

where $\mathbf{x}_{r,8r}$ are the neighbors of pixel i on radius r , and $\mathbf{x}'_{r,8r}$ is the result by a circular k_i step bit-wise right shift from $\mathbf{x}_{r,8r}$ (see in Fig. 2). We can also obtain $N'(j)$ rotated from $N(j)$ using the same strategy.

- (3) Enhance the inpainting results
 The enhanced image \hat{f} can be computed as follows:

$$\hat{f}(i) = \frac{\sum_{j \in I} w(i,j)g(j)}{\sum_{j \in I} w(i,j)} \tag{3}$$

$$w(i,j) = \exp\left(-\frac{d(i,j)}{h^2}\right)$$

$$d(i,j) = \|N'(i) - N'(j)\|_{2,a}^2$$

where a is the standard deviation of the Gauss function, $d(i,j)$ is the distance between patches, I is the neighborhood pixel of pixel i . And the family of weights $w(i,j)$ depend on the similarity between the pixels i and j .

3 Experiments and Discussion

The proposed algorithm is tested on a variety of images, the experimental results convincingly demonstrate the effectiveness of our method. Figure 5 shows the comparison between the proposed algorithm and the classical time domain [8] and the frequency [13] algorithm. The method in [8] combined histogram equalization and gaussian spatial smoothing. However, gauss filter will cause image blur, and cannot remove the invalid pixels. The algorithm in [13] also cause a strong blurring effect. However, the result by our proposed method has better visual effect, the image contrast is higher, besides, the edge and the detail information are preserved better and more clearly. Further more, the invalid pixels have already removed.

Figure 6 shows the image and processing results of the fiber optical fiber image captured by our nuclear environment telerobot. We use our proposed method to eliminate the honeycomb patterns in fiber bundle image. The results are basically similar to the images captured by general camera, and can meet the requirements of nuclear environment observation mission and the following image processing needs.

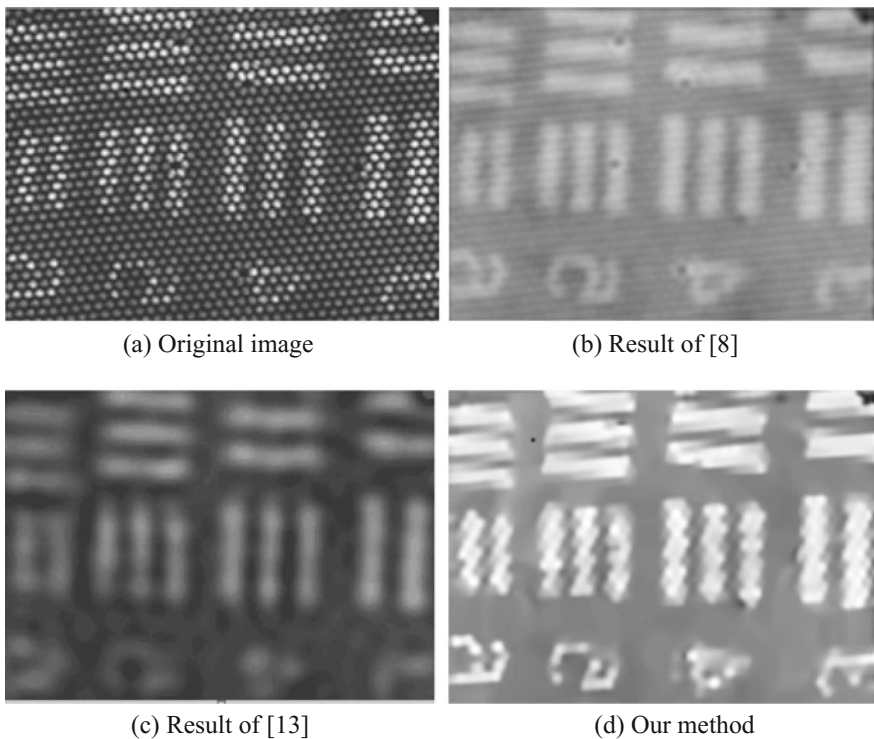


Fig. 5 Comparison of different method

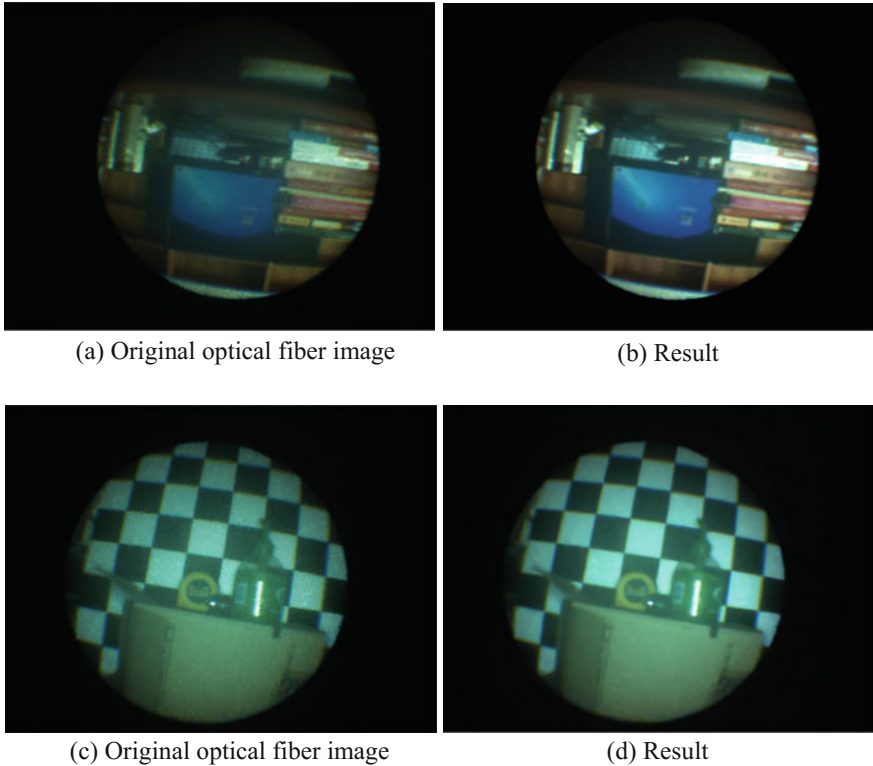


Fig. 6 Results of target images

4 Conclusion

We have proposed a method for fiber optic image enhancement algorithm based on the image inpainting algorithm. Compared with some state-of-art algorithm, our proposed method is more effective in removing the pixelation artifact. And it has already applied in the observation system of the nuclear environment telerobot in the International Thermonuclear Experimental Reactor (ITER).

This algorithm can be extended to the case of continuous frames, the real scenes correspond to optical cladding in different frames are not same. Through the fusion of multi frame information, the performance of the algorithm can be further improved.

Acknowledgements This research work was supported by the China National Magnetic-Confinement Fusion Energy R&D Program (ITER) and funded by the National Science and Technology Major Project of the Ministry of Science and Technology of China (No. 2012GB102007) and the National Natural Science Foundation of China (NSFC) under Grant No. 61503361. The authors wish to thank the research group for their great contributions to the presented research. The views and opinions expressed in this paper are the sole responsibility of the authors.

References

1. Göbel, W., Kerr, J.N.D., Nimmerjahn, A., Helmchen, F.: Miniaturized two-photon microscope based on a flexible coherent fiber bundle and a gradient-index lens objective. *Opt. Lett.* **29**(21), 2521–2523 (2004)
2. Dubaj, V., Mazzolini, A., Wood, A., Harris, M.: Optic fibre bundle contact imaging probe employing a laser scanning confocal microscope. *J. Microsc.* **207**, 108–117 (2002)
3. Oh, W.Y., Bouma, B.E., Iftimia, N., Yelin, R., Tearney, G.J.: Spectrally-modulated full-field optical coherence microscopy for ultrahigh-resolution endoscopic imaging. *Opt. Express* **14**, 8675–8684 (2006)
4. Winter, C., Rupp, S., Elter, M., Münzenmayer, C., Gerhäuser, H., Wittenberg, T.: Automatic adaptive enhancement for images obtained with fiberoptic endoscopes. *IEEE Trans. Biomed. Eng.* **53**, 2035–2046 (2006)
5. Rupp, S., Winter, C., Elter, M.: Evaluation of spatial interpolation strategies for the removal of comb-structure in fiber-optic images. In: 31st Annual International Conference of the IEEE EMBS, 2009, pp. 3677–3680
6. Dickens, M.M., Houlne, M.P., Mitra, S., Bornhop, D.J.: Soft computing method for the removal of pixelation in microendoscopic images. *Proc. SPIE* **3165**, 186–194 (1997)
7. Suter, M., Reinhardt, J., Montague, P., Taft, P., Lee, J., Zabner, J., McLennan, G.: Bronchoscopic imaging of pulmonary mucosal vasculature responses to inflammatory mediators. *J. Biomed. Opt.* **10**(3), 034013–0340135 (2005)
8. Han, J.-H., Lee, J.H., Kang, J.U.: Pixelation effect removal from fiber bundle probe based optical coherence tomography imaging. *Opt. Express* **18**, 7427–7439 (2010)
9. Han, J.-H., Yoon, S.M.: Depixelation of coherent fiber bundle endoscopy based on learning patterns of image prior. *Opt. Lett.* **36**(16), 3212–3214 (2011)
10. Ford, T.N., Lim, D., Mertz, J.: Fast optically sectioned fluorescence HiLo endomicroscopy. *J. Biomed. Opt.* **17**(2), 0211051–0211057 (2012)
11. Lee, C.-Y., Han, J.-H.: Integrated spatio-spectral method for efficiently suppressing honeycomb pattern artifact in imaging fiber bundle microscopy. *J. Opt. Commun.* **306**, 67–73 (2013)
12. Lee, C.-Y., Han, J.-H.: Elimination of honeycomb patterns in fiber bundle imaging by a superimposition method. *Opt. Lett.* **38**(12), 2023–2025 (2013)
13. Cheon, G.W., Cha, J., Kang, J.U.: Spatial compound imaging for fiber-bundle optic microscopy. *Opt. Fibers Sens. Med. Diagn. Treat. Appl. XIV* **8938**(1), 393–398 (2014)
14. Cai, B., Liu, W., Zheng, Z., Wang, Z.F.: A new similarity measure for non-local means denoising. In: CCF Chinese Conference on Computer Vision, CCCV2015, Xi'an, vol. 546, pp. 306–316. Springer (2015)

Water-Surface Stability Analysis of a Miniature Surface Tension-Driven Water Strider Robot

Jihong Yan, Xinbin Zhang, Jie Zhao and Hegao Cai

Abstract When water strider robots row on water, the periodically stroking water surface of the actuating legs will unavoidably bring vibrations and instabilities that might cause the robots to sink into water. In this work, a stability analysis model for water strider robots rowing on water was proposed and a mass-spring-damper-like model was defined to describe the robot-water interactions. We applied this model to evaluate the water-surface stability of a miniature surface tension-driven water strider robot by detailing the effects of the actuating legs' rowing with different rowing frequencies on the vibration, pitching and swinging motions. The theoretical results indicate the robot possesses a good water-surface stability. The stability analysis model presented in this study can help with the design of water strider robots in future.

Keywords Water strider robot · Stability analysis model · Mass-spring-damper-like model · Robot-water interactions

1 Introduction

Human beings mainly live on land, and have less knowledge of some rural water areas due to the limited detection capabilities. Aquatic devices and robots can extend the scope of mankind's activities, and help us explore unknown water areas.

J. Yan (✉)

State Key Laboratory of Robotics and System, Laboratory for Space Environment and Physical Sciences, Harbin Institute of Technology, Harbin, China
e-mail: jhyan@hit.edu.cn

X. Zhang (✉)

Laboratory for Space Environment and Physical Sciences,
Harbin Institute of Technology, Harbin, China
e-mail: xbzhang2014@126.com

J. Zhao · H. Cai

State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, China

Nowadays, miniaturization technique and multi-robot system are popular research fields due to their own special features. Unique structure configuration and stroke mechanics enable water strider with quick, agile and high-efficiency water-surface locomotion, which is of great rationality and scientificity. In future, mini/micro water strider robots can walk on water with small resistance to motion and little disturbance in water, resulting in low noise and high efficiency, which possesses a surface tension-dominated locomotion, enabling it with agile motion capability even in shallow water regions, and can carry out various tasks such as water quality monitoring, aquatic search and rescue.

Rowing on the air-water interface and making full use of surface tension force greatly contribute to the agile and high-efficiency locomotion of water strider [1–3]. So it is of essential to analyze the water-surface stability for the design of water strider robots. Since Hu et al. [1] reported the first water strider robot that mimicking water strider's rowing locomotion by utilizing elastic band as the actuator, substantial water strider robot prototypes have been developed with various actuators and driven mechanisms [4–11]. Typically, Song et al. [4] designed a miniature water strider robot driven by three piezoelectric unimorph actuators; Takonobu et al. [5] developed a robot with its leg motion similar to that of an actual water strider; Suzuki et al. [6] proposed two different mechanisms for autonomous water strider robots by utilizing supporting legs decorated with hydrophobic microstructures by using MEMS techniques; Zhang et al. [7] developed a kind of copper-based superhydrophobic material by combing electrochemical deposition and self-assembly techniques at the nano scale and fabricated a microrobot capable of walking on water with two spiral-shaped actuating legs; Shin et al. [8]. Koh et al. [9] reported a water jumping robot by using shape memory alloys as actuators and Zhao et al. [10] proposed a water jumping robot capable of continually and stably jumping on water. However, till now, none of them have taken the robots' water-surface stability into account.

In this study, to evaluate the water-surface stability of a miniature surface tension-driven water strider robot we developed recently [11], for the first time, a model for the analysis of the stability of water strider robot's water-surface motion is proposed based on a theoretical model of long-and-thin cylinders interacting with water [12]. A mass-spring-damping-like model is defined to describe the robot-water interactions. Then the effects of actuating legs' stroking water surface on the robot's stability are discussed detailly. This study provides a stability analysis model for water strider robots, and might help us design new surface tension-driven aquatic mobile devices and develop relevant techniques in future.

2 Problem Statement

Aiming at mimicking the rowing locomotion of a water strider on water, we proposed a new miniature surface tension-driven robot by using a cam-link mechanism that can generate an ellipse-like spatial trajectory like a real water strider [11].

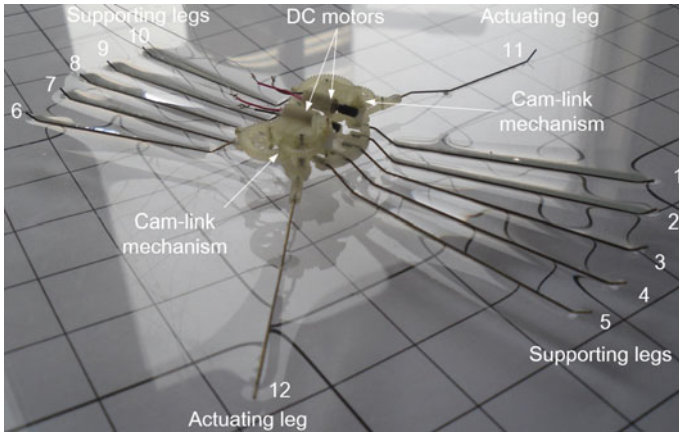


Fig. 1 Photograph of the fabricated miniature surface tension-driven water strider robot

Table 1 Dimension parameters of the supporting and actuating legs

Leg number	1	2	3	4	5	6	7	8	9	10	11	12
Length (mm)	50	50	50	50	50	38	43	50	43	38	30	30

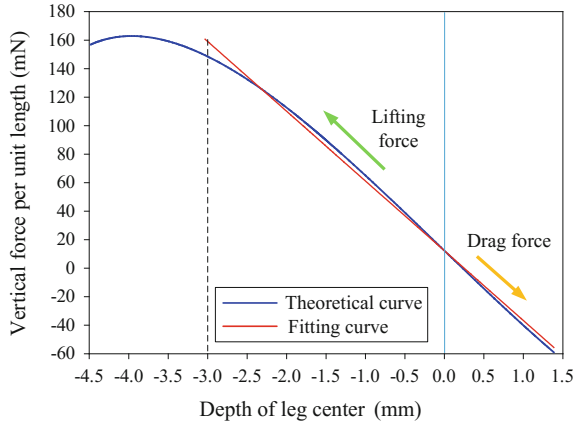
Figure 1 shows the fabricated robot prototype, which consists of ten supporting legs, two actuating legs driven by two miniature DC motors via cam-link mechanisms, respectively. The supporting legs are numbered 1–10, and the actuating legs are numbered 11 and 12, respectively. Both the supporting and actuating legs are made of copper tubes with superhydrophobic surfaces, of which the static contact angle is up to 155.8° . The diameter of the legs is about 0.5 mm and the lengths are listed in Table 1. The whole robot is about 136 mm in length and 150 mm in width, and weighs about 3.9 g.

The work area of water strider robots, water surface, is highly mobile and easily perturbed. Any factor that produces fluctuations in water or shakiness in the robot body might cause the robot to sink into water, so water-surface stability is an important performance index and relevant analysis is of essential for the design of water strider robots. There are many factors that can influence the robot’s stability. Expect for some external factors such as winds and water waves, the periodically stroking water surface of the two actuating legs will unavoidably bring vibrations and instabilities associated with adverse effects on the robot’s water-surface motion. This work is mainly focused on the instabilities induced by the two rowing actuating legs.

3 Stability Analysis Model

For water strider robots, the supporting legs can be treated as long and thin cylinders in parallel contact with water, of which the section radius is much smaller than the length, then the leg-water interactions can be analyzed by a 2-D model

Fig. 2 Vertical force acting on a unit-length leg



[12]. Then the vertical force obtained by a unit-length leg with the cross section diameter and static contact angle of 0.5 mm and 155.8° respectively can be calculated by solving Young-Laplace equation. As shown in Fig. 2, the vertical force is lifting force if directed upward and drag force if directed downward. It is interesting that the vertical force is linear with leg center depth within the range of -3 to 1.3 mm, where water strider robots usually float. Thus the vertical force f_{vl} within this range can be written as

$$f_{vl} = (0.012 - 49h_0)dL \tag{1}$$

where h_0 is the leg center depth, and L denotes the leg length.

When water strider robots move on water, the legs also suffer from viscous resistance and hydrodynamic pressure in the vertical direction, which can be expressed as a whole f_{vr} by an empirical expression [11]:

$$f_{vr} = \frac{1}{2}c_v\rho U_v^2dS \tag{2}$$

where ρ is the density of water, c_v denotes the resistance coefficient, S is the leg-water contact area and U_v represents the vertical velocity.

According to Eqs. (1) and (2), the robot-water interactions can be treated as a mass-spring-damper-like model, as is shown in Fig. 3: “mass” is located on the gravity center of the robot; “spring” describes the lifting force and drag force exerted on the legs in static states; “damper” represents the sum of viscous resistance and hydrodynamic pressure arised from the leg-water relative motions in the vertical direction. Unlike a real mass-spring-damper system, the damping factor is proportional to not velocity but the square of velocity, and dependent on the direction of the velocity vector. By using the geometric relations shown in Fig. 3, we get

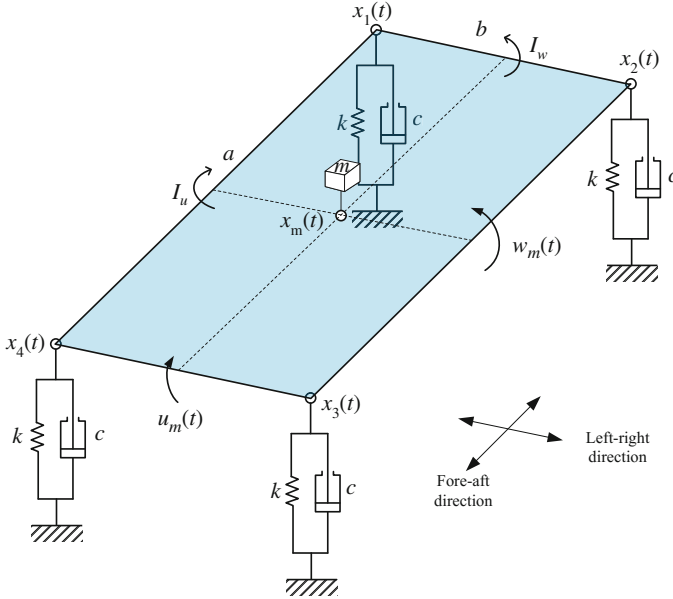


Fig. 3 Analysis model for the water-surface stability of water strider robots

$$\begin{cases} x_m(t) = \frac{1}{2}(x_1(t) + x_3(t)) = \frac{1}{2}(x_2(t) + x_4(t)) \\ x_2(t) = x_1(t) + b \sin w_m(t) \\ x_3(t) = x_4(t) + b \sin w_m(t) \\ x_3(t) = x_2(t) + a \sin u_m(t) \\ x_4(t) = x_1(t) + a \sin u_m(t) \end{cases} \quad (3)$$

where $x_1(t)$, $x_2(t)$, $x_3(t)$ and $x_4(t)$ are vertical displacements, $x_m(t)$, $w_m(t)$ and $u_m(t)$ denote the vertical displacement, swinging and pitching angles of the center of mass, respectively, and a and b are structural parameters.

According to Eq. (3), the equation for the robot’s water-surface stability can be expressed as

$$\begin{cases} f_m(t) - mg - 4kx_m(t) - 4c\dot{x}_m(t) = m\ddot{x}_m(t) \\ M_w(t) - b^2(k \sin w_m(t) \cos w_m(t) + \frac{1}{2}c\dot{x}_m(t)\dot{w}_m(t) \cos^2 w_m(t)) = I_w\ddot{w}_m(t) \\ M_u(t) - a^2(k \sin u_m(t) \cos u_m(t) + \frac{1}{2}c\dot{x}_m(t)\dot{u}_m(t) \cos^2 u_m(t)) = I_u\ddot{u}_m(t) \end{cases} \quad (4)$$

where $f_m(t)$ is the disturbance forces in the vertical direction, k and c are the equivalent elastic coefficient and damping factor of the spring, respectively, $M_w(t)$ and $M_u(t)$ are the swinging and pitching torques, respectively, m denotes the mass of the robot, and I_w and I_u are defined as the moment inertias of the robot in the left-right and fore-aft directions, respectively.

Table 2 Some parameter values of the robot

Parameters	k (N/m)	a (m)	b (m)	I_w (kg · m ²)	I_u (kg · m ²)
Values	5.6	0.073	0.023	1.7×10^{-9}	8.3×10^{-10}

The initial conditions are

$$\begin{cases} x_i(0) = -\frac{mg}{4k} \\ w_m(0) = u_m(0) = 0 \end{cases} \tag{5}$$

where the subscript $i = 1, 2, 3$ and 4 .

The equivalent elastic coefficient k and damping factor c can be determined by Eqs. (1) and (2). By ignoring the air resistance, c equals zero if the leg moves upward and $0.045 \text{ N} \cdot \text{s}^2/\text{m}^2$ if the leg moves downward. The structural parameter a and b can be configuration of the robot and the arrangement of the supporting legs. The moment inertias can be obtained via CAD model of the robot by ignoring the shift of the robot gravity center when the actuating legs rowing. Relevant values are listed in Table 2.

4 Disturbance Produced by the Actuating Legs

In this work, we mainly focus on the effects of actuating legs’ stroking water surface on the robot’s stability. Figure 4 shows the schematic diagram of the cam-link mechanism. The disturbance force and torques produced by an actuating leg rowing on water can be written as

$$\begin{cases} f_m(t) = \int_{SE} (f_{vl} + f_{vr}) dx \\ M_w(t) = \int_{SE} [(f_{vl} + f_{vr})(d_a + (d+x) \cos w_D) \cos \theta \\ \quad - (f_{hs} + f_{hr})(d_v - (d+x) \sin \theta) \sin w_D] dx \\ M_u(t) = \int_{SE} [(f_{vl} + f_{vr})(d_h + (d+x) \sin w_D) \cos \theta \\ \quad + (f_{hs} + f_{hr})(d_v - (d+x) \sin \theta) \cos w_D] dx \end{cases} \tag{6}$$

where d_a , d_v and d_h the relative position parameters between the installation location of the actuating legs and the gravity center of the robot in three orthogonal

Fig. 4 Schematic diagram of the cam-link mechanism

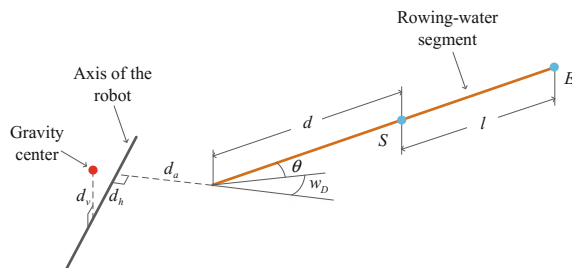


Table 3 Some parameter values of the robot

Parameters	d_v (m)	d_h (m)	d_a (m)	d (m)	l (m)
Values	0.008	0.0025	0.012	0.03	0.03

axes, θ and w_D are the up-down and fore-aft swinging angles of the actuating leg, respectively, d and l are the structural parameters of the actuating leg. Relevant values in this case can be find in Table 3. And f_{hs} and f_{hr} are the surface tension force, viscous resistance and hydrodynamic pressure in the horizontal direction, which can be find in [11].

According to our previous publications on this robot, the rowing trajectory of the actuating legs can be divided into four stages, including pressing (from $42^\circ/w$ to $55^\circ/w$), rowing (from $55^\circ/w$ to $85^\circ/w$), detaching (from $85^\circ/w$ to $133^\circ/w$) and backward swinging (from $133^\circ/w$ to $42^\circ/w$) processes, as is shown in Fig. 5, in

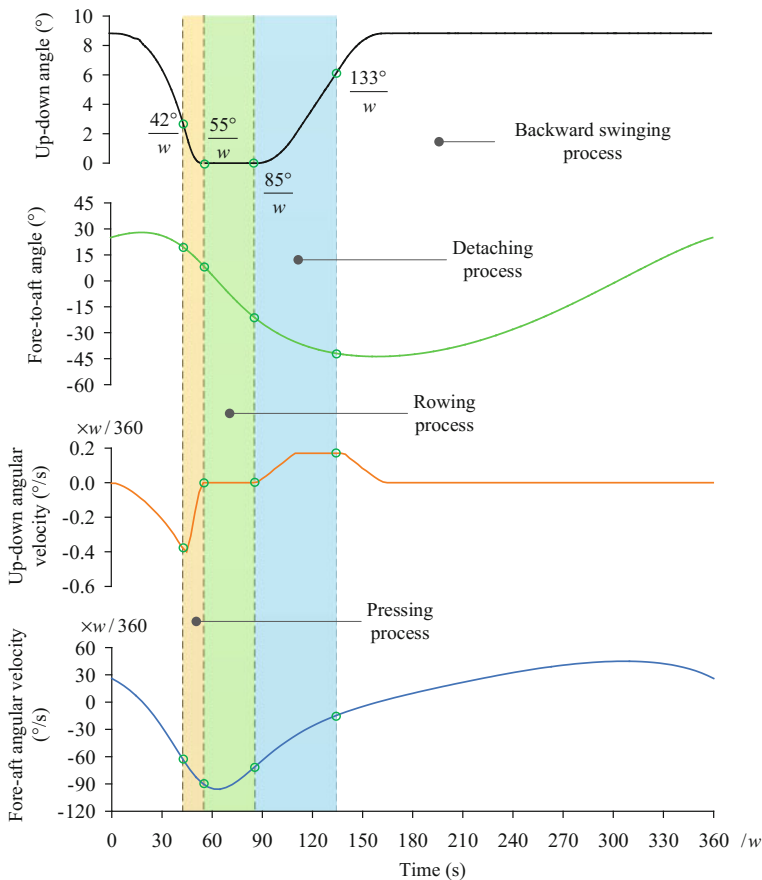


Fig. 5 Swing angles and angular velocities of the actuating leg varying with time during a driving stroke

which w represents the rotating velocity of the driving link [11] and can be expressed as a function of rowing frequency f ($w = 2\pi f$). Then in order to facilitate the following calculation, the relations between the swing angles and angular velocities are polynomially fitted as piecewise linear functions:

$$\theta = \begin{cases} 13.75 - 0.25wt & \frac{42^\circ}{w} \leq t < \frac{55^\circ}{w} \\ 0 & \frac{55^\circ}{w} \leq t < \frac{85^\circ}{w} \\ -10.2 + 0.12wt & \frac{85^\circ}{w} \leq t \leq \frac{133^\circ}{w} \end{cases} \quad (7)$$

$$\theta = \begin{cases} -0.0046w + 0.0000083w^2t & \frac{42^\circ}{w} \leq t < \frac{55^\circ}{w} \\ 0 & \frac{55^\circ}{w} \leq t < \frac{85^\circ}{w} \\ -0.0016 + 0.000002w^2t & \frac{85^\circ}{w} \leq t \leq \frac{110^\circ}{w} \\ 0.00047w & \frac{110^\circ}{w} \leq t \leq \frac{133^\circ}{w} \end{cases} \quad (8)$$

$$w_D = 87.8 - 1.78wt + 0.006w^2t^2 \quad \frac{42^\circ}{w} \leq t \leq \frac{133^\circ}{w} \quad (9)$$

$$\dot{w}_D = 0.42w - 0.026w^2t + 0.00028w^3t^2 - 0.000001w^4t^3 \quad \frac{42^\circ}{w} \leq t \leq \frac{133^\circ}{w} \quad (10)$$

The maximum rowing frequency of this robot is about 2 Hz, beyond which the actuating leg will pierce into water resulting in a great reduction in propulsion. Therefore the rowing frequency we discussed in this work ranges from 0 to 2 Hz. Then according to the models of leg-water interactions [11], the disturbance force and torques produced by an actuating leg rowing on water can be obtained.

5 Results and Discussion

According to the analysis model of interactions between multi long and thin cylinders and water, the robot can at most bear the up-down vibration, left-right swinging angle and fore-aft pitching angle of 3 mm, $\pm 6.5^\circ$ and $\pm 2^\circ$ before sinking.

5.1 A Single Actuating Leg Rowing on Water

Based on Eqs. (6)–(10), the disturbance force and torques exerted on the robot in the case of a single leg stroking water with different rowing frequencies are calculated and showed in Fig. 6. Then the relevant vibration displacement, pitching and swinging angles of the robot are obtained as shown in Fig. 7, from which we can see that the maximum vibration amplitude, swinging and pitching angles are all far smaller than the critical values; after a drive stroke, the robot can recover the stable state before the next rowing motion, indicating that this robot possesses a stable motion when a single leg stroking water.

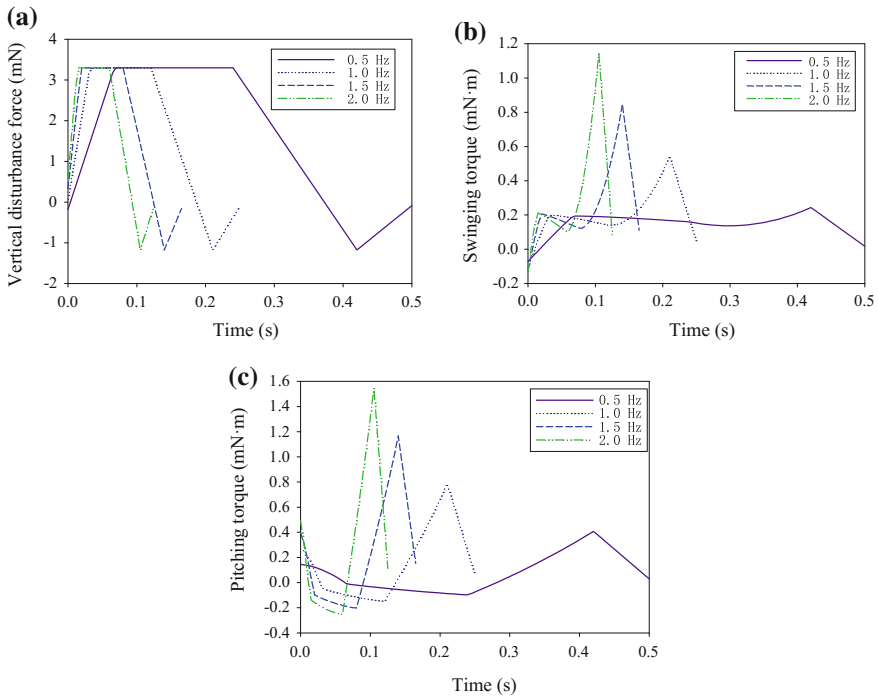


Fig. 6 Disturbance force and torques exerted on the robot in the case of a single leg stroking on water with different rowing frequencies: **a** vertical disturbance force, **b** swinging torque and **c** pitching torque

5.2 Two Actuating Legs Synchronously Rowing on Water

When the two actuating legs synchronously stroking water surface, the robot moves linearly. In this case, the two left-right disturbance torques cancel out, and we only need to discuss the effects of the robot’s vertical vibration and fore-aft pitching on the robot’s stability. Apparently, the disturbance force and torques exerted on the robot in this case are two times of those when a single leg stroking water. Then the relevant vibration displacement and pitching angle of the robot are obtained as shown in Fig. 8. We can see that, in comparison with the case of a single leg stroking water, when the two actuating legs synchronously rowing, the robot need more time to recover stable. Both the maximum vibration displacement and pitching angle are doubled, but still within the allowance. The robot can recover the stable state before the next rowing motion, indicating that this robot can stably moving on water by synchronously rowing its two actuating legs.

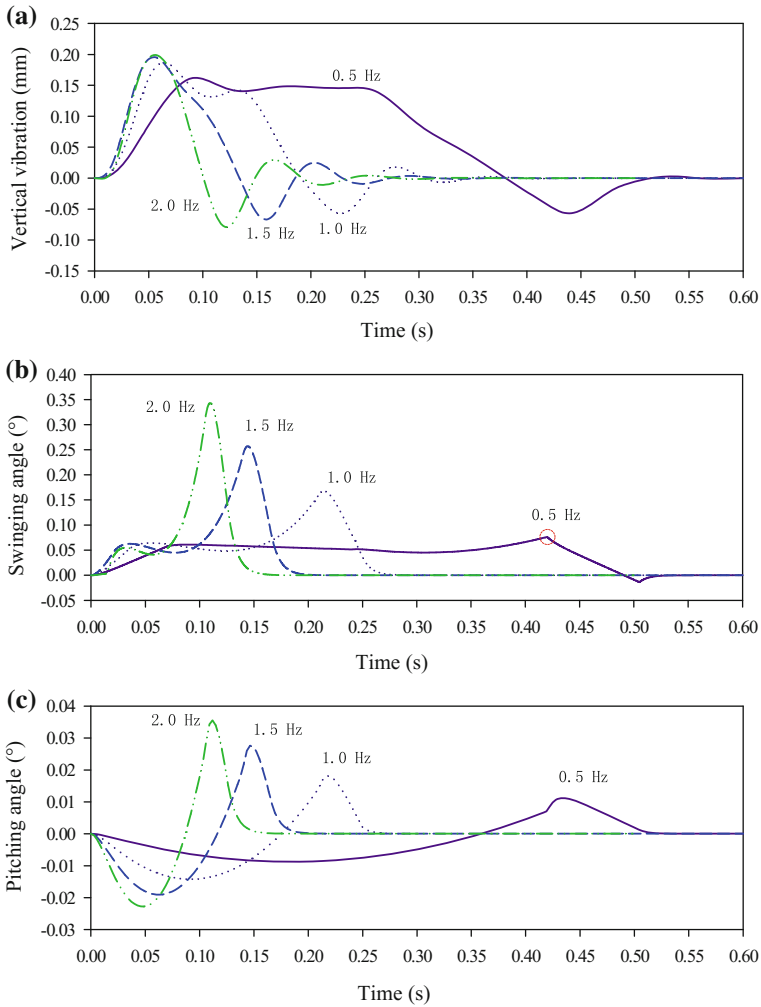


Fig. 7 Effects of a single actuating leg stroking water surface on the robot’s stability with different rowing frequencies: **a** vertical vibration, **b** swinging angle and **c** pitching angle

5.3 Two Actuating Legs Alternately Rowing on Water

In this case, the two actuating legs row alternately with a half-rowing cycle difference. From Fig. 7 it can be seen that the robot recovers the stable state within with a half-rowing cycle except the vertical vibration when the rowing frequency is 2 Hz. Therefore we here only discuss the effects of the two actuating legs row alternately with the rowing frequency of 2 Hz on the robot’s water-surface stability. Figures 9 and 10 show the vertical disturbance torque and induced vibration of the

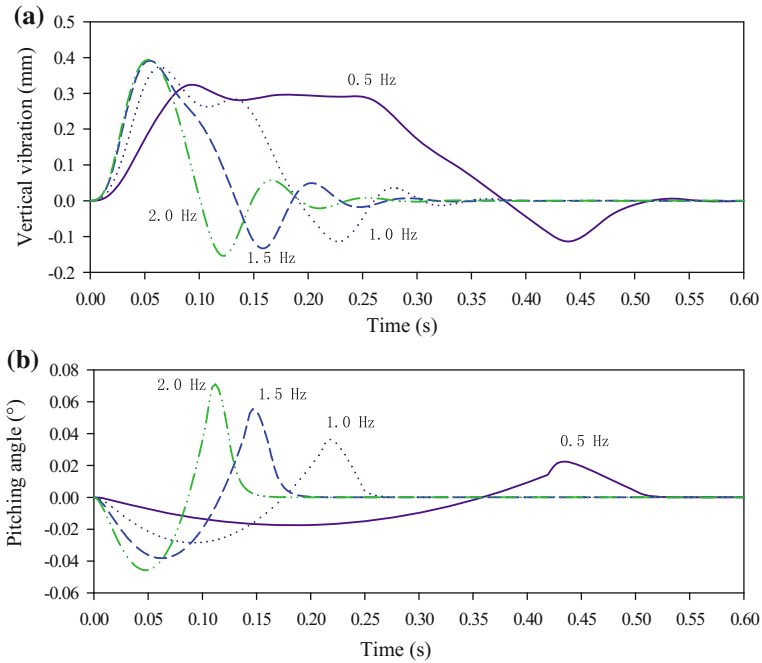


Fig. 8 Effects of two actuating legs synchronously stroking water surface on the robot's stability with different rowing frequencies: **a** vertical vibration, **b** pitching angle

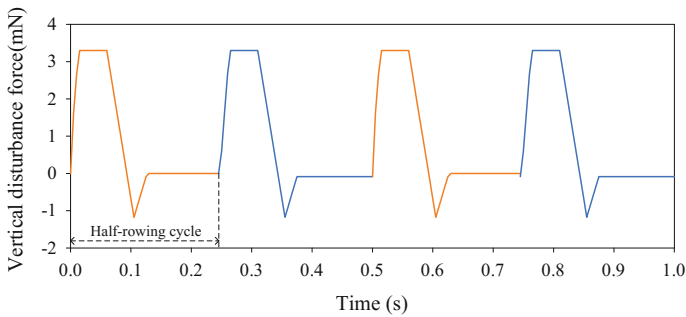


Fig. 9 Vertical disturbance force exerted on the robot when the two actuating legs rowing alternately on water with a rowing frequency of 2 Hz

robot when the two actuating legs rowing alternately on water, respectively. It is found that the vibration displacement periodically varies with time. The maximum amplitude is about 0.201 mm, which is far smaller than the critical value of piercing into water, indicating that this robot is very stable when the two actuating legs row alternately with a half-rowing cycle difference.

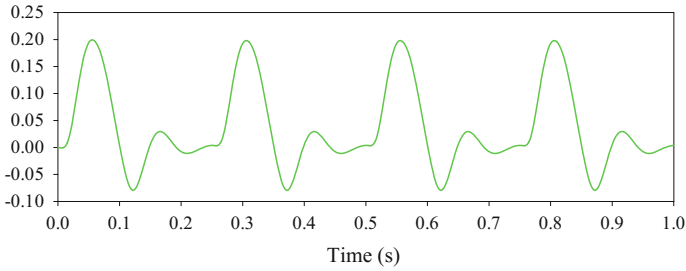


Fig. 10 Effects of the vertical disturbance force on the robot's water-surface stability when the two actuating legs rowing alternately on water with a rowing frequency of 2 Hz

6 Conclusions

Aiming at evaluating the water-surface stability of a miniature surface tension-driven water strider robot, for the first time, we proposed an analysis model for the water-surface motion stability of water strider robots in terms of theoretical models of long-and-thin cylinders interacting with water. A mass-spring-damping-like model was defined to describe the robot-water interactions. Then the effects of actuating legs' stroking water surface on the robot's stability are analyzed by discussing three cases: a single leg stroking water surface, the two actuating legs synchronously rowing on water and the two actuating legs row alternately with a half-rowing cycle difference. Results indicates that this miniature surface tension-driven water strider robot can stably move forward and make turns with different gaits on water, which might help us carry out various aquatic tasks in future. The stability analysis model we proposed in this work can help with the design of water strider robots and might provide reference to develop new mini/micro surface tension-driven devices.

Acknowledgements This work was financially supported by Natural Science Foundation of China (NSFC, Grant 51305098).

References

1. Hu, D.L., Chan, B., Bush, J.W.: The hydrodynamics of water strider locomotion. *Nature* **424**, 663–666 (2003)
2. Gao, X., Jiang, L.: Biophysics: water-repellent legs of water striders. *Nature* **432**, 36–36 (2004)
3. Hu, D.L., Bush, J.W.: The hydrodynamics of water-walking arthropods. *J. Fluid Mech.* **644**, 5–33 (2010)
4. Suhr, S., Song, Y., Lee, S., Sitti, M.: Biologically inspired miniature water strider robot. *Proc. Robot. Sci. Syst.* I 319–326 (2005)

5. Takonobu, H., Kodaira, K., Takeda, H.: Water strider's muscle arrangement-based robot. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1754–1759. IEEE Press, Edmonton (2005)
6. Suzuki, K., Takano, H., Noya, K., Koike, H., Miura, H.: Water strider robots with microfabricated hydrophobic legs. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 590–595. IEEE Press, San Diego (2007)
7. Zhang, X., Zhao, J., Zhu, Q., Chen, N., Zhang, M., Pan, Q.: Bioinspired aquatic microrobot capable of walking on water surface like a water strider. *ACS Appl. Mater. Inter.* **3**, 2630–2636 (2011)
8. Shin, B., Kim, H., Cho, K.: Towards a biologically inspired small-scale water jumping robot. In: Proceedings of IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, pp. 590–595. IEEE Press, Scottsdale (2008)
9. Koh, J.S., Yang, E., Jung, G.P., Jung, S.P., Son, J.H., Lee, S.I., Cho, K.J.: Jumping on water: Surface tension-dominated jumping of water striders and robotic insects. *Science* **349**, 517–521 (2015)
10. Zhao, J., Zhang, X., Chen, N., Pan, Q.: Why Superhydrophobicity is crucial for a water-jumping microrobot? Experimental and theoretical investigations. *ACS Appl. Mater. Inter.* **4**, 3706–3711 (2012)
11. Yan, J., Zhang, X., Zhao, J., Liu, G., Cai, H., Pan, Q.: A miniature surface tension-driven robot using spatially elliptical moving legs to mimic a water strider's locomotion. *Bioinspiration Biomimetics* **10**, 046016 (2015)
12. Zhang, X., Yan, J., Zhao, J., Wang, Y., Pan, Q.: Vertical force acting on partly submerged spindly cylinders. *AIP Adv.* **4**, 047118 (2014)

Mechanism Allowing a Mobile Robot to Apply a Large Force to the Environment

Shouhei Shirafuji, Yuri Terada and Jun Ota

Abstract In this study, we investigated a mechanism that allows a mobile robot to apply a large force to the environment. We first investigated the limits on the force that a mobile robot can apply to a target object by analyzing the forces between the robot, ground, and object and the limits on the frictional forces between them. To prevent the mobile robot from falling when applying a large force, we developed a prototype in which the manipulator was connected via a passive rotational joint. We investigated the pushing capacity of the prototype robot through an experiment in which it tilted a large object. The results confirmed that the mechanism allows a mobile robot to apply a large force to an object without falling by trial and error.

Keywords Mobile robot · Large force

1 Introduction

When considering object manipulation by a robot, it is often assumed that the target object can be grasped firmly by the robot's end-effector. In the case of a robot designed to move objects, this is called pick-and-place, and is a widely used technique in the field of industrial robotics. However, in the case of a robot designed to explore unknown environments and to manipulate unknown objects, e.g., in a domestic environment or at a disaster site, this approach is often impossible because the objects offers no gripping point or is difficult to locate. In such cases, an approach is needed that allows the robot to manipulate the object without grasping it.

In one such approach, the robot will slide the object by pushing it [1–5]. A number of studies have examined the planning and stable execution of such a maneuver by wheeled mobile robots [1–3]. Harada et al. [4] proposed a control method

S. Shirafuji (✉) · J. Ota

Research into Artifacts, Center for Engineering, The University of Tokyo,
Kashiwanoha 5-1-5 Kashiwa-shi, Chiba 277-8568, Japan
e-mail: shirafuji@race.u-tokyo.ac.jp

Y. Terada

Department of Precision Engineering, Graduate School of Engineering,
The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

© Springer International Publishing AG 2017

W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_58

795

for a legged humanoid robot using control of the zero moment point (ZMP). Other approaches have included rotation of the object by a single or multiple robots [6, 7]. These approaches are collectively called nonprehensile manipulation or grasp-less manipulation [8]. Nonprehensile manipulation requires less force to be applied than in manipulation with grasping because the gravitational force acting on the object is transferred to the environment. Nonprehensile manipulation is particularly useful when a small mobile robot is manipulating a relatively large object. However, when applying nonprehensile manipulation, the force that can be exerted by the robot is limited because there is risk of the robot tipping over or slipping if a large reaction force is applied.

A key factor is the frictional force between the object and the end-effector of the robot. When executing a given task with nonprehensile manipulation, the robot often has to support the object at a single frictional point of contact or several frictional points of contact concentrated in a small area. When a mobile robot manipulates an object without grasping, the frictional force between the robot and the ground must also be taken into account. As the robot is not fixed to the ground, the rotational forces applied to the robot through the contact point must be carefully controlled. These factors determine how much force the robot can apply to the environment without falling or slipping.

In this research, we clarified the pushing capacity of a mobile robot by calculating the limits of the frictional force with the ground and with the object, and the rotational force that could be applied to the robot when executing nonprehensile manipulation. We used these findings to design a mechanism in which a linear actuator is connected to the mobile robot through a passive rotational joint, allowing large force to be applied to the object without the risk that the robot would fall. We showed that the range of directions in which the force can be applied to the object is more limited than that of a mobile robot with a conventional manipulator, but the maximum force that can be applied is the same in both cases. We developed a prototype mobile robot fitted with the proposed mechanism and conducted experiments to show that it was able to apply a large force to an object without falling. We further demonstrated that the mobile robot can determine the frictional coefficient between the end-effector and the object by trial and error.

To simplify the analysis, we considered the motions and forces only in a plane, but the analysis can also be applied in three dimensional space.

2 Pushing Capacity of a Mobile Robot

In this section, we analyze the maximum force that a given mobile robot can apply to an object by pushing. As shown in Fig. 1, we assumed that a mobile robot pushes an object in front of its body by direct contact. Figure 1a shows a legged humanoid robot pushing the object using a hand and Fig. 1b shows a wheeled robot pushing the object using a manipulator.

In both cases, the object is manipulated through one or more points of frictional contact with the end-effector. The force applied to the end-effector can be derived by

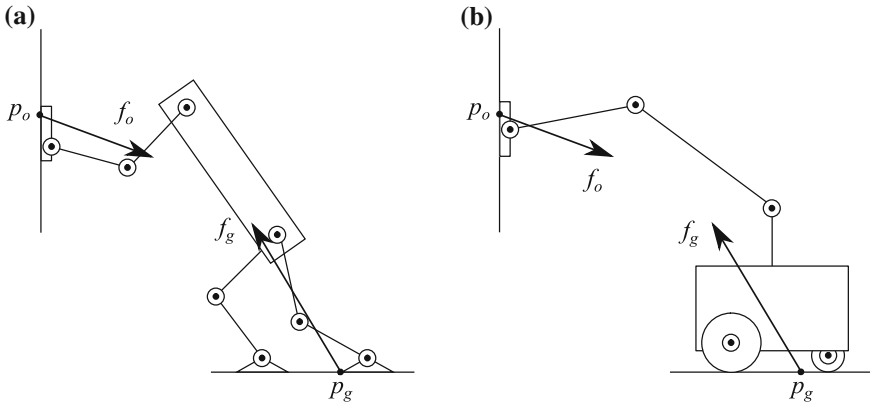


Fig. 1 Forces applied by a mobile robot to an object and to the ground when pushing an object. **a** Legged mobile robot. **b** Wheeled mobile robot

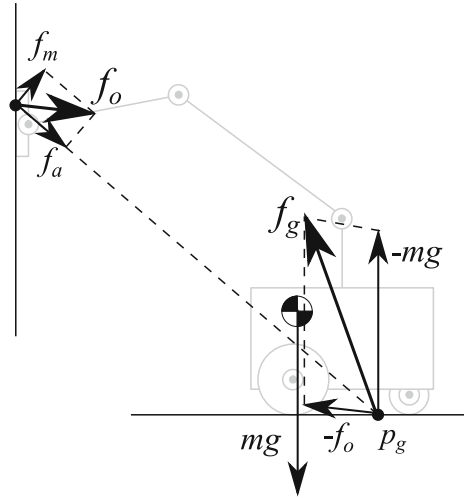
considering a point that does not produce any moment. This point can be calculated or measured at the center of pressure and is called the zero moment point (ZMP) [9]. In the cases shown in Fig. 1, the reaction force applied to the end-effector by the object can be represented as force $f_o \in \mathbb{R}^2$ applied to ZMP $p_o \in \mathbb{R}^2$ as shown in Fig. 1.

In the case of a mobile robot, multiple frictional contact points between the robot and the ground must also be taken into account because the robot is not fixed to the ground. These can also be represented by the force applied to a ZMP. For example, the force $f_g \in \mathbb{R}^2$ applied to ZMP $p_g \in \mathbb{R}^2$ represents the forces applied to the feet or wheels, as shown in Fig. 1. The ZMP need not be located directly on the feet or wheels, but it must fall in the region between the contact points.

When considering nonprehensile manipulation by a mobile robot using frictional forces, it is necessary to avoid sliding both between the robot and object and between the robot and ground. The frictional forces provide the main limitation on the force that the robot can apply to the object. We therefore considered the relationship between the forces applied to the robot by the ground ZMP f_g and from the ZMP on the surface of the object f_o . In this analysis, we considered only the wheeled robot shown in Fig. 1b. For the ZMPs at the ground and the surface of the object p_g and p_o , respectively, the static equilibrium of the forces applied to the robot around p_g are shown in Fig. 2, where g is the gravitational acceleration and m is the mass of the robot. For simplicity, the dynamics of the robot were not considered. The component of the reaction force applied to p_o in the direction of the straight line connecting the two ZMPs p_g and p_o is f_a and the component of the reaction force in the direction orthogonal to f_a is f_m . The static equilibrium of the moments applied to p_g , f_m corresponds to the component that cancels the moment caused by the gravitational force at center of mass. Thus, the moments around p_g are in balance, and the reaction force applied to p_o is given by

$$f_o = f_a + f_m. \tag{1}$$

Fig. 2 Static equilibrium of the forces applied to the robot around the ZMP between the robot and the ground p_g



The reaction force applied to p_g is given by

$$f_g = -f_o - mg \tag{2}$$

from the static equilibrium of forces applied to p_g as shown in Fig. 2, where $g = [0 \ -g]^T$.

We next considered the limitations on the frictional forces that could be applied, using these reaction forces. The frictional force between a robot and the environment is generally represented by the Coulomb friction model, and the relationship between a normal force applied to the contact surface and a tangential force is defined using a static friction coefficient in which there is no relative motion between the robot and the environment. We represented the static frictional coefficient between the robot and the ground as μ_g and the static frictional coefficient between the robot and the object as μ_o . Using μ_o , the set of reaction forces that can be applied to the robot by the object, given the frictional force between the robot and the object, can be derived by

$$F_o = \{f_o \in \mathbb{R}^2 : \|f_o^t\| \leq \mu_o \|f_o^n\|, f_o^n \geq 0\} \tag{3}$$

where f_o^n is the component of the reaction force applied by the object in the direction orthogonal to the contact surface and f_o^t is the component in the direction tangential to the contact surface. The region defined by this set is represented as a cone, as shown in Fig. 3a. The set of reaction forces that can be applied to the robot from the ground is given by the friction between the robot and the ground, as follows:

$$F_g = \{f_g \in \mathbb{R}^2 : \|f_g^t\| \leq \mu_g \|f_g^n\|, f_g^n \geq 0\} \tag{4}$$

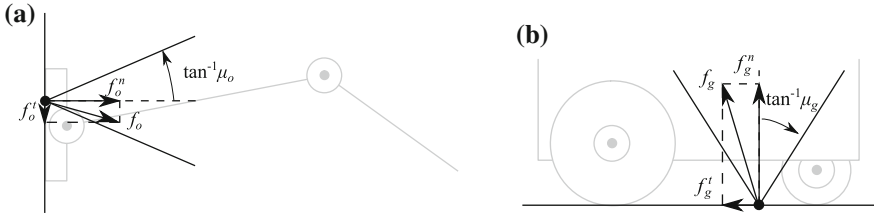


Fig. 3 Frictional cones defining the limits of the reaction forces applied to the object and the ground. **a** Frictional cone of the contact point between the robot and the object. **b** Frictional cone of the contact point between the robot and the ground

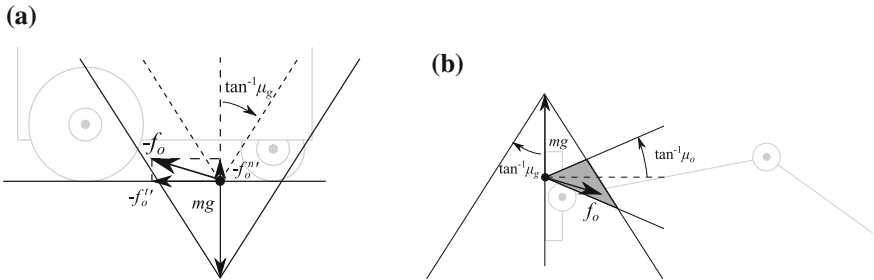


Fig. 4 **a** Region of force that can be applied to the object given the limitations of the frictional forces between the robot and the ground. **b** Region of force that can be applied to the object given the limitations of the frictional forces with both the object and the ground

where f_g^n is the component of the reaction force from the ground in the direction orthogonal to the contact surface, and f_b^t is the component in the direction tangential to the contact surface. This frictional cone is shown as Fig. 3b. The reaction forces applied to the ZMPs have to fall within these cones to avoid slippage.

To derive the maximum force that the robot can apply to the object, we considered the two frictional cones together. Substituting Eq. (2) into Eq. (4) gave

$$F_o^g = \{f_o \in \mathbb{R}^2 : \|f_o^t + mg^t\| \leq (\mu_g \|f^n + mg^n\|), \{f_o^n\}^t \leq -mg^n\} \quad (5)$$

where f_o^n and f_o^t are the components of f_o in the direction of orthogonal and tangential to the contact surface of the ground, respectively. The components of gravitational force in the directions orthogonal and tangential to the contact surface of the ground are given by g^n and g^t respectively, and $g^n = g$ and $g^t = 0$ when the contact surface is horizontal. This region can be seen as the frictional cone on p_o shifted by the amount of mg as shown in Fig. 4a. The set of reaction forces that can be applied to the robot by the object, taking into account of the frictional forces from both the ground and the object, is given by

$$F_o = F_o^o \cap F_o^g. \quad (6)$$

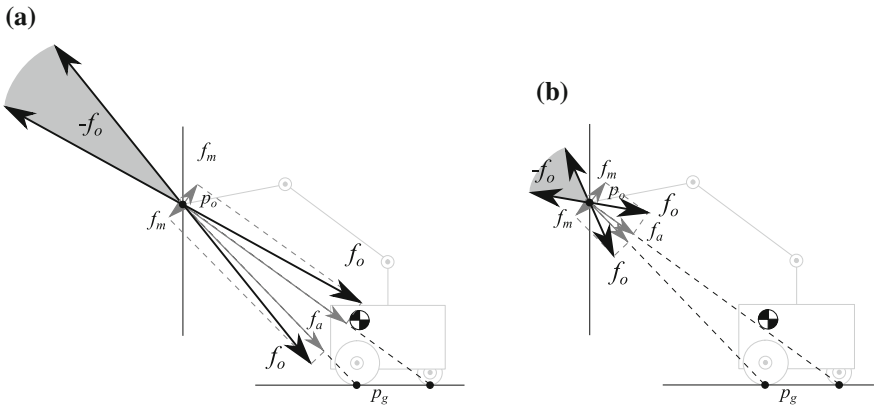


Fig. 6 Region of force that can be applied to the object given the limitations of the angles without falling of the robot. **a** Region of force when the force is large in relation to the gravitational force acting on the robot. **b** Region of force when the force is small in relation to the gravitational force acting on the robot

One approach to prevent the robot from falling when exerting a large force on the object is to prevent the moment generated between the object and the robot being transferred to the body of the robot. This is achieved by connecting the manipulator through a passive joint placed near the ground, as shown in Fig. 7a. This reduces or eliminates the moment conveyed to the body of the robot, becoming trivial when the location of the passive joint is sufficiently close to the ground. The static equilibrium of the forces applied to the ZMP (FRI) on the ground p_g is shown in Fig. 7b, where m_1 and m_2 are the mass of the robot body and the manipulator, respectively, and $m_1 + m_2 = m$.

A mobile robot with the proposed mechanism can exert a large force on an object without the risk of falling because the moment that rotates the mobile robot is not conveyed to the body of the robot. Moreover, the direction of force that the proposed robot exerts to an object cannot be adjusted while manipulating the object; however, a mobile robot to which a manipulator is connected rigidly can adjust the direction of a force in the range defined by the location of the contact points and the mass of robot as shown in Fig. 6. Thus, we can consider two approaches to the nonprehensile manipulation of an object by a mobile robot: a manipulation that adjusts the direction of the force applied to an object using a manipulator rigidly connected to the robot and a manipulation that exerts a force in one direction using a manipulator connected to the robot through a passive joint. In the latter approach, it is unnecessary to use a manipulator with multiple degrees of freedom as the robot cannot control the direction of the force. Thus, a manipulator comprising a linear actuator, as shown in Fig. 7a, is adequate in the latter approach.

These two approaches represent a tradeoff, as noted above, and choosing one of these approaches should reflect the given task. The latter approach is particularly effective when the force applied by a robot is large in comparison to the gravitational force acting on the robot.

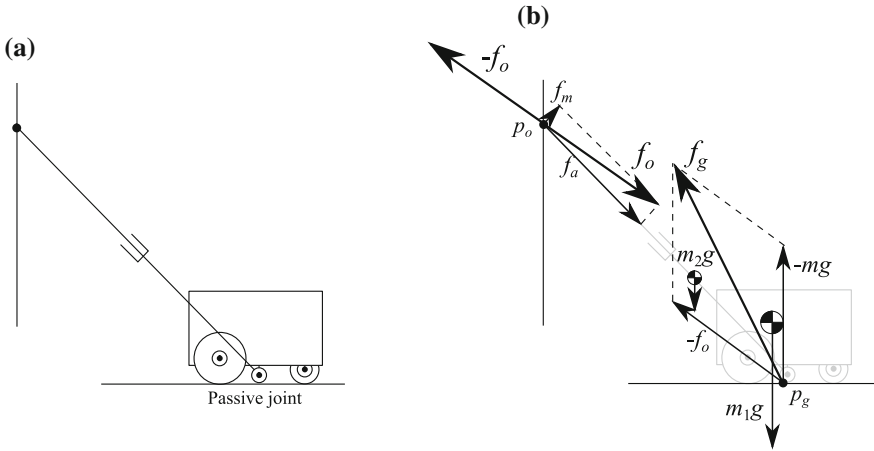


Fig. 7 **a** Mobile robot with a linear actuator connected by a passive joint. **b** Static equilibrium of forces applied to p_g

The region of force that can be applied to the object within the limitations of the frictional forces derived in the previous section is the same in both approaches, because in the both cases, the total mass applies to the ZMP (FRI) p_g . The maximum force that can be applied in a given direction is therefore essentially the same in both approaches.

4 Prototype of Mobile Robot with a Passive Joint

We developed a prototype mobile robot in which the manipulator was connected via a passive rotational joint, allowing a large force to be exerted on an object. Figure 8a shows this prototype, which was based on a commercial mobile robot, Pioneer 2-DX of Active Media. A manipulator comprising a linear actuator and an end-effector was connected to the robot via a passive rotational joint. A potentiometer was attached to the passive joint, allowing the angle of the manipulator relative to the body of the robot to be tracked.

The end-effector comprised a load cell, a potentiometer, and a slip sensor and was connected to the linear actuator via a second passive joint, as shown in Fig. 8b, c. The potentiometer was used to measure the angle of the surface of the object in relation to the manipulator, and the load cell measured a normal force applied to the surface. The slip sensor was used to detect any slippage between the end-effector and the object. Slippage was detected by rotating a cylindrical roller against the contact surface of the end-effector. The rotation of the roller was transformed into a digital signal via an optical absolute encoder. This type of slip sensor was first proposed by Ueda et al. [11]. By measuring the normal force applied to the object surface using

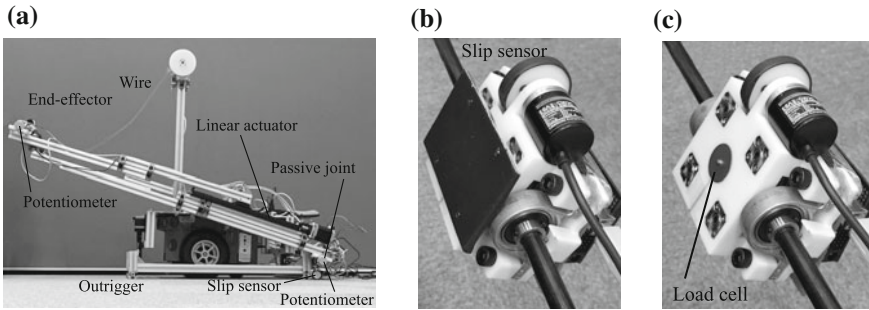


Fig. 8 Prototype mobile robot with a passive joint and an end-effector connected to the manipulator. **a** Prototype robot. **b** End-effector with rubber plate. **c** End-effector without rubber plate

the potentiometer and the load cell on the end-effector when the slip sensor detected a slippage, the frictional coefficient between the end-effector and the object μ_o could be estimated.

The same type of slip sensor was attached to an outrigger of the robot. The outrigger lifted the wheels of the robot to prevent them from rotating when a large force was applied. A contact was made with the ground by the outrigger rather than the wheels when the robot was manipulating an object. The sensor attached to the outrigger then attempted to detect any slippage between the robot and the object. As in the determination of μ_o , measurements were taken of a normal force applied to the object surface and the angle of the object surface relative to the ground using the potentiometers and a load cell. If the slip sensor detected a slippage between the ground and the robot, the frictional coefficient between the end-effector and the ground μ_g could have been determined.

By determining μ_o and μ_g , the limitation on the force the robot can exert on an object can be derived. This allows the limitation to be estimated by trial and error. However, such an approach incurs a risk that the robot will fall. In the case of our proposed mechanism, falling can be prevented by the passive joint.

The angle of the passive joint could not be controlled because no actuator was installed. Instead, a wire was connected to the manipulator, which was wound by an actuator attached to the robot to set the angle of the passive joint before the robot began to manipulate the object. While the robot was manipulating the object, the wire was slackened to prevent it from affecting the motion of the passive joint.

5 Experiments

To identify the limitation on the force that can be applied, we performed a series of trial and error experiments in which the robot applied a strong force to tilt a heavy object. The object was a square box in which water bottles were stacked to increase the weight to 80.0 kg.



Fig. 9 Slippage between the end-effector and an object being pushed by the robot

The object was tilted using the following steps. First, the robot set the angle of its manipulator by adjusting the wire length. The robot then moved forward until its end-effector was touching the object. The wire was loosened and the outrigger was put in contact with the ground. The robot then started to manipulate the object using the linear actuator while monitoring slippage between the robot, the ground, and the object.

Slippage between the robot and object depended on the angle of the manipulator and the force exerted by the robot. Figure 9 shows the slippage that occurred when the robot set the angle of the manipulator to 45° from the horizontal. Figure 10 shows the outputs of the sensors when the slippage occurred. Figure 10a shows the distance of the slippage measured by the slip sensor and that the distance increased sharply. We judged that the slippage occurred at the same time at which this steep increase occurred. The increase in distance was larger than the measured increase because of the limitations of the rotary encoder and because the gradual increase in the sliding distance before the slippage occurred changed the state of contact.

The static friction coefficient between the robot and object was derived from the normal force measured by the load cell (Fig. 10b), the angle of the passive joint (Fig. 10c), the angle of the end-effector (Fig. 10d), the weight of the manipulator (23.0 kg), and the position of center of mass of manipulator using the relationship of forces described in Sect. 4. This gave a static friction coefficient of $\mu_o = 0.44$.

Slippage between the robot and ground depended on the angle of the manipulator. Figure 11 shows the slippage that occurred when the robot set the angle of the manipulator to 41.5° from the horizontal. Figure 12 shows the outputs of the sensors when the slippage occurred and Fig. 12a shows the distance of sliding caused by the slippage that was measured by the slip sensor on the outrigger.

The static friction coefficient between the robot and the ground could also be calculated from the normal force measured by the load cell (Fig. 12b), the angle of the passive joint (Fig. 12c), the angle of the end-effector (Fig. 12d), the total weight of the robot (59.6 kg), and the position of the center of mass of the manipulator using the relationship of forces described in Sect. 4. It was estimated as $\mu_g = 0.18$.

Using the method of visualization proposed in Sect. 2, the range of force that the robot can exert on an object is shown in Fig. 13. Although the range of force depends on the materials from which the target object and floor are made, manipulation using mobile robots can be estimated in advance, as done in this experiment.

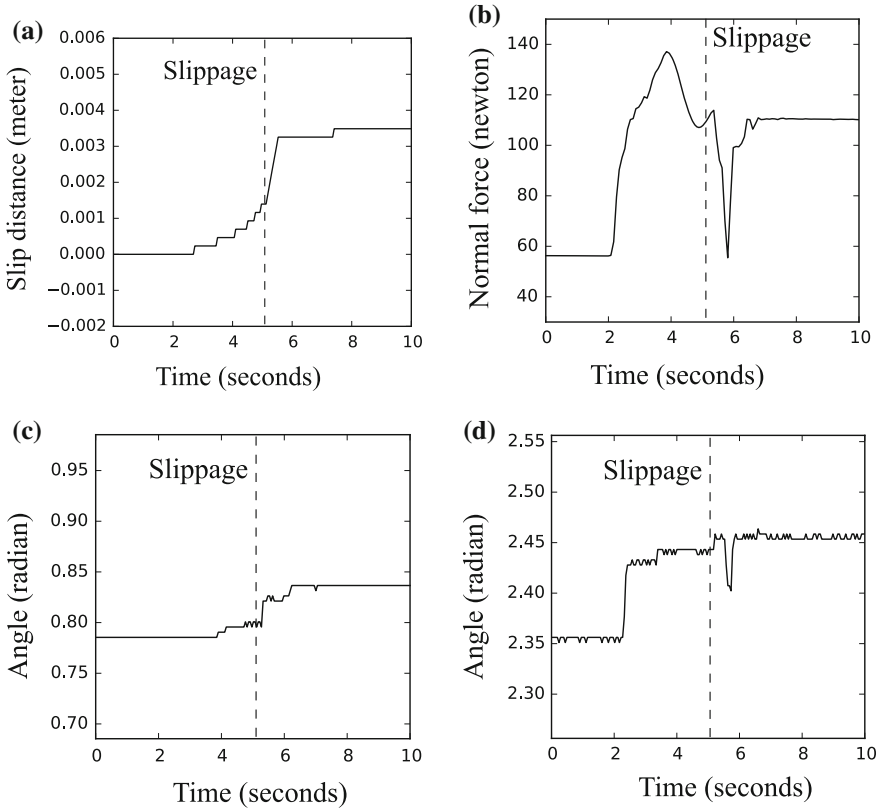


Fig. 10 Outputs of sensors when slippage occurred between the end-effector and object. **a** Output of the slip sensor on the end-effector. **b** Output of the load cell. **c** Output of the pentimeter on the passive joint. **d** Output of the potentiometer on the end-effector



Fig. 11 Slippage between the outrigger and ground while the robot pushed the object

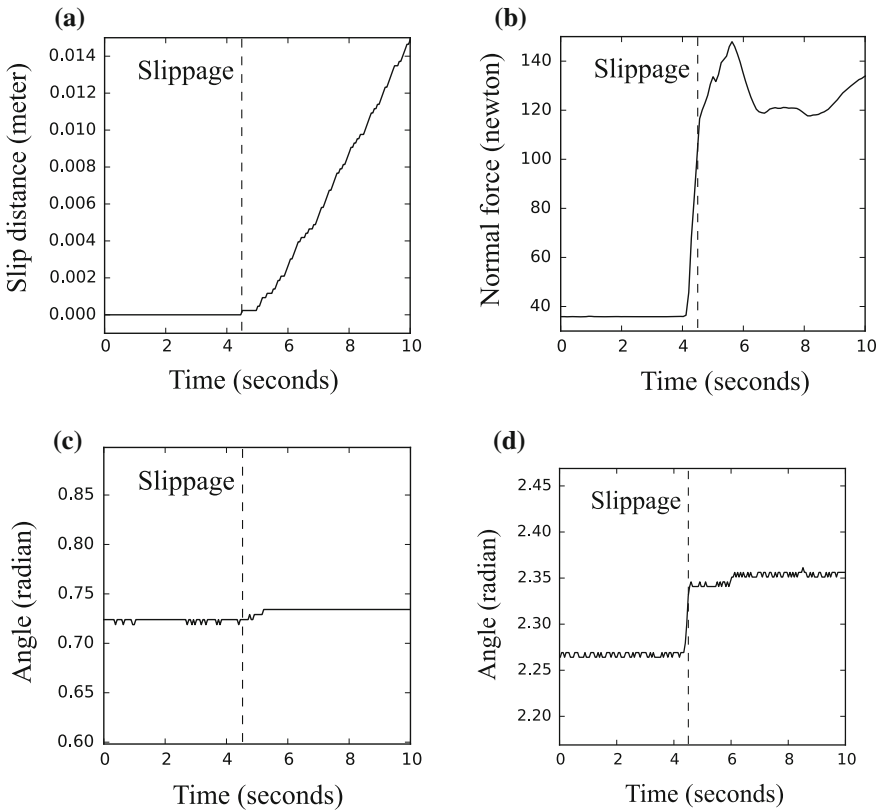
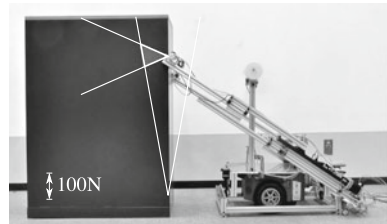


Fig. 12 Outputs of sensors when slippage occurred between the outrigger and the ground. **a** Output of the slip sensor on the outrigger. **b** Output of the load cell. **c** Output of the potentiometer on the passive joint. **d** Output of the potentiometer on the end-effector

Fig. 13 Range of force that can be applied to the object (defined by the estimated static friction coefficients)



6 Conclusions

In this paper we presented a design for a mobile robot using a passive joint in order to allow objects to be manipulated with a large force without the robot tipping over. We first clarified the limitations on the force that can be applied to the object by analyzing the frictional forces between the robot, the object, and the ground and then discussed the use of a passive joint in our proposed mobile robot and its advantages and disadvantages. We showed that the risk of falling could be averted, but at the cost of the robot's ability to change the direction of force in the course of object manipulation. This represents the tradeoffs between the proposed mobile robot and a standard mobile robot with a rigidly connected manipulator. The choice of mobile robot should reflect the specific task, and appropriate planning must be undertaken.

The proposed mechanism is especially helpful when manipulating an unknown large object in an unknown environment because the robot may use trial and error to identify the properties of the object and the environment, including their static friction coefficients. This was demonstrated in the experiment presented in Sect. 5. When a large reaction force is applied to the robot's manipulator, errors arise, including slippage between the robot and environment. This makes it difficult to avoid the robot falling because of the reaction force by controlling its manipulator. Our proposed mobile robot has the advantage that falling under a large reactive force is ruled out, which allows unknown environments to be explored. This was demonstrated when our robot successfully identified the static friction coefficients of an object and of the floor by trial and error.

Our analysis of the limitation of force, the demonstration of the use of a passive joint, and the prototype introduced in this paper represent the first steps toward the realization of nonprehensile manipulation of relatively large objects using small mobile robots. In future studies we will extend this analysis to manipulation in a three dimensional space. As a further step, we will improve the hardware and investigate nonprehensile manipulation using a mobile robot with a passive joint.

Acknowledgements A part of this study is the result of "HRD for Fukushima Daiichi Decommissioning based on Robotics and Nuclide Analysis" carried out under the Center of World Intelligence Project for Nuclear S&T and Human Resource Development by the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1. Lynch, K.: The mechanics of fine manipulation by pushing. In: Proceedings IEEE International Conference on Robotics and Automation. pp. 2269–2276 (1992)
2. Lynch, K., Mason, M.: Stable pushing: mechanics, controllability, and planning. *Int. J. Robot. Res.* **15**(6), 533–556 (1996)
3. Mason, M.: *Mechanics of Robotic Manipulation*. MIT press (2001)

4. Harada, K., Kajita, S., Kaneko, K., Hirukawa, H.: Pushing manipulation by humanoid considering two-kinds of zmps. In: Proceedings IEEE International Conference on Robotics and Automation. pp. 1627–1632 (2003)
5. Murooka, M., Nozawa, S., Kakiuchi, Y., Okada, K., Inaba, M.: Whole-body pushing manipulation with contact posture planning of large and heavy object for humanoid robot. In: Proceedings IEEE International Conference on Robotics and Automation. pp. 5682–5689 (2015)
6. Yoshida, E., Poirier, M., Laumond, J.P., Kanoun, O., Lamiroux, F., Alami, R., Yokoi, K.: Pivoting based manipulation by a humanoid robot. *Auton. Robots* **28**(1), 77–88 (2010)
7. Murooka, M., Noda, S., Nozawa, S., Kakiuchi, Y., Okada, K., Inaba, M.: Manipulation strategy decision and execution based on strategy proving operation for carrying large and heavy objects. In: Proceedings IEEE International Conference on Robotics and Automation. pp. 3425–3432 (2014)
8. Maeda, Y., Kijimoto, H., Aiyama, Y., Arai, T.: Planning of grasplless manipulation by multiple robot fingers. In: Proceedings IEEE International Conference on Robotics and Automation. pp. 2474–2479 (2001)
9. Vukobratović, M., Stepanenko, J.: On the stability of anthropomorphic systems. *Math. Biosci.* **15**(1), 1–37 (1972)
10. Goswami, A.: Postural stability of biped robots and the foot-rotation indicator (fri) point. *Int. J. Robot. Res.* **18**(6), 523–533 (1999)
11. Ueda, M., Iwata, K., Shingu, H.: Tactile sensors for an industrial robot to detect a slip. In: Proceedings International Symposium on Industrial Robots. pp. 63–70 (1972)

Part XIII
Robot Learning

Self-improving Robot Action Management System with Probabilistic Graphical Model Based on Task Related Memories

Yuki Furuta, Yuto Inagaki, Kei Okada and Masayuki Inaba

Abstract Robots on home environment have to deal with their environment that changes every time they perform tasks. In order to reduce improving descriptions for task and action planning manually, we propose a new framework to use probabilistic graphical model, which enables robot agent know which action in the task affects the result of the task the most, and by improving parameters of the action robot can maintain high success rate of tasks. Our framework let robot infer failure action of tasks using data from early task performance which are automatically recorded and retrieved with high-level data retrieval query interface. We evaluated our approach using mobile manipulation robot PR2 on daily assistive environment and task.

Keywords Home-assistant robot · Long-term memory · Bayesian network · Self-improving learning · Task execution monitoring

1 Introduction

Consider daily living tasks, for example, tidying up room, loading a washing machine, bringing a coke from fridge, taking the elevator. These tasks consist of complicated procedure, a lot of conditions with which robot must handle on runtime. Home assistant robot is required to perform these long-term tasks for deployment on

Y. Furuta (✉) · Y. Inagaki · K. Okada · M. Inaba
Graduate School of Information Science and Technology,
The University of Tokyo, 7-3-1 Hongo, Tokyo, Bunkyo-city 113-8656, Japan
e-mail: furushchev@jsk.imi.i.u-tokyo.ac.jp
URL: <http://www.jsk.t.u-tokyo.ac.jp/>

Y. Inagaki
e-mail: inagaki@jsk.imi.i.u-tokyo.ac.jp

K. Okada
e-mail: k-okada@jsk.t.u-tokyo.ac.jp

M. Inaba
e-mail: inaba@i.u-tokyo.ac.jp

real world. Though it is important to perform these tasks with success even once, it is also mandatory to let robots perform without failure every time they do. Robot system with failure recovery was first tackled in space robotics in early '90 [1], and has been challenged also on service robots these days [2]. We have also conducted research on robot system with failure detection and recovery, focusing on the aspect of generalized architecture [3], component abstraction and automation with high-level planning [4].

As discussed in [5], in typical robot architecture we must “*implement robot task by defining nominal operation, failure modes and the respective recovery sequences*” by hard-coding. Also planning system with failure tolerance, we can generate failure only from symbolic description defined in advance of robot deployment. In such robot system, robot agent can deal with failure written in advance though, actually robot still fails to execute tasks for various reasons which are not written in task level description.

To deal with all these failures, we found that not only defining recovery action and supposed failure, but also improving each actions in order not to let robot fail to perform tasks. To let robot agent learn better behavior to do tasks in each environment without expert, they can profit greatly from experience of earlier task executions. There are some work on self-improving robot behavior using experience of past task execution.

In this paper, we propose a novel framework to solve these problem using bayesian belief network to estimate independency between the task result and each actions done by robot, then improve the action that affects the result of task the most. In this approach robot can improve their actions in more generalized cases. Our approach consists of inference of actions that affects the result of the task the most, and improving robot behavior by modifying the parameters of the actions and maintain good success rate by training. This allows robot system to achieve their tasks continuously without any parameter adjustment by people with professional skill after deployment. We also propose logging framework of previous task execution which is associated with planning symbolic information and provides high-level query for data retrieval. In this self-improving framework, experience of previous task performance is used to infer specific actions needed to be improved. By this logging framework, robot agents easily select only data that is needed for learning system. By integrating our proposed system to existing fault-tolerant system of task planning and action management working for the well-known middle-ware for robot ROS (Robot Operating System), we can easily deploy this high-level framework to various robots.

In the remainder of this paper, we proceed as follows: we, at first, explain entire configuration of our framework in Sect. 2, and clarify what is the problem for that robot perform task continuously and key points of our solution in Sect. 3, then describe the technical detail that is needed to achieve our solution in Sects. 4 and 5. We show some experiments performing daily living tasks in home environment with our framework using Willow Garage's PR2 robot and evaluated our framework in Sect. 6. Finally, we conclude by discussing the result of the experiments, and future work in Sects. 7 and 8.

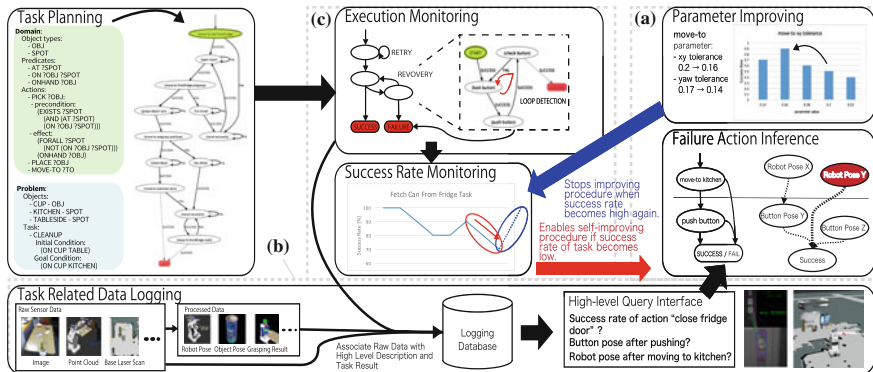


Fig. 1 System configuration diagram of our proposed framework

2 System Configuration

The entire configuration of our framework is depicted in Fig. 1. We have three points on this architecture: The first point of these (as depicted in Fig. 1a) is self-improving action management system based on inference of relationship between task result and action states in the task. With this component, robot can train their task performing by searching semi-optimal parameters with previous task execution and adapt to environment in which robot is deployed, which contribute for improving robot behavior by themselves and maintaining high success rate of tasks.

The second and third points are for making it work well self-improving system. The second is autonomous action related data logging system (Fig. 1b, described in Sect. 5) that decreases overhead of logging data and enables accurate data retrieval for learning on self-improvement of behavior by providing high-level queries.

The third point (Fig. 1c, described in Sect. 4) is task execution monitoring system with loop detection in task executive state machine that enables task execution with more safety and outputs the result of tasks for certain, which is important for estimating structure and parameters in failure action inference. This system is built on the top of our current task planning system with failure recovery [6] which enables a robot agent to continuously collect data with symbolic task description even if robots are in failure situation to perform task.

3 Self-improving Action Management System

In this section, we describe the algorithms to self-improve the robot behavior using experience of previous task performance. Our proposed self-improving framework can be divided into two parts: finding which action affect the result of the task the most (Sect. 3.1), and then searching good parameters for the action by training (Sect. 3.2).

3.1 Failure Action Inference

To construct a model for inference of relationship between the result of task and actions, we used Linear Gaussian Bayesian Belief Network (Bayesian Network) [7]. Bayesian Network is one of probabilistic graphical models that represents conditional dependencies among variables from data based on Bayes' theorem. As depicted in Fig. 2, each events are represented as nodes and dependencies between events are shown as edges, by which a bayesian network can represent the probabilistic causal relationships among events. One of the benefit to use Bayesian Network is that we can use set of data even if part of data is unobserved. As we mentioned above, robot cannot necessarily observe all states of environment, so we can get benefit from the feature to use all data collected. Learning the parameters of each nodes and the graph structure of a network enables the estimation of the causal dependencies among these events and the degree of dependencies [8].

In normal discrete bayesian network, we must discretize each sensor values and action parameters. Discretization of continuous values affects degree of independence between each actions and success rate, which is different for each task, so it is hard to define condition of discretization beforehand. We solved this problem by using linear gaussian node for model of bayesian network, which can handle continuous probability distribution.

In Linear Gaussian Bayesian Network, all nodes of graph are continuous and where all of the conditional probability tables (CPTs) are linear gaussians. For example, consider the graph consisting of node A with parent nodes B, C, D. The gaussian model of node A is described as Eq. 1.

$$\begin{aligned}
 P(a|b, c, d) &= \mathcal{N}(\mu_a, \sigma_a) \\
 &= \mathcal{N}(\beta_0 + \beta_1 b + \beta_2 c + \beta_3 d, \sigma_a)
 \end{aligned}
 \tag{1}$$

Normal Task Execution Procedure

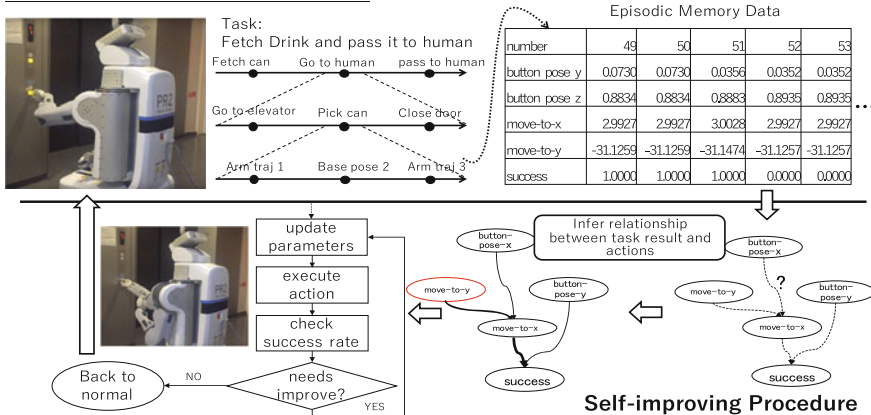


Fig. 2 Flow of self-improving behavior using probabilistic graphical model

where:

- a, b, c, d : values of each nodes
- $\mathcal{N}(\mu, \sigma)$: normal distribution with mean μ and variance σ^2
- β_k : coefficients which means weights of each values from parent nodes and bias.

The data we use in learning are environmental states related the executed actions which are automatically semantically extracted and recorded in early task performance (e.g. the relative button pose and robot pose for Fig. 2). And then we added a node whose value represents a result of task (1 for success or 0 for failed), which enables to construct graph that represents the causal relationship between each actions and the task result.

First of all, all nodes of a network are fully connected in LGBN, and we examine independency of each actions with structure estimation with simple constraint-based algorithm, then remove connections if no dependency is detected. In this phase, we also remove connections which is impossible when considering in chronological order. After estimated structure of network, we apply parameter estimation which inspects the degree of influence. This process is equivalent to estimate coefficients (β_k in Eq. 1). By tracing the edge of which has the highest coefficient from the node which represents the result of task, we can find the parent node (action) that affects result of the task the most. If the inferred parent node also has parents, we then use the most ascendant node to improve behavior.

Since this structure estimation algorithm has computational complexity $O(N^2)$ (where N is the number of nodes) in usual, or super-exponential in the worst case, it takes too long time to estimate structure of all actions of the task. We solved the problem by splitting entire graph to small size separated by “unified actions” [9]. At first we classify each actions into three types:

- (i) $Act_I : P_e \rightarrow S_{robot}$
- (ii) $Act_{II} : P_{ref} \rightarrow P_e$
- (iii) $Act_{III} : (P_e, S_{robot}) \rightarrow \text{Success} \mid \text{Failure}$

where:

- P_e : Parameters of environment on real world
- P_{ref} : Parameters of environment that is referred to by robot agent
- S_{robot} : States of robot

Act_I is an action that changes states of robot, but doesn't change environment (e.g. object recognition), while Act_{II} changes the environment (e.g. move base, reach out a hand, push button). Act_{III} is verification (e.g. check if object is grasped using gripper force sensor). If all actions in task sequences are divided into these types of actions, actions in planning system with failure recovery are mostly finished with verification (Act_{III}), and if a result of Act_{III} is “success”, then we can consider the action sequence (“unified actions”) from previous Act_{III} to this action succeeded. These actions are generated and classified automatically by planning.

3.2 Parameter Improving

Although robot agents now can find which action to be improved, still they cannot know which and how parameters of the action should be optimized. As depicted in Fig. 2, robot agent performs the task multiple times with gradually randomizing parameters of the inferred action. During this trial-and-error task performance, success rate monitoring module watches the success rate in the latest some task performance, and if the success rate is high enough, robot agent stops to randomize and self-improving phase is disabled with new parameters optimized to current environment.

4 Loop Monitoring in Action Transition in Planning with Failure Recovery [10]

Consider fetching coke in refrigerator or taking the elevator which includes manipulating home electrical appliances or facilities. To achieve these tasks, robot agent need to understand and estimate environment state. Also environmental states (e.g. object location, object poses, operation procedure to use home electrical appliances or tools and so on) changes every time and everywhere robot performs tasks. Robot agents that perform such a complex task need high level task description and fault-tolerant planning system. We use task compiler [5] based on PDDL (Problem Domain Description Language) [11] as task planning language to generate action sequence from task description. The output of task compiler is a state machine graph whose nodes are actions (e.g. move, pick, push), and edges are results of them, which can deal with failure conditions and recovery actions (Fig. 3).

As depicted in Fig. 1 for example, in a state machine especially one that includes recovery actions there can be closed loops in flow. Robot that follows loops infinitely does not return the result of task, which occurs miscounting of success rate and luck

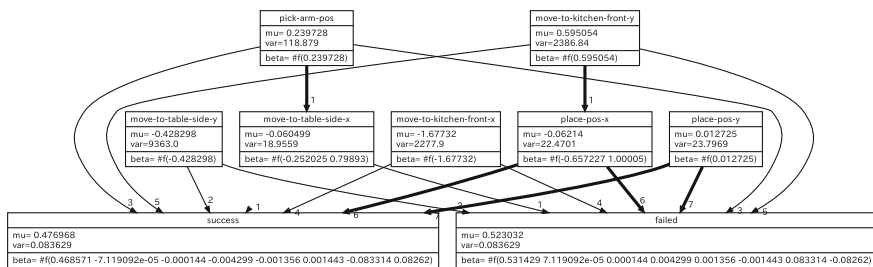
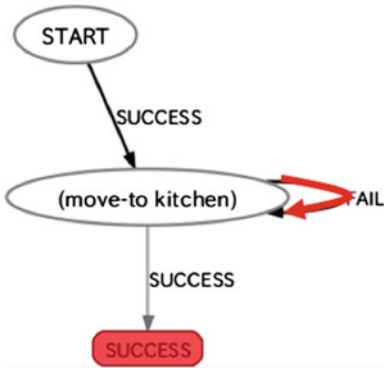
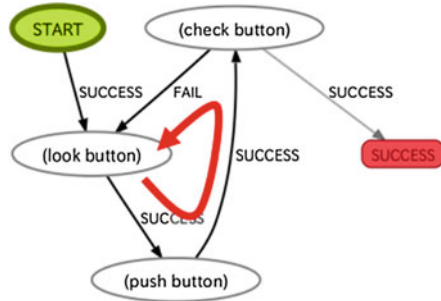


Fig. 3 Example graph which represents the relationship between task result and actions by inference. Each node are equivalent to action parameters. μ is a mean value of each parameter, σ is its variance. β represents coefficients of each μ . Numbers of arrows correspond to each coefficients



(a) State Machine that Represents Retry Loop



(b) State Machine that Represents Recovery Loop



(c) State Machine that Represents Mixed Loop

Fig. 4 Monitored looping transition in auto-generated task plan graph

of data to generate action inference graph. To avoid this problem and output the result of task even in such a situation, we also extended this fault-tolerant task planning system by integrating with execution monitoring module. This module traces each actions that robot executed and detects potential failure by monitoring three types of loops as follows:

- Retry Loop (Fig. 4a): Loop in which robots retry one action. (e.g. “move” action may recover from failure by just retrying.)
- Recovery Loop (Fig. 4b): Loop including recovery actions. (e.g. actions with perception may occur this type of loop, which consists of actions like “look object”, “do something with object”, “check if the action is done”)
- Mixed Loop (Fig. 4c): Loop including both types of loop.

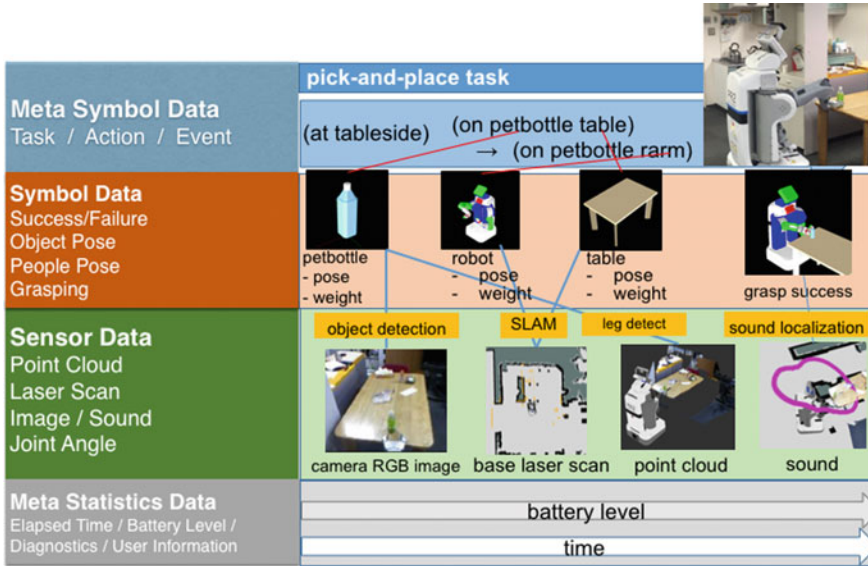


Fig. 5 Association between raw data in robot system and symbolic semantic data

This monitoring module is executed asynchronously and parallel with task execution.

5 Task Related Data Logging System

Consider simple pick-and-place task which contains “move base”, “pick object”, and “place object” actions. In self-improving phase, we generate action inference graph from experience of early robot task performance. For “move base” action, data needed to estimate dependency of this action on the result of task are “robot poses after move base action in pick-and-place task”. Instead of paying high cost for logging all raw sensor data related tasks and scanning all logged data for searching them that suits condition afterward, it is worth using high-level description to make data semantically grounded. We use symbolic data and predicate to extract necessary data from all sensor data (Fig. 5).

6 Experiment and Evaluation

We evaluate our proposed self-improving framework through experiment in which PR2 robot performs the task “fetch a coffee drink for human” in home environment. In this experiment, robot is asked by remote user to fetch coffee which robot agent knows coffee can is in the refrigerator.

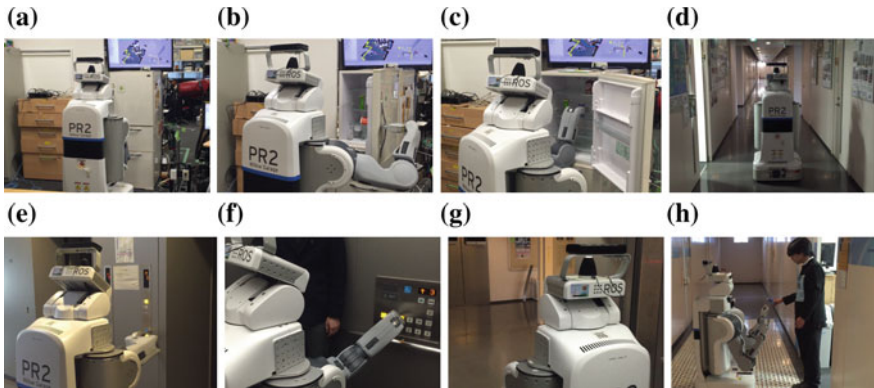


Fig. 6 Task sequence of experiment

At first, robot plans bring a coffee can manipulating a fridge (Fig. 6a–c), then moves to upper floor taking the elevator (Fig. 6d–f), then passes can to user (Fig. 6g, h).

There are some points in this task:

- The coffee can is in the refrigerator whose door is closed. Robot must open the fridge door, search a coffee drink, grasp can, and finally close the door.
- The human who wants a coffee is located at 8th floor, while the refrigerator is at 7th floor. So robot must take the elevator to pass a can to human.
- The elevator has its own state. Robot does not know the state directly, but can estimate it by reading the display of the elevator.

On task planning, robot agent knows some information, a map of these floors for localization and navigation and geometric information of immobile objects (such as elevator panel, furniture, refrigerator, etc.) and their locations in advance. These objects are represented as symbols and associated each other by predicates (e.g. “on”, “at”, etc.). The goal state of this task is represented as the set of predicates and symbols: (AND (ON COFFEE HUMAN) (NOT (OPEN FRIDGE-DOOR))). Each actions (e.g. “open door”, “move base”, “push elevator button”, etc.) are represented as the transition among states.

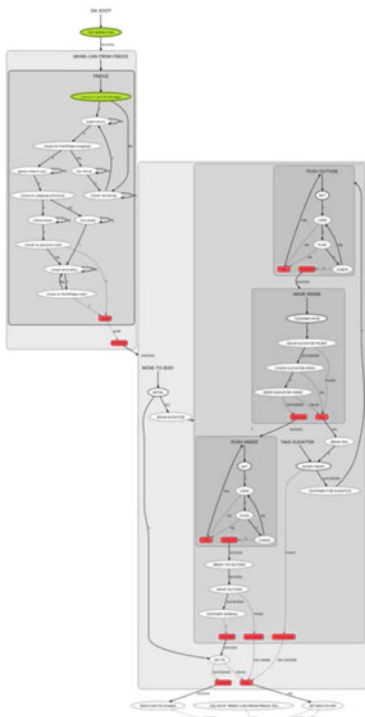
For example, the action “close door” is defined as below:

```
(CLOSE-DOOR OBJ?)
- PARAMETER: OBJ? (type object)
- PRECONDITION: (OPEN OBJ?)
- EFFECT: (NOT (OPEN OBJ?))
```

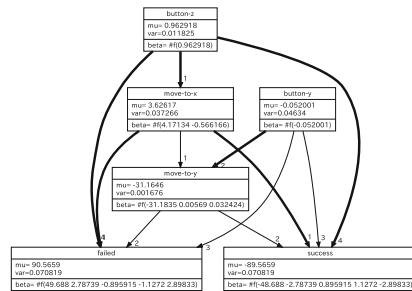
When robot is in the state (AND (ON COFFEE LEFT-ARM) (OPEN FRIDGE-DOOR)), then by executing (CLOSE FRIDGE-DOOR) the state changes into (AND (ON COFFEE LEFT-ARM) (NOT (OPEN FRIDGE-DOOR))).

To evaluate proposed methods, we conducted this experiment including 85 task performance in 3 months. While deploying robot in the environment, robot sometimes failed to do “push-elevator-button” (Act_{III}) action before taking the elevator. Also in this task, we found that robot failed the task some times with loops, which were detected as failure by execution monitoring module and robot achieved the task performance without stuck. When the success rate becomes lower than limit (set to 80% in this experiment), robot agent tries to self-improve behavior.

Figure 7b shows the result of failure action inference. To generate this inference graph of failure actions, the results of “push-elevator-button” action as task results, data related to actions following “push-elevator-button” action (in this graph “move-to” and “push-button” action) were retrieved from logging database (“robot poses after moving base to front of elevator call panel” which is represented as 2-dimensional vector in cartesian space, and “button poses relative to the robot” which is also 2-dimensional vector. x-component is ignored because of no meaning for detecting button panel pose). The most significant coefficient value was for action “move-to”, which has two parameters “xy_goal_tolerance” (the tolerance for the con-



(a) State Machine of Entire Task Auto-Generated by Task Planning



(b) Graph of Failure Action Inference



(c) Success Rate based on the Latest 10 Trials of Fetching Drink Task

Fig. 7 Result data of experiment

troller in the x & y distance when achieving a goal) and “yaw_goal_tolerance” (the tolerance for controller in the yaw when achieving a goal) to be improved. Robot agent performs the task multiple times with gradually randomizing these parameters, then robot could keep high success rate of the task with changing parameters from 0.20 to 0.14 for “xy_goal_tolerance”, and from 0.17 to 0.16 for “yaw_goal_tolerance”. Figure 7c shows the actual success rate of achieving this task while this self-improving phase. We achieved high success rate by exploring semi-optimal parameters of the failure action inferred by logged data of early task performance.

7 Related Work

Research on system for self-improving robot behavior has been long conducted before. Mahadevan et al. allow robot to learn parameters of actions using reinforcement learning in pushing boxes task [12]. Robot with their framework can get the best parameters while performing trial-and-error behavior evaluated by rewarding function that is pre-defined. While this has the same goal as we have, there was a problem when we apply their framework to robot performing daily assistive task, which is that there are large search spaces needed to be solved with reinforcement learning.

Mahadevan also mentioned this problem and solved by integrating with behavior-based system such as Subsumption architecture [13] in pushing boxes task domain. However it is hard to define home assistive task including complex procedure and deeply dependent on states of environment with such a behavior-based context. We instead adapt classical planning-based architecture, and reduce search spaces of parameters with inference of relationship between actions and result of task.

Stulp et al. proposed the concept of Action-Related Place as the probability distribution of successful manipulation using previous task execution and Monte Carlo based predictive model targeting for pick-and-place task with moving base of robot [14]. The benefit of this framework is robustness for uncertainty of environment by handling parameters statistically, and this is effective against especially pick-and-place task where geometric relation between manipulating objects and robot affects the success rate of the task a lot. This includes parameter improving using early task performance, but only “move” action is improved and we achieved more generalized self-improving method for various home assistant tasks.

In robot system proposed by Kirsch [15], robot collects data while performing task and improves each of their action parameters by machine learning methods. Though this is similar to our work, parameters are improved only if the action which robot fails to do the task and the action to be improved are the same.

Maintaining high success rate of tasks using action parameters with planning-based architecture was researched by Winkler et al. [16]. They collect data of early task performance as experience and construct decision tree from which predict the result of behavior that robot is performing next using parameters generated by task planning. Then if the planned actions are predicted to fail, robot re-plans their actions

that achieve the task. This is similar with our work, but we also allow robot system to improve parameters of each actions instead just avoiding expected failure actions.

8 Summary and Conclusion

In this paper, we proposed a novel approach for inferring failure action and improving the parameters of the action based on data recorded during early task performance. We confirmed the necessity of our approach and the algorithm to construct graph that represents the independencies between actions and the result of tasks from data of earlier task executions using bayesian belief network that can deal with continuous variables. To let robot agent infer failure action by this machine learning method, we integrate execute monitoring module that detects potential failure loop in state machine of the task and task related data logging module into existing planning-based robot system that can execute complex home assistant task autonomously.

Then our framework is applied and evaluated on a real PR2 robot in home environment which can change every day by various reasons including people, other robots living with. we deployed PR2 performing long-term task that includes manipulation of home electrical appliances and taking the elevator. Through long-term task executions, we confirmed that after so many as 85 times of task executions, robot still can perform tasks with high success rate of 90 % and no maintenance by professional skills.

References

1. Papadopoulos, E., Dubowsky, S.: Failure recovery control for space robotic systems. *Am. Control Conf.* **1991**, 1485–1490 (1991)
2. Bohren, J., Rusu, R.B., Gil Jones, E., Marder-Eppstein, E., Pantofaru, C., Wise, M., Msenlechner, L., Meeussen, W., Holzer, S.: Towards autonomous robotic butlers: Lessons learned with the pr2. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 5568–5575 (2011)
3. Yamazaki, K., Ueda, R., Nozawa, S., Kojima, M., Okada, K., Matsumoto, K., Ishikawa, M., Shimoyama, I., Inaba, M.: Home-assistant robot for an aging society. *Proc. IEEE* **100**(8), 2429–2441 (2012)
4. Furuta, Y., Inagaki, Y., Kakiuchi, Y., Okada, K., Inaba, M.: Tidyup task sequence using pr2 by irt home assistant robot. In: The 31th Annual Conference on Robotics Society of Japan, pp. 112–02 (2013)
5. Okada, K., Kakiuchi, Y., Azuma, H., Mikita, H., Murase, K., Inaba, M.: Task compiler: transferring high-level task description to behavior state machine with failure recovery mechanism. In: *ICRA Workshop on Combining Task and Motion Planning* (2013)
6. Kakiuchi, Y., Furuta, Y., Mikita, H., Nozawa, S., Okada, K., Inaba, M.: Automatically generating system of hierarchized interruptible task execution machine for assistant robot. In: *Proceedings of The 19th Robotics Symposia*, vol. 3, pp. 631–636 (2014)
7. Pearl, J., Russell, S.: Bayesian networks. University of California, Computer Science Department (1998)

8. Heckerman, D., Geiger, D., Chickering, D.: Learning bayesian networks: the combination of knowledge and statistical data. *Mach. Learn.* **20**(3), 197–243 (1995)
9. Saito, M.: Construction of service robot system for autonomous navigation in multistory building with learning and inference mechanisms. Master's thesis (2012)
10. Furuta, Y., Kakiuchi, Y., Mikita, H., Ueda, R., Okada, K., Inaba, M.: Onsite interruptible action management system on daily assistant robot. In: 2014 JSME Conference on Robotics and Mechatronics, pp. 1P2–Q06 (2014)
11. Mcdermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D., Wilkins, D.: Pddl—the planning domain definition language. Technical Report TR-98-003, Yale Center for Computational Vision and Control (1998)
12. Mahadevan, S., Connell, J.: Automatic programming of behavior-based robots using reinforcement learning. *Artif. Intell.* **55**(2–3), 311–365 (1992)
13. Brooks, R.: A robust layered control system for a mobile robot. *IEEE J. Robot. Autom.* **2**(1), 14–23 (1986)
14. Stulp, F., Fedrizzi, A., Beetz, M.: Action-related place-based mobile manipulation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009, pp. 3115–3120. IEEE (2009)
15. Kirsch, A.: Robot learning languageintegrating programming and learning for cognitive systems. *Robot. Auton. Syst.* **57**(9), 943–954 (2009)
16. Winkler, J., Beetz, M.: Robot action plans that form and maintain expectations. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany (2015). Accepted for publication

View-Based Teaching/Playback with Photoelasticity for Force-Control Tasks

Yoshinori Nakagawa, Soichi Ishii and Yusuke Maeda

Abstract We study a novel robot programming method that uses the view-based approach: “view-based teaching/playback.” This method directly uses images for robot programming and can accommodate itself to changes of task conditions. However, our previous view-based teaching/playback cannot perform force-control tasks; for example, it cannot deal with pressing objects against walls, in which view of images does not change. In this paper, we extend the view-based teaching/playback so that it is applicable to force-control tasks using photoelasticity. In the experiment, the extended view-based teaching/playback succeeded in wall-pressing tasks.

Keywords Robot programming · View-based approach · Principal component analysis · Neural network · Photoelasticity

1 Introduction

In the factory, many industrial robots are used in order to increase efficiency of production. It is necessary for operators to teach the details of the operations of the robots in advance. This is called robot programming.

As a representative example for robot programming, teaching/playback has been adopted widely. In this method, a human worker operates a robot to memorize its joint variables and playback using the data. This method enable to playback robots precisely in unchanging environment. However, the robot cannot be operated adaptively when the task condition is changed. To solve the problem, it is necessary to enable the system to recognize the change of the task condition.

Y. Nakagawa (✉) · S. Ishii
Division of Systems Integration, Graduate School of Engineering,
Yokohama National University Kanagawa, Yokohama, Japan
e-mail: nakagawa-yoshinori-rz@ynu.jp

Y. Maeda
Division of Systems Research, Faculty of Engineering,
Yokohama National University Kanagawa, Yokohama, Japan

Visual sensing is one of the methods to recognize the task condition. For example, by making object models and extracting geometrical features from input images, objects can be detected and localized with a camera. However, making models for each object is not trivial labor. Moreover, it is difficult to make models of some objects that have no distinct geometrical features. Therefore, model-free robot programming methods that can cope with changes of task conditions are demanded.

Recently, we proposed “view-based teaching/playback” using the view-based approach (or appearance-based approach) [1] for model-free robot programming [2, 3]. In our proposed method, it is not necessary to create a model of the object because of using images directly. It was successfully applied to manipulation tasks with un-fixed task conditions. However, it is not applicable to force-control tasks, in which there may be no visual changes. In this paper, we therefore try to realize view-based teaching/playback of force-control tasks by visualizing the force information using the photoelasticity.

2 Related Work

There are many previous studies on robot programming by human demonstrations. For example, Abdo et al. presented a practical approach to learn manipulation skills, including preconditions and effects, based on only a small number of teacher demonstrations [4]. Pastor et al. proposed a method in which a robot learned a pick-and-place operation and a water-serving task by using the dynamic movement primitives [5]. Jen et al. used virtual reality (VR) to perform demonstrations for robot programming [6].

To obtain human demonstrations for robot programming, model-based image processing is widely used. On the other hand, previous studies that use view-based image processing for robot programming are very few. Zhang et al. proposed a method to determine gripper position using a neuro-fuzzy controller based on the view-based approach [7]. Zhao presented a view-based visual servoing for robot positioning [8]. Shibata and Iida showed that a mobile robot could learn appropriate actions to reach and push a box using view-based reinforcement learning [9].

3 Outline of View-Based Teaching/Playback

Just like the conventional teaching/playback method, our view-based teaching/playback is divided into two stages, the teaching phase and the playback phase, as follows:

Teaching Phase (Fig. 1a)

An operator moves a robot to perform a task. During this time, the movements of the robot and its corresponding scene images are recorded. From the teaching

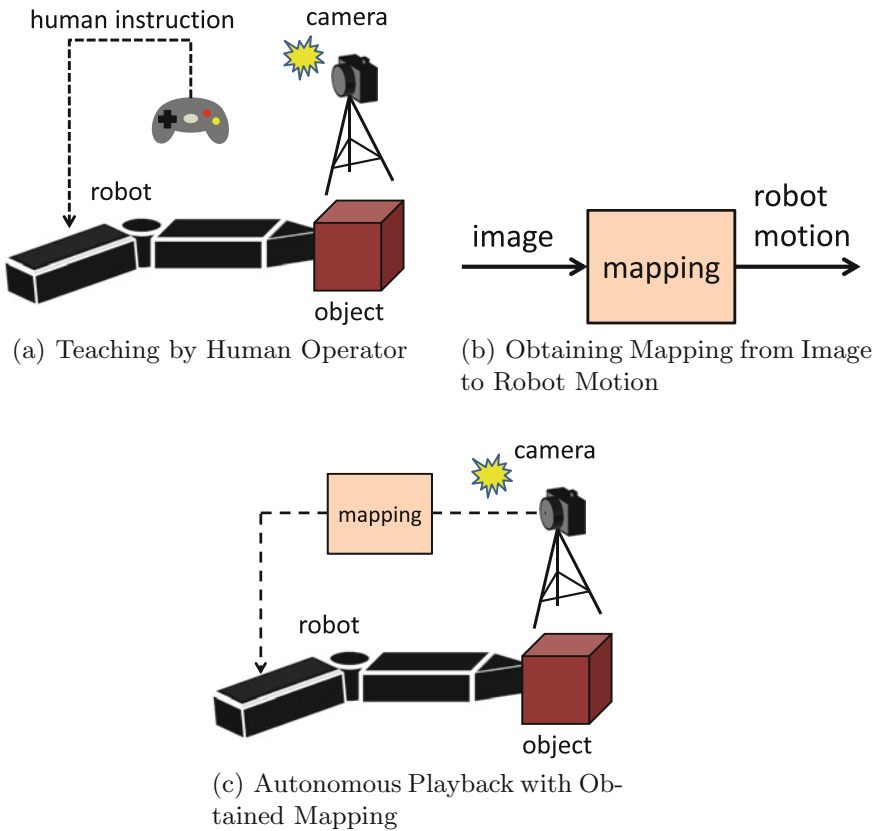


Fig. 1 Outline of view-based teaching/playback

data, a neural network that maps robot movements onto scene images is attained (Fig. 1b).

Playback Phase (Fig. 1c)

Robot motion is outputted based on the mapping which was obtained in the teaching phase. If the neural network functions properly, the task of the teaching phase is reproduced.

Our view-based teaching/playback can obtain a mapping from multiple demonstrations with different task conditions (e.g., the initial position of the object is perturbed) in the teaching phase. In this case, the task can be executed under conditions that were not taught in the teaching phase by the generalization capability of the neural network. The details of the teaching phase and the playback phase will be shown below.

3.1 Teaching Phase

3.1.1 Teaching by Operator

An operator commands the movement with an input device such as a joystick in order to complete a task. All the movements and its corresponding scene images are recorded. Teaching of the task can be repeated in different task conditions.

3.1.2 Image Compression by Principal Component Analysis

Since the image has enormous pixel data, it is not realistic to use the raw pixel data as input for the mapping neural network. In other words, we need image compression. Similar to Zhang et al. [7], we compress images by performing a principal component analysis (PCA) [10] for all images in the teaching phase.

3.1.3 Mapping by Neural Network

To map the robot movements onto the scene images, we use a three-layered feed-forward neural network as shown in Fig. 2. We input the factor scores of the scene image ($FS(t)$), the current robot configuration ($Config(t)$) and the target robot configurations in the previous n steps ($Config_T(t - \Delta t)$, $Config_T(t - 2\Delta t)$, ..., $Config_T(t - n\Delta t)$) to the neural network, and its output should be the target robot configuration ($Config_T(t)$).

Fig. 2 Neural network for teaching/playback

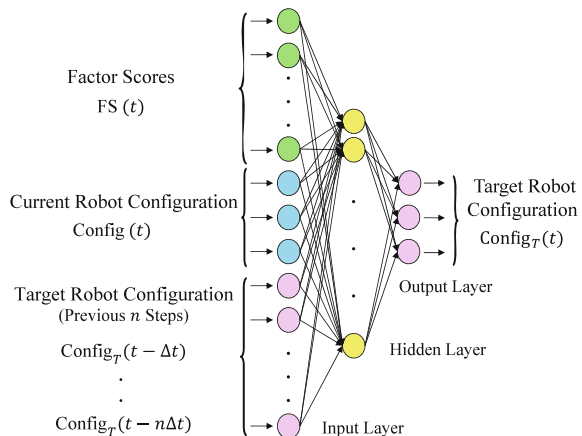
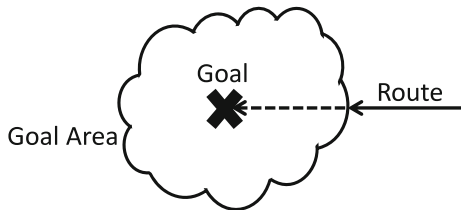


Fig. 3 Goal condition

3.2 Playback Phase

In the playback phase, a robot motion is determined by the output of the neural network. We should define a terminate condition to finish the playback at the goal position. However, the object configuration at the goal is not given explicitly because our method is view-based. Therefore, it is necessary to set a view-based terminate condition.

We use the RMS error of the factor scores of the current image and those of the teaching image at the goal. RMS error ΔE_S can be written as follows:

$$\Delta E_S = \sqrt{\frac{\sum_{i=1}^N |\text{FS}(t)_i - \text{FS}_{\text{Goal}i}|^2}{N}}, \quad (1)$$

where $\text{FS}(t)_i$ and $\text{FS}_{\text{Goal}i}$ are the i -th factor scores of the current image and the goal image, respectively, and N is the number of the factor scores used in the image compression.

The factor scores of the current image implicitly include the information of the position and orientation of the object. Thus, the terminate condition can be defined as the following inequality using a threshold T_S :

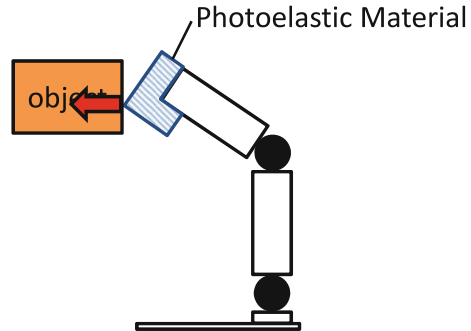
$$\Delta E_S < T_S. \quad (2)$$

However, even if this inequality becomes satisfied, the object may be able to approach the goal position further by continuing the playback. For example in Fig. 3, Eq. (2) becomes satisfied at the edge of the goal area. When the playback goes on along the route, the object would be closer to the goal position in the goal area. Therefore, even after Eq. (2) is satisfied, we continue playback as far as $\Delta E_S(t) < \Delta E_S(t - \Delta t)$.

4 Visualization of Force Information Using Photoelasticity

Photoelastic materials show temporarily optical anisotropy and cause birefringence phenomenon when external force is applied. Therefore it is possible to visualize the internal stress state in transparent photoelastic objects (Fig. 4). This can be observed

Fig. 4 Visualization with photoelastic method



by making photoelastic materials sandwiched between two polarizing plates through which the light travels.

In this study, we use the photoelasticity to apply our view-based teaching/playback to force-control tasks. Attaching a photoelastic part to the robot, we obtain force information at the part visually by a polarized light source and a camera with a polarizing lens filter. Note that we do not have to analyze photoelastic images explicitly because we use the view-based approach.

5 Verification Experiment

5.1 Experimental Setup

We describe our experimental setup for view-based teaching/playback with a two-joint finger. A schematic view of the setup is shown in Fig. 5, and the numbered devices are explained in Table 1. Definition of the symbols are shown in Fig. 6. Table 2 shows kinematic parameter values.

A photoelastic part made of epoxy resin is bolted to the tip of the finger. We place a liquid crystal display under the robot hand. This is because the light from the liquid crystal display polarizes.

A camera is placed above the finger and captures grayscale scene images at 80 [FPS]. A real-time OS (ART-Linux) is used in the PC to control the finger. Another PC obtains force data at about 1 (ms) interval from a force sensor. Note that the force sensor is used only for the evaluation of the experimental results and not used for the operation of the view-based teaching/playback.

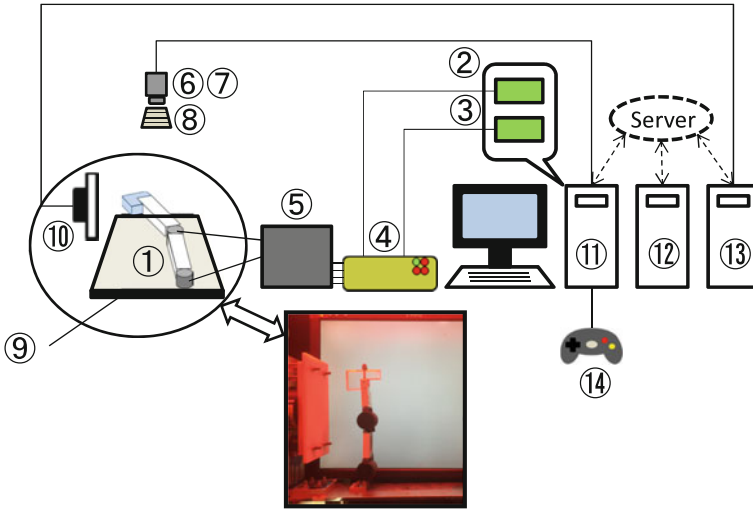


Fig. 5 Experimental setup

Table 1 Equipment

1. Servomotor	YR-KA01-A000 (YASKAWA)
2. D/A Board	PCI-3341A (Interface)
3. Counter Board	PCI-6205C (Interface)
4. Relay Box	
5. Servo driver board	JARCH-ZKD01 (YASKAWA)
6. CCD camera	FL2-03S2M (Point Grey Research)
7. Lens	HD880MIR (SPACECOM)
8. Polarized film	
9. LCD (Light Source)	Dell E176FP - LCD monitor
10. Force Sensor	67M25A25-I40 (NITTA)
11. PC for control	OS: ART-Linux 2.6 (Ubuntu 10.04) CPU: Intel Core 2 Duo E8400 3.00 GHz
12. PC for offline calculation	OS: Ubuntu 10.04 CPU: Intel Xeon E5645 2.40 GHz
13. PC for force sensor	OS: Ubuntu 10.04 CPU: Core i7-4770 3.40 GHz
14. Gamepad	F310r (Logicool)

Fig. 6 Definition of angles and pressing force

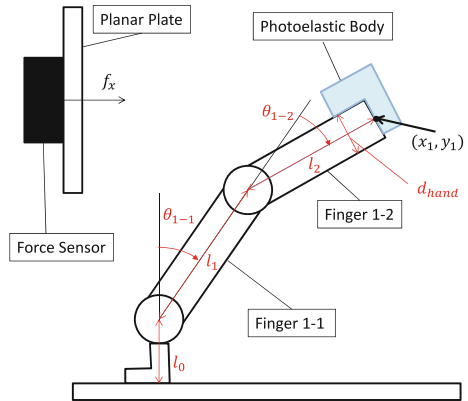


Table 2 Robot parameters

l_0	l_1	l_2	d_{hand}
30 (mm)	81 (mm)	66 (mm)	10 (mm)

5.2 Target Task

Here our target task is wall-pressing; the finger presses a planar plate by a designated force. To maintain pressing, playback is continued for 5 s after the terminate condition is satisfied.

5.3 Application of View-Based Teaching/Playback with Photoelasticity

In order to perform the target task, the parameters of the view-based teaching/playback are determined; $n = 2$, $N = 50$, and $T_S = 3.0$. The number of the neurons in the input layer of the neural network is 56, that in the hidden layer is 50, and that in the output layer is 2.

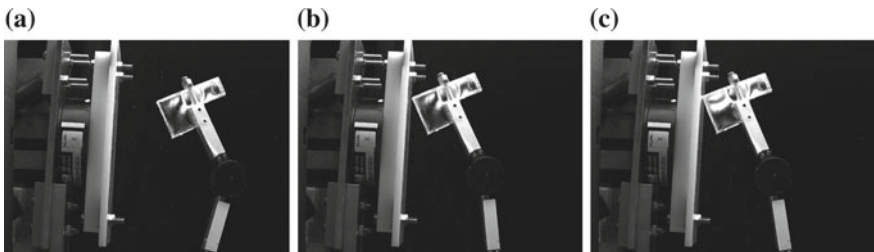


Fig. 7 Teaching images with photoelasticity

5.4 Teaching/Playback with Single Demonstration

We firstly conducted view-based teaching/playback of wall-pressing using a single demonstration. Teaching and playback images are shown in Figs. 7 and 8, respectively where changes of the brightness of the fingertip part caused by the photoelasticity can be found. Figure 9 shows experimental results of our view-based teaching/playback (three times) and the conventional teaching/playback, in which joint angles in teaching phase were simply played back. Table 3 summarizes the results of the pressing force during the last 5 s.

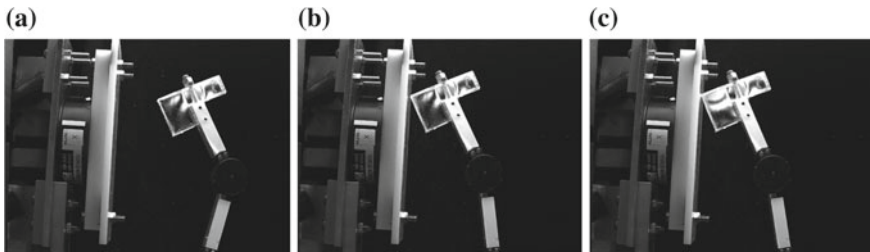


Fig. 8 Playback images with photoelasticity

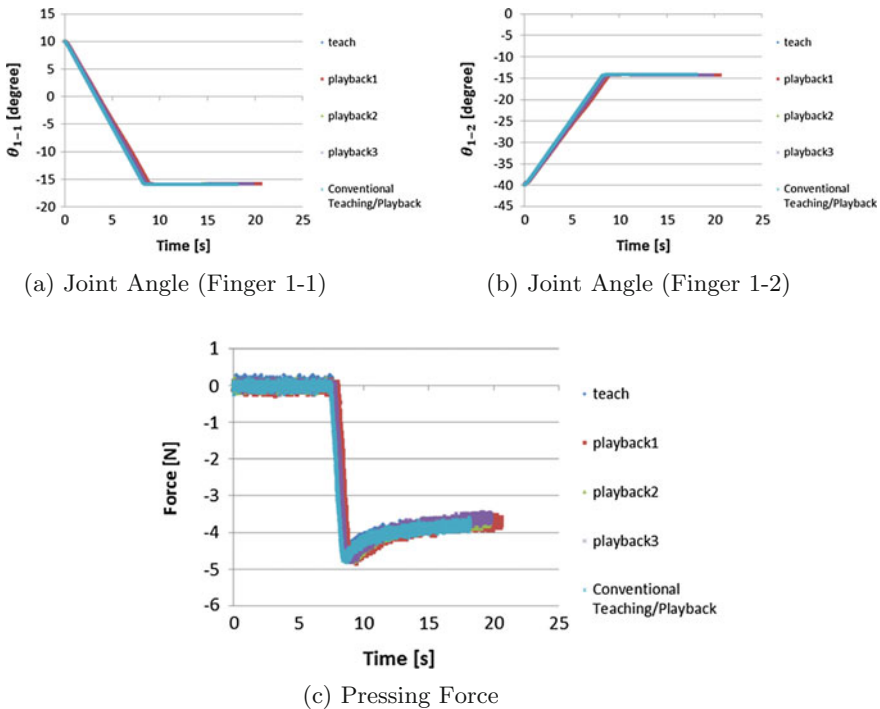


Fig. 9 Results of wall-pressing with single demonstration

Table 3 Results of pressing force (Single Demonstration)

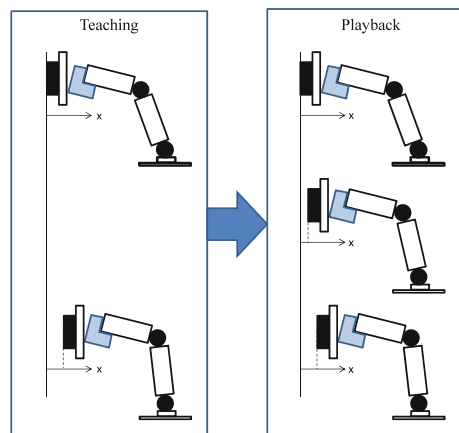
	Average (N)	Standard deviation (N)
Teaching	-3.81	0.08
Playback 1	-3.79	0.08
Playback 2	-3.78	0.08
Playback 3	-3.71	0.09
Conventional Teaching/Playback	-3.88	0.07

The results show that our proposed method achieved the wall-pressing tasks with small force errors, which is comparable to the conventional teaching/playback.

5.5 Teaching/playback with Multiple Demonstrations

The advantage of our view-based teaching/playback should be the adaptability to changes of task conditions. To verify the advantage we deal with pressing of the wall whose position is not fixed. The robot should apply a designated pressing force regardless of the wall position. For this purpose, we performed multiple demonstrations in the teaching phase. Even if the wall position changes, the generalization capability of the neural network would help the task achievement (Fig. 10). Specifically, we perform two demonstrations with the wall at 0.0 (mm) and 2.0 (mm). Playback was performed at the same two positions (0.0 (mm) and 2.0 (mm)), and the middle position between them (1.0 (mm)).

Fig. 10 Wall positions in teaching/playback



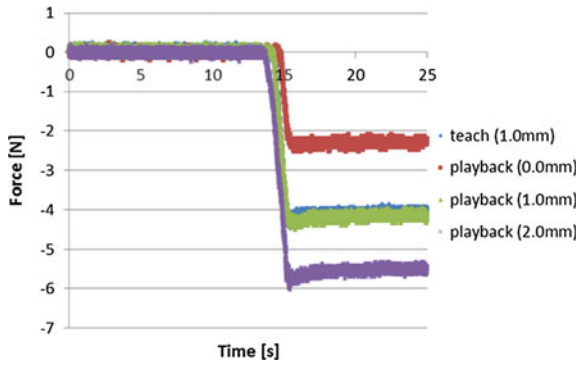
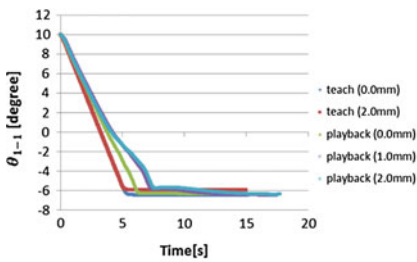


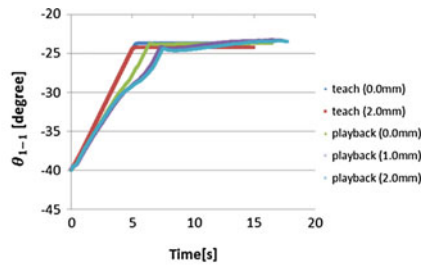
Fig. 11 Pressing force of conventional teaching/playback

Table 4 Results of pressing force of conventional teaching/playback

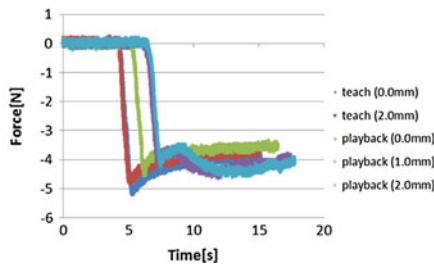
	Average (N)	Standard deviation (N)
Teaching (1.0 (mm))	-4.06	0.06
Playback (0.0 (mm))	-2.30	0.06
Playback (1.0 (mm))	-4.15	0.06
Playback (2.0 (mm))	-5.51	0.06



(a) Joint Angle (Finger 1-1)



(b) Joint Angle (Finger 1-2)



(c) Pressing Force

Fig. 12 Results of wall-pressing with two demonstrations

Table 5 Results of pressing force with view-based teaching/playback photoelasticity (Multiple Demonstrations)

	Teaching		Playback		
	0.0 (mm)	2.0 (mm)	0.0 (mm)	1.0 (mm)	2.0 (mm)
Average (N)	-3.97	-3.81	-3.58	-4.15	-4.29
Standard deviation (N)	0.08	0.07	0.07	0.13	0.13

We also performed the conventional teaching/playback for comparison; teaching was performed at 1.0 (mm), and playback was performed at three points (0.0 (mm), 1.0 (mm), and 2.0 (mm)). The results of the conventional teaching/playback are shown in Fig. 11. The pressing force during the last 5 s is summarized in Table 4. As easily expected, small force error was achieved at 1.0 (mm), but large errors were found at 0.0 (mm) and 2.0 (mm).

The results of the view-based teaching/playback with photoelasticity are shown in Fig. 12. The pressing force during the last 5 s are summarized in Table 5. Not only the playback at the same position with the demonstrations, but also the playback at the middle position achieved wall-pressing with small force errors. This is the effect of the generalization capability of the neural network. Thus the experiments showed the adaptability of the view-based teaching/playback with photoelasticity in the wall-pressing task.

6 Conclusion

In this study, we aimed to realize a force-control task by the view-based teaching/playback, by visualizing the force information using the photoelasticity. In the experiments, it was shown that a wall-pressing task was achieved successfully. Moreover, the view-based teaching/playback showed higher adaptability to the changes of the wall positions than the conventional teaching/playback. Future work should address its application to more complex force-control tasks.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number JP24560286 and JP15K05890.

References

1. Matsumoto, Y., Inaba, M., Inoue, H.: View-based navigation using an omniview sequence in a corridor environment. *Mach. Vis. Appl.* **14**(2), 121–128 (2003)
2. Maeda, Y., Nakamura, T.: View-based teaching/playback for robotic manipulation. *ROBOMECH J.* **2**(1), 2 (2015)

3. Maeda, Y., Moriyama, Y.: View-based teaching/playback for industrial manipulators. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 4306–4311 (2011)
4. Abdo, N., Kretzschmar, H., Spinello, L., Stachniss, C.: Learning manipulation actions from a few demonstrations. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1268–1275 (2013)
5. Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: Learning and generalization of motor skills by learning from demonstration. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 763–768 (2009)
6. Jen, Y.H., Taha, Z., Vui, L.J.: VR-Based robot programming and simulation system for an industrial robot. *Int. J. Ind. Eng.: Theory, Appl. Practice* **15**(3), 314–322 (2008)
7. Zhang, J., Knoll, A., Schmidt, R.: A neuro-fuzzy control model for fine-positioning of manipulators. *Robot. Auton. Syst.* **32**(2), 101–113 (2000)
8. Zhao, Q., Sun, Z., Sun, F., Zhu, J.: Appearance-based robot visual servo via a wavelet neural network. *Int. J. Control Autom. Syst.* **6**(4), 607–612 (2008)
9. Shibata, K., Iida, M.: Acquisition of box pushing by direct-vision-based reinforcement learning. In: SICE 2003 Annual Conference, vol. 3, pp. 2322–2327. IEEE (2003)
10. Abdi, H., Williams, L.J.: Principal component analysis. *Wiley Interdiscip. Rev.: Comput. Stat.* **2**(4), 433–459 (2010)

Discovering the Relationship Between the Morphology and the Internal Model in a Robot System by Means of Neural Networks

Angel J. Duran and Angel P. del Pobil

Abstract Supervised machine learning techniques have proven very effective to solve the problems arising from model learning in robotics. A significant limitation of such approaches is that internal models learned for a specific robot are likely to fail when transferred to a robot with a different morphology. One of the challenges to relate the morphology and the internal model is the difference in the number of parameters that define them. We propose three neural network architectures for solving this problem, along with a case study to evaluate their performance, namely saccadic movements in a robotic head. We generate a huge dataset to test the performance of the proposed architectures. Our results suggest that the best solution is provided by the parallel neural network, due to the fact that the trained weights are independent of one another.

Keywords Morphology · Internal Model · Neural networks

1 Introduction

The widespread use of neural networks for troubleshooting in robot control is undoubtable when adaptability of the internal model of the robot is required. In that context, on-line adaptation of models is a valuable strategy because it can capture non-stationarities in the mechanical properties of the robotic system [1]. From an operational point of view there are two approaches to deal with the adaptation of the

A.J. Duran (✉) · A.P. del Pobil (✉)
Robotic Intelligence Lab, Universitat Jaume I, Castellon, Spain
e-mail: abosch@uji.es

A.P. del Pobil
e-mail: pobil@uji.es
URL: <http://robinlab.uji.es>

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_61

internal model of the robot: adaptive control [2] and model learning [3]. Whereas the first one uses on-line parameter identification, the second one uses supervised learning. The aim of this paper is focused on this second group. There are multiple methods for all kind of robotic systems to learn their adaptive controllers. They typically propose a particular robot immersed in a certain environment that is able to learn its internal model when it executes a task. The main problem of these methods is that when the environment, the task or the robot morphology are changed significantly, the system has to learn again. Even if the new robotic system is similar, it is necessary to start a new training process.

Frequently, the training task in the model learning approaches has a high time cost because the initial information about the system is poor. Learning an internal model of a plant is a regression problem where training samples are obtained from the state and controls of the plant along time [1]. There are multiple options to solve this question, among them multilayer feedforward neural networks. These machine learning tools are capable of approximating any measurable function to any desired degree of accuracy [4]. The main characteristics of a neural network can be explained in two parts, i.e. its architecture and functional properties [5]. The architecture is related to the network topology and the functional properties are concerned with how learning takes place to decompose complex information into its fundamental elements, i.e. weights. Therefore, the real world model interaction with the robot system is stored in the weights of the neural network for a given network topology. If we keep the same environment and behavior of the robot system, but we modify its morphology, the neural network weights will have to be changed by a new training process. We can keep fixed the network topology so that the change in the robot morphology only alters the neural network weights.

The aim of this paper is to propose several procedures to relate the changes in the neural network weights with the alteration of the morphology of a robot system, evaluating which one presents a better performance. To achieve this goal, we propose a practical case study in simulation which allows us to use different kinds of machine learning tools in order to learn the relationship between the morphology of a robot system and the weights learned by a neural network working as an adaptive controller.

We propose a model to take advantage of knowledge previously learned by similar robot systems (Sect. 2). However, this model requires solving a high dimensional regression problem, therefore we propose a case study (Sect. 3) which is useful to generate a big dataset. This is based on the simulation of the learning behavior for many robotic systems (Sect. 4). To extract the knowledge from this dataset we present three possible neural networks (Sect. 5). Finally, we compare the performance of these networks when solving this learning problem (Sect. 6).

2 Model Description

2.1 The Relationship Between Morphological and Internal Model Parameters

The methodology that is proposed in this article, considers a robotic system (i) immersed in a stable environment. The sensors of the robot transform the data from the environment and generate an input into the robotic system which processes the inputs and generates the outputs to the robot actuators modifying the environment. This data processing follows an aim, so it is possible to identify a behavior of the robotic system for an external observer. A schema of the considered system is drawn in Fig. 1. The robot behavior emerges from the relationship between morphology, intelligence and the environment [6]. In this schema, several data flux and data transformations can be observed. The flux of data from the environment to the system sensors is called E_x . This is generated by the environment and it just depends on the state of the environment. This data flux is transformed by the robot sensors into other data (X_i). This transform is a function of several parameters Γ_i^s that characterize the robot system (i). The X_i flux is processed using the system internal model. The result of this process is another data flux called Y_i . Then, the actuators transform this flux considering several morphological parameters Γ_i^a in such a way that the new signal E_y modifies the environment. Both E_x and E_y are independent of a particular robot system. If the robot intends to develop a behavior in a particular environment, it has to understand the E_x data flux and it has to be able to generate the adequate E_y sig-

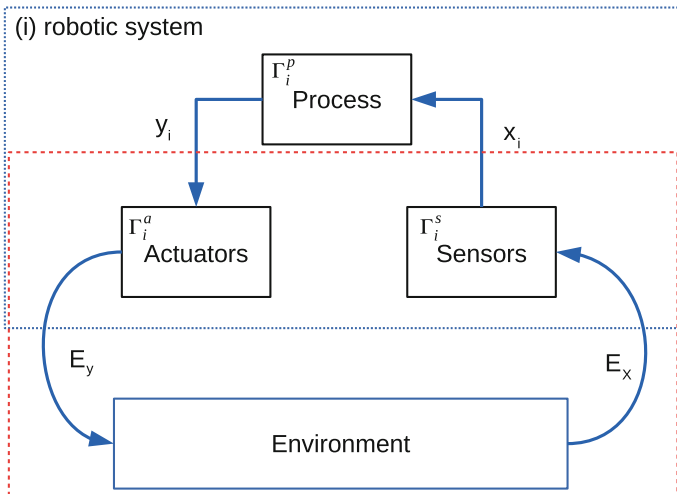


Fig. 1 The system schema, a representation of the data flux and transformations between the environment and the robot system

nals. These relations can be expressed in a mathematical way: $\mathbf{X}_i = \mathcal{I}(\Gamma_i^s, \mathbf{E}_x)$; $\mathbf{Y}_i = \mathcal{M}(\Gamma_i^p, \mathbf{X}_i) \equiv \mathcal{M}^*(\Gamma_i^s, \Gamma_i^p, \mathbf{E}_x)$; $\mathbf{E}_y = \mathcal{F}(\Gamma_i^a, \mathbf{Y}_i) \equiv \mathcal{F}^*(\Gamma_i^s, \Gamma_i^p, \Gamma_i^a, \mathbf{E}_x)$ then:

$$\begin{aligned} \mathbf{E}_x &\xrightarrow{\mathcal{I}} \mathbf{X}_i \xrightarrow{\mathcal{M}} \mathbf{Y}_i \xrightarrow{\mathcal{F}} \mathbf{E}_y \\ \mathbf{E}_y &= \mathcal{F}^*(\Gamma_i, \mathbf{E}_x) \end{aligned} \quad (1)$$

$\Gamma_i = \{\Gamma_i^m, \Gamma_i^p\} \mid \Gamma_i^m = \{\Gamma_i^a, \Gamma_i^s\}$ represents the parameters of the robotic system both morphological and those of the internal model. The \mathcal{F}^* function is, so to speak, the estimation that the robot system makes about the real model of the environment. Now, we change the point of view to focus on the environment (red and dashed line in Fig. 1). The input flux to the environment will be the output flux of the robot system internal model \mathbf{Y}_i . We can consider these functional transforms: $\mathbf{E}_y = \mathcal{F}(\Gamma_i^a, \mathbf{Y}_i)$; $\mathbf{E}_x = \mathcal{E}(\mathbf{E}_y) \equiv \mathcal{E}^*(\Gamma_i^a, \mathbf{Y}_i)$; $\mathbf{X}_i = \mathcal{I}(\Gamma_i^s, \mathbf{E}_x) \equiv \mathcal{I}^*(\Gamma_i^s, \Gamma_i^a, \mathbf{Y}_i)$

$$\begin{aligned} \mathbf{Y}_i &\xrightarrow{\mathcal{F}} \mathbf{E}_y \xrightarrow{\mathcal{E}} \mathbf{E}_x \xrightarrow{\mathcal{I}} \mathbf{X}_i \\ \mathbf{X}_i &= \mathcal{I}^*(\Gamma_i^a, \Gamma_i^s, \mathbf{Y}_i) \end{aligned} \quad (2)$$

Therefore, if the internal model is a rough approximation of the real model ($\mathbf{Y}_i = \mathcal{M}(\Gamma_i^p, \mathbf{X}_i)$ and $\mathbf{X}_i = \mathcal{I}^*(\Gamma_i^a, \Gamma_i^s, \mathbf{Y}_i)$), there must exist a correlation between the morphological parameters and the internal model parameters.

$$\Gamma_i^p = \mathcal{S}(\Gamma_i^m) \quad (3)$$

The (Eq. 3) will be true if the morphological parameters accomplish:

$$\left. \frac{\partial \Gamma_i^m}{\partial X_i} \right|_i = \left. \frac{\partial \Gamma_i^m}{\partial Y_i} \right|_i = 0 \quad (4)$$

In the model learning frameworks, the optimal model structure should be obtained from training data [3], and the model can be fitted into this function:

$$\mathbf{Y}_i = f(\mathbf{X}_i) + \epsilon_i; f(\mathbf{X}_i) = \phi(\mathbf{X}_i)^T \theta_i \quad (5)$$

where ϵ_i is an error term that we assume follows a Gaussian distribution as $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$. The function $\phi(\mathbf{X}_i)$ maps the vector \mathbf{X}_i into an N dimensional feature space. In these conditions, the probability density distribution of \mathbf{Y}_i given the model parameters θ_i is: $\rho_i(\mathbf{Y}_i \mid \mathbf{X}_i, \theta_i) = \mathcal{N}(\phi(\mathbf{X}_i)^T \theta_i, \sigma_i^2 I)$. This is valid for any robotic system (i). We can approximate the knowledge about the model of the environment using θ_i for a particular morphology.

2.2 *Extracting Knowledge from Multiple Robotic Systems*

We assume that there are several robotic systems interacting with the same environment to execute the same task and to show similar behaviors, but they have different morphologies.¹ If the same learner is used to estimate the features of these systems, a set of pairs features/parameters will be available.

Each pair $\{X_i, Y_i\}$ is related with Γ_i^m as it was defined in (Eq. 2). Therefore we can model this relationship using a regression model as:

$$\theta_i = g(\Gamma_i^m) + \xi; g(\Gamma_i^m) = \Phi(\Gamma_i^m)^T \mathbf{W} \quad (6)$$

In this case, $\xi \sim \mathcal{N}(0, \Sigma^2)$ and it is an additive error. The probability density distribution is Gaussian as: $\rho(\theta_i | \Gamma_i^m, \mathbf{W}) = \mathcal{N}(\Phi(\Gamma_i^m)^T \mathbf{W}, \Sigma^2 I)$. (Eqs. 5 and 6) are parametric models, so it is necessary to estimate these parameters using some machine learning regression tool. When a new robot system is presented with its particular morphology Γ_i^m , we can use the information acquired from other systems represented by \mathbf{W} in (Eq. 6) to estimate its internal model $\hat{\theta}_i$. On the other hand, fitting the model of the regression (Eq. 6) is a machine learning challenge due to the size of θ_i . Depending on the number of the system inputs and outputs the size of θ_i can be thousands of values. Therefore, there exists an undue difference between the size of the morphological and internal model parameters.

3 Case Study

We are interested in how we can learn the model expressed by (Eq. 6). Here we propose several machine learning methods. However, we need a dataset to check their ability to solve this problem. We generated a dataset based on many robotic systems which have to show a particular behavior immersed in a particular environment.

3.1 *The Morphology of the Robot System*

We consider a simulated robot head with two cameras and three control DOF. We are using this kind of system because it presents morphological parameters that correspond to both actuators and sensors. This system has eleven DOF in total, but eight are just prismatic joints which allow us to change its morphology. Table 1, describes the Denavit-Hartenberg model for one side of the head, the other side would be the same sharing the first joint. Depending on the prismatic joint values, the morphology of the head will be changed. These joints do not modify their value when the

¹The same number and kind of morphological parameters but different values.

Table 1 Denavit-Hartenberg model of the left side of the head. ρ_p, ρ_t are the revolute joints of the pan and common tilt motors. The right side is the same model and it shares the ρ_t joint

Joint	ρ (rad)	r(m)	a(m)	α (rad)	Offset	Type
q_1	ρ_t	0	0	$\pi/2$	$\pi/2$	R
q_2	0	0	0	$-\pi/2$	0	P
q_3	$\pi/2$	0.055	0	$\pi/2$	0	P
q_4	$\pi/2$	0.055	0	$\pi/2$	0	P
q_5	ρ_p	0	0	$\pi/2$	π	R
q_6	0	0.01	0	$\pi/2$	0	P

robot head is executing a task, so they are not control joints. We consider that the movement of these joints are precise enough to avoid adding noise to the system. The parameters which describe the head actuators in relation to robot morphology, are the values of these eight prismatic joints ($\Gamma_i^a \in \mathbb{R}^8$). In addition, the system is completed with two camera sensors. The pinhole model is used to simulate both cameras. Each camera has several morphological parameters: focal length (f), pixel size (s) (supposed squared), height (h) and width (w) images resolution. Therefore, the morphological parameters of the sensors are eight ($\Gamma_i^s \in \mathbb{R}^8$). There are 16 morphological parameters for each robotic head configuration ($\Gamma_i^m \in \mathbb{R}^{16}$).

3.2 The Behavior of the Robot System

A saccade is a fast eye movement that shifts the gaze at a target point and it can be used to scan the visual space [7]. From a robotics point of view, generating a saccade requires to solve an inverse control problem, in which the retinotopic position of the stimulus has to be converted into a shift of the eye position. We focus on the transformation that links the visual position of the stimulus into a target position of the eyes (open loop with respect to vision). Learning this transformation requires to learn the inverse kinematic model of the robot head. We will use a *feedback error learning* [8] (FEL) model which consist of two inverse controllers. A fixed feedback controller (\mathbf{B}) slowly drives the system toward the target and provides a learning signal to a second adaptive controller (\mathbf{C}_f) (Fig. 2). This last one provides an inverse model of the robot head which is used to rectify the output of \mathbf{B} . The input of the controller is the visual target (\mathbf{t}) and the current eye position (\mathbf{e}), while the output is the saccade command ($\Delta\mathbf{e}$). This value is sent to the robot head actuators, moving the head accordingly. After the movement, the new visual position of the stimulus \mathbf{t}' is converted into a motor error to adapt the inverse controller \mathbf{C}_f . Using this approach, the adaptive filter learns to compensate the poor response of the fixed feedback control. We need an *a priori* estimation of \mathbf{B} which converts the visual target into a movement. We consider a static stimulus in the center of the image and generate motor babbling. In

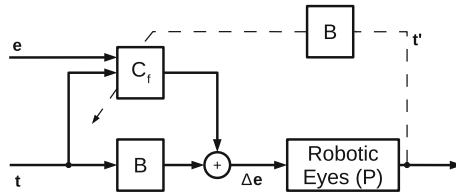


Fig. 2 FEL. The visual position of the stimulus (t') and the current eye position e are converted into a motor command Δe by summing up the contributions of a fixed element B and adaptive element C_f . Changing the C_f state requires a motor error $P^{-1}(t') \approx Bt'$

this way, we can correlate the t with Δe and we directly obtain B from least squares regression. A general approach to learn the proposed C_f is to use a neural network. The main characteristics of a neural network can be explained in two parts, i.e. its architecture and functional properties [5]. The architecture is related to the network topology and the functional properties are concerned with how learning takes place to decompose complex information into its fundamental elements, i.e. weights (θ_i). When C_f is learning, the mean visual error of a saccade movement decreases until the system reaches a point in which it learns so slowly that its weights barely change.

The performance of the system is evaluated by the visual error after a saccade movement. The visual stimulus should be centered in the image after a saccade. In the proposed model we considered an approximation $P^{-1}(t') \approx Bt'$. So though we work without noise inputs, the model has an error. We are not interested in this specific algorithm to improve the saccade behavior. We simply use it, in order to implement a behavior in the robotic system. Furthermore, we intend to test the ability of the proposed neural network architectures to learn the model described by (Eq. 6). For this reason we are not going to use any noise inputs from the environment to the robot system in the experimental tests. From the internal robot model point of view, we need two kinds of parameters to represent the behavior of the system, $\Gamma_i^p = \{\theta_i, B_i\}$ corresponding to the two controllers.

3.3 The Environment

The environment is a virtual object that is randomly placed in the vision field of the two robot cameras of the robot. This object is static while the robotic system is moving in a saccadic way. In this first approach, we are not going to consider the error in the stimulus perception because our aim is to search for a correlation between the robot morphology and its internal model and to find out to what extent this relationship can be learned.

Table 2 Morphological parameters to generate the robotic system dataset

Prismatic joints (cm)	
Left side	Right side
$\Gamma_{i,1}^p = q_2 \in [-0.054, 0.054]$	$\Gamma_{i,2}^p = q_2 = \Gamma_{i,1}^p + [0, 0.01]$
$\Gamma_{i,3}^p = q_3 \in [0, 0.07]$	$\Gamma_{i,4}^p = q_3 = \Gamma_{i,3}^p + [0.035, 0.07]$
$\Gamma_{i,5}^p = q_4 \in [-0.02, 0.054]$	$\Gamma_{i,6}^p = q_4 = \Gamma_{i,5}^p + [0, 0.02]$
$\Gamma_{i,7}^p = q_6 \in [0, 0.01]$	$\Gamma_{i,8}^p = q_6 = \Gamma_{i,7}^p + [0, 0.01]$
Cameras parameters	$f(\text{px}); s(\text{m/px}); w(\text{px}); h(\text{px})$
Left camera	Right camera
$\Gamma_{i,9}^p = f_l \in [340, 1920]$	$\Gamma_{i,10}^p = f_r = \Gamma_{i,9}^p + [0, 200]$
$\Gamma_{i,11}^p = s_l \in [3 \cdot 10^{-6}, 7 \cdot 10^{-6}]$	$\Gamma_{i,12}^p = s_r \in [3 \cdot 10^{-6}, 7 \cdot 10^{-6}]$
$\Gamma_{i,13}^p = h_l \in [340, 1920]$	$\Gamma_{i,14}^p = h_r = \Gamma_{i,13}^p + [0, 200]$
$\Gamma_{i,15}^p = w_l \in [340, 1920]$	$\Gamma_{i,16}^p = w_r = \Gamma_{i,15}^p + [0, 200]$
if $\Gamma_{i,13}^p > \Gamma_{i,15}^p$, swap($\Gamma_{i,13}^p, \Gamma_{i,15}^p$)	if $\Gamma_{i,14}^p > \Gamma_{i,16}^p$, swap($\Gamma_{i,14}^p, \Gamma_{i,16}^p$)

4 The Dataset Creation

In this section, we explain how we created a dataset $\{\Gamma_i^m, \Gamma_i^p\}, \forall i \in [1, m]$ using the case study proposed before. We randomly simulated different setups of the robot head ($m = 44271$) with the morphological parameters described in the previous section. We fixed a reasonable interval for each $\Gamma_{i,j}^p, \forall i \in [1, m], \forall j \in [1, 16]$ to avoid unfeasible configurations. We randomly chose the values for the left side of the head and then selected the right side values in a random interval as shown in Table 2. A generated head sample is showed in Fig. 3. For each head configuration, we have to learn their internal models from the interaction with the environment. We are following the FEL model as described above. Thus, we must learn the fixed controller (**B**) parameters and the adaptive controller (**C_f**).

Learning the fixed controller.

For each robotic head configuration, we generated a set of b pairs $\{\mathbf{t}_j, \Delta \mathbf{e}_j\} \forall j \in [1, b]$ by motor babbling. In our case, $\mathbf{t}_j \in \mathbb{R}^4$ (stimulus image position in both cameras) and $\Delta \mathbf{e}_j \in \mathbb{R}^3$ (motor control actions), therefore **B_i** has to be a matrix with size 3×4 :

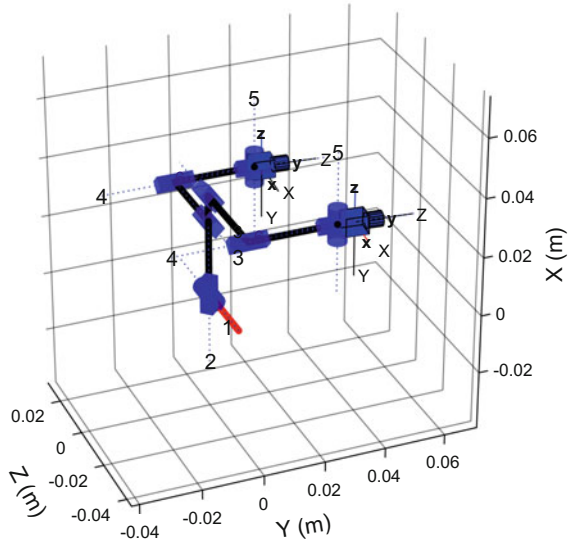
$$\mathbf{B}_i = (\Delta \mathbf{e}^T \Delta \mathbf{e})^{-1} \Delta \mathbf{e}^T \mathbf{t} \quad (7)$$

We tested beforehand the stability of the **B_i** values for different values of b with several robotics systems and concluded to use $b = 600$.

Learning the adaptive controller.

We trained on-line every robotic head setup to learn their adaptive controllers. We used a single layer neural network for all trainings with 7 inputs $\{\mathbf{t}, \mathbf{e}\}$ and 3 outputs ($\Delta \mathbf{e}$). Gaussian activations using random sparse features [9] were used for the

Fig. 3 Example of the head model



hidden layer. As we consider that these features are integrated in the network architecture, we generated a unique set of random sparse features (Ω_m). They are used for all network training and their subsequent performance validations. The algorithm for adapting the neural networks weights was the incremental sparse spectrum Gaussian process regression (I-SSGPR) [10]. Besides handling high dimensionality input, the I-SSGPR updates the weights using an incremental algorithm that computes the *maximum a posteriori* estimation. We have previously tuned the network parameters [10]: variance of the model ($\sigma_n^2 = 0.1$), signal variance ($\sigma_f^2 = 1.0$) and number of the projections ($D = 300$), therefore $\Omega_m \in \mathbb{R}^{600}$. The initial weights are random and different for each head setup. For training, a virtual object is placed in the environment. The cameras acquire an image of the object and its image position. From the fixed controller value (previously estimated) \mathbf{B}_i and the current output of the neural network (\mathbf{C}_f), the incremental control action for the head motors is generated. The head is moved with this control commands and the cameras acquire the new virtual object position in the images. The neural network is updated using the visual error in this position [7]. After a number of iterations, the value of the visual error is stabilized and the neural network weights barely change. In our case, to ensure that this point is reached, considering the time consumption of the training process simulation, each robotic head setup is trained for 1000 iterations.

The resulting weights of the neural network at this point are considered a part of the internal model parameters $\theta_i \in \mathbb{R}^{600 \times 3}$ since they partially determine the behavior of the system. For each head configuration, this training process is repeated three times and θ_i is taken as the mean of the each resulting weights.

5 The Proposed Neural Networks

After learning the robot behavior, we have a huge dataset composed of 44271 vectors and matrices: $\Gamma^m \in \mathbb{R}^{16}$, $\mathbf{B} \in \mathbb{R}^{3 \times 4}$ and $\theta \in \mathbb{R}^{600 \times 3}$. We center on the regression between Γ^m and θ because there exists a great difference between the sizes of Γ_i^m and θ_i . We propose three different kinds of neural networks to solve the regression problem. By using the generated dataset, we are going to test their ability to learn the relationship between the morphology and the internal model of the robotic systems. We changed the scale of some morphological parameters to avoid errors in the learning algorithm, e.g. the height and width for the camera resolution were converted to decimeters using the pixel size. In this way, all inputs had the same magnitude.

5.1 The Single Layer Feedforward Neural Network

We have to solve a regression problem for which the most widespread tool is a feed-forward neural network. We propose a single layer neural network with 16 input units, 1000 units in the hidden layer and 1800 units in the output layer.² The hidden layer has as the hyperbolic tangent as nonlinear function and the output layer is linear. The algorithm used to train the network is *scaled conjugate gradient back-propagation* (SCG) [11].

5.2 The Deep Neural Network

Autoencoders (AE) are simple learning circuits which aim to transform inputs into outputs with the least possible amount of distortion [12]. In other words, an AE is trying to learn an approximation to the identity function subjected to several constraints such as limiting the number of the hidden units. The AE neural network is an unsupervised learning algorithm. An AE is composed of two parts, an encoder and a decoder. The output of the encoder is a representation of the input, whereas the output of the decoder is the input reconstruction from the encoder representation [13]. If the number of the hidden units is less than the number of the input units the AE algorithm is forced to learn a short representation of the input. This kind of autoencoder is called *contractive autoencoder* (CAE). Moreover, when the hidden layer has more units than the input layer and a sparsity constraint is imposed, the AE will still discover interesting structure in the data [14]. This kind of autoencoder is called *sparse autoencoder* SAE.

We propose a deep neural network architecture combining two stacks of AEs, one of CAEs and the other of SAEs. Both stacks extract the maps of relevant features from the Γ^m and θ . Then the network executes two regressions to correlate both

²The parameters of the network were calculated previously using cross validation.

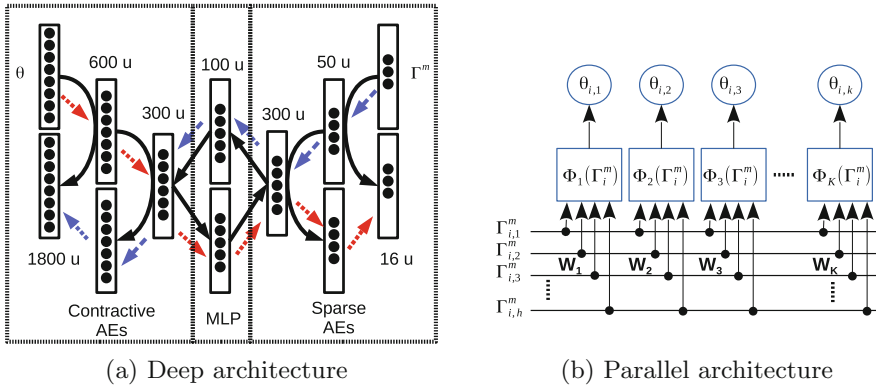


Fig. 4 **a** The proposed deep neural network architecture. It is composed of two stacks of AEs connected by two feedforward neural networks. Therefore, when the network is trained, it is possible to get the morphological parameters for a θ , following the *red arrows*. Furthermore, θ can be obtained from a Γ^m following the *blue arrows*. **b** Neural network scheme to solve the regression problem. We use a neural network for each internal model parameter

maps. A drawing of the used architecture is shown in Fig. 4a. In the first CAE, there are 1800 input units (θ) and 600 units in the hidden layer. The activation outputs of this hidden layer are used as input for the second CAE when the first CAE has been trained. The second CAE has 300 units in the hidden layer. Moreover, the input of the first SAE has 16 units (Γ) and its hidden layer has 50 units. The activation outputs of these units are used as input for the second SAE, when the first SAE has been trained. The second SAE has 300 units in the hidden layer. The activation outputs of this hidden layer are correlated with the activation outputs of the second CAE using a feedforward neural network. The activation function of the hidden layer for all AEs is the logistic function and the output function is linear. The parameters used for learning are summarized in Table 3. All these parameters have been estimated previously in each AE using cross validation. The AEs have been trained using a SCG. The two feedforward neural networks used to correlate both AE stacks have 100 units in the hidden layer and they have been trained using a SCG algorithm. The activation function of this two neural networks is the hyperbolic tangent.

Table 3 AE parameters

AE	Hidden units	L2 reg.	Sparsity reg.	Sparsity prop.
CAE	600	10^{-6}	10^{-6}	10^{-2}
CAE	300	10^{-2}	10^{-4}	10^{-2}
SAE	300	10^{-5}	10^{-5}	10^{-1}
SAE	50	10^{-5}	10^{-5}	10^{-1}

5.3 The Parallel Feedforward Neural Network

Another alternative is to consider each θ_{ij} as an individual value and to attempt to correlate it with Γ^m . We can use a feedforward neural network for each pair $\{\theta_{ij}, \Gamma^m\}$ to solve (Eq. 6). This network architecture is shown in Fig. 4b. We used 1800 feedforward neural networks with 16 inputs and 1 output. The activation function of the hidden layer units is the hyperbolic tangent.

6 Experimental Results and Discussion

We made a partition of 26500 robotic head configurations for training the three classes of neural networks previously described. Once they were trained, we evaluated their performance using random partitions of the robotic head configurations that were not used in the training process. Each one was composed of 11000 head configurations. We repeated the training procedure four times and the results can be viewed in Table 4. The magnitude used to evaluate the performance is the mean squared error (MSE). The shown values are the mean of the four training attempts. We can conclude from the results in Table 4 that the best method to solve the initial regression problem is the parallel neural network. This is due to the nature of θ . In the proposed case study, θ contains the weights of a neural network trained online. Frequently, (Eq. 6) is solved in an iterative way. If we consider the adaptation equation for a weight in a neural network training algorithm, it is calculated as:

$$\theta_{i,j}(t+1) = \theta_{i,j}(t) + \frac{\partial \mathcal{L}(\theta_i, \mathbf{X}_i, \mathbf{Y}_i)}{\partial \theta_{i,j}} \quad (8)$$

where $\mathcal{L}(\theta_i, \mathbf{X}_i, \mathbf{Y}_i)$ is a loss function. The value of the derivative term is changing during the training; however, when the neural network is trained properly, this term has to tend to zero. At this point, the value of $\theta_{i,j}$ is independent of the rest of the weights. When we attempt to learn the relationship between morphology and internal model by using a single layer neural network or the proposed deep architecture, we force them to learn a relationship between weights that does not actually exist. On the other hand, the parallel neural network assumes this hypothesis from its conception.

Table 4 Mean values of the MSE for the four attempts for each proposed neural network

Network	MSE
Single neural network	$(1.7340 \pm 0.0178)10^{-3}$
Deep neural architecture	$(0.5029 \pm 0.0197)10^{-3}$
Parallel neural network	$(0.2194 \pm 0.0429)10^{-3}$

7 Conclusion and Future Work

Learning the relationship between the morphological parameters of a robot system and those in its internal model is a high dimensional regression problem (Eq. 6). We have proposed three neural network architectures for solving this problem, along with a case study to evaluate their performance. With the robot system suggested in the case study, we have generated a huge dataset that we used to test the performance of the proposed neural network architectures. Due to the fact that the trained weights are independent of one another, the best proposal is the parallel neural network, that exploits this characteristic. Our results suggest that this neural network can be a proper tool to learn the relationship between the morphology and the internal model of a robot system. It could then be possible to take advantage of the knowledge learned in one system to improve the learning process in a system with different morphology. We will aim our next experiments at verifying this hypothesis.

Acknowledgements This paper describes research done at the UJI Robotic Intelligence Laboratory. Support for this laboratory is provided in part by Ministerio de Economía y Competitividad (DPI2015-69041-R), by Generalitat Valenciana (PROMETEOII/2014/028) and by Universitat Jaume I (P1-1B2014-52, PREDOC/ 2013/06).

References

1. Sigaud, O., Salan, C., Padois, V.: On-line regression algorithms for learning mechanical models of robots: a survey. *Robot. Auton. Syst.* **59**(12), 1115–1129 (2011)
2. Siciliano, B., et al.: *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer, London (2008)
3. Nguyen-Tuong, D., Peters, J.: Model learning for robot control: a survey. *Cogn. Process.* **12**(4), 319–340 (2011)
4. Hornik, K.: Approximation capabilities of multilayer feedforward networks. *Neural Networks* **4**(2), 251–257 (1991)
5. Gupta, M.M., Homma, N., Jin, L., Homma, N.: *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory* (2003)
6. Kikuchi, K., Kobayashi, H.: A study on functional characteristics of robotic system with morphology and intelligence. In: *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 733–738 (2000)
7. Antonelli, M., Duran, A.J., Chinellato, E., del Pobil, A.P.: Learning the visual-oculomotor transformation: effects on saccade control and space representation. *Robot. Auton. Syst.* **71**, 13–22 (2015)
8. Kawato, M.: Feedback-error-learning neural network for supervised motor learning. In: Eckmiller, R. (ed.) *Advanced Neural Computers*, pp. 365–372. North-Holland, Amsterdam (1990)
9. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: *Advances in neural information processing systems*, pp. 1177–1184 (2007)
10. Gijbbers, A., Metta, G.: Real-time model learning using Incremental Sparse Spectrum Gaussian Process Regression. *Neural Networks* **41**, 59–69 (2013)
11. Möller, M.F.: A scaled conjugate gradient algorithm for fast supervised learning supervised learning. *Neural Networks* **6**, 525–533 (1993)
12. Baldi, P.: Autoencoders, Unsupervised Learning, and Deep Architectures. *ICML Unsupervised and Transfer Learning*, pp. 37–50 (2012)

13. Rifai, S., Muller, X.: Contractive auto-encoders : explicit invariance during feature extraction. *icml* **85**(1), 833–840 (2011)
14. Ng, A.: Sparse autoencoder. *CS294A Lecture Notes* **72**, 1–19 (2011)

Part XIV
Robot Navigation

Combining Feature-Based and Direct Methods for Semi-dense Real-Time Stereo Visual Odometry

Nicola Krombach, David Droschel and Sven Behnke

Abstract Visual motion estimation is challenging, due to high data rates, fast camera motions, featureless or repetitive environments, uneven lighting, and many other issues. In this work, we propose a two-layer approach for visual odometry with stereo cameras, which runs in real-time and combines feature-based matching with semi-dense direct image alignment. Our method initializes semi-dense depth estimation, which is computationally expensive, from motion that is tracked by a fast but robust feature point-based method. By that, we are not only able to efficiently estimate the pose of the camera with a high frame rate, but also to reconstruct the 3D structure of the environment at image gradients, which is useful, e.g., for mapping and obstacle avoidance. Experiments on datasets captured by a micro aerial vehicle (MAV) show that our approach is faster than state-of-the-art methods without losing accuracy. Moreover, our combined approach achieves promising results on the KITTI dataset, which is very challenging for direct methods, because of the low frame rate in conjunction with fast motion.

Keywords Visual SLAM · Visual Odometry (VO) · 3D-reconstruction · Semi-dense · MAVs

1 Introduction

For the autonomous navigation of mobile robots, a robust and fast state estimation is of great importance. Many mobile robots contain cameras since they are inexpensive and lightweight and can be used for a variety of tasks, including visual obstacle detection, 3D scene reconstruction, visual odometry, and even visual simultaneous localization and mapping (SLAM).

Visual odometry (VO) describes estimating the egomotion solely from images, captured by a monocular or stereo camera system. A variety of VO methods exists

N. Krombach (✉) · D. Droschel · S. Behnke
Autonomous Intelligent Systems Group, University of Bonn, Bonn, Germany
e-mail: krombach@ais.uni-bonn.de

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_62

855

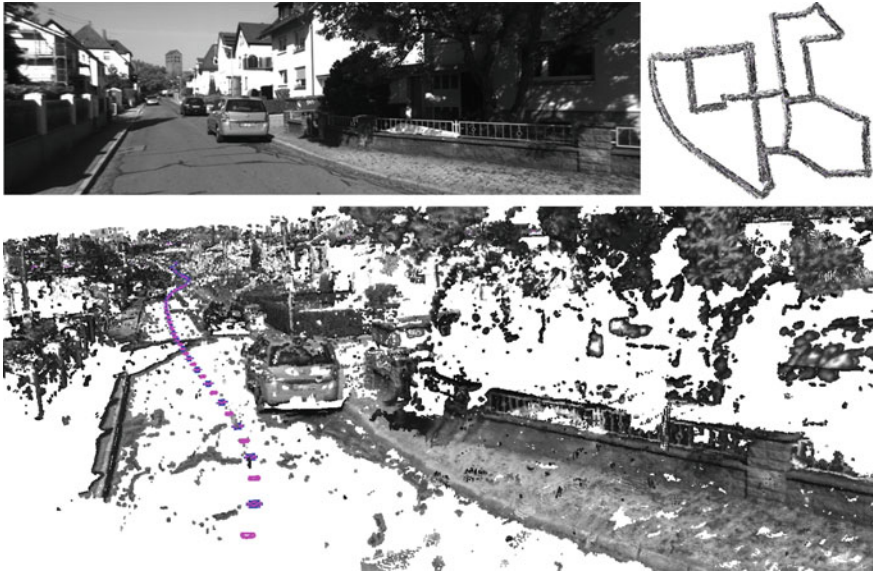


Fig. 1 Semi-dense 3D reconstruction of KITTI 00: *Top left* Camera image. *Bottom* Semi-dense 3D reconstruction with colored camera trajectory (key frames *blue*, feature-based tracked frames *pink*). *Top right* Bird's eye view of the complete reconstructed scene

that can be classified into feature-based and direct methods. Most VO methods are feature-based and work by detecting feature points and matching them between subsequent frames. In contrast, direct VO methods estimate the camera motion by minimizing the photometric error over all pixels. As the minimization over all pixel is computationally more demanding than determining the reprojection error of sparse feature points, direct methods are often slower than feature-based methods. In this work, we propose a novel approach that combines direct image alignment with sparse feature matching for stereo cameras (Fig. 1). By combining both approaches, we are able to process images with high frame rate and to fill gaps caused by large motions. Due to the distinctiveness of the tracked features, our method performs well on datasets with low frame rates, which is often a problem for direct methods since they need sufficient image overlap. Our work is based on LSD-SLAM [6], which is a fully direct method for monocular SLAM that is real-time capable on strong CPUs due to its semi-dense approach.

We extend LSD-SLAM to stereo and restrict semi-dense tracking to key frames to achieve a higher frame rate. To estimate the motion between key frames, we employ a feature-based VO method and use the estimated motion as prior for the direct image alignment. Thus, we restrict the search space for direct image alignment and gain real-time performance even on CPUs for mobile applications.

2 Related Work

VO methods estimate egomotion using only images of a single or of multiple cameras. They can be classified into feature-based, direct and hybrid methods.

2.1 *Feature-Based Methods*

The general pipeline for feature-based VO methods can be summarized as follows: Features are detected and either matched or tracked over time. Based on these feature correspondences, the relative motion between two frames is computed. To compensate for drift, many methods make use of pose-graph optimization. Popular feature-based methods are MonoSLAM [4] and Parallel Tracking and Mapping (PTAM) [12]. PTAM is a widely used feature-based monocular SLAM method, which allows robust state estimation in real-time and has been successfully used on MAVs with monocular cameras [23]. Recently, ORB-SLAM [14] has been proposed as a monocular visual SLAM method that tracks ORB features in real-time. When using monocular methods, additional sensors are needed to estimate the absolute scale of a scene. In contrast, stereo or multi-camera VO methods [10, 17, 18] do not suffer from scale ambiguity. Recent feature-based methods also incorporate readings from an inertial measurement unit (IMU) as high-frequency short-term estimates between frames [1, 13].

In our work, we rely on an efficient feature-based library for stereo visual odometry [10], which provides a good trade-off between accuracy and runtime.

2.2 *Direct Methods*

In contrast to feature-based methods, which abstract images into a sparse set of feature points, direct methods use the entire image information in order to minimize the photometric error. Therefore, these methods are computationally very intense and thus much slower than feature-based methods. Direct approaches exist for stereo, RGB-D and monocular cameras [3, 6, 7, 20, 21]. They often need to use GPUs to achieve real-time performance [16, 19]. By using only pixels with sufficient gradient, LSD-SLAM [6] reduces the computational demand and real-time monocular semi-dense SLAM becomes possible with a strong CPU. This approach has been extended to stereo cameras recently [7].

2.3 *Hybrid Methods*

For the 3D environment reconstruction, direct methods have the advantage of estimating a dense map, while feature-based methods can only rely on sparse features

that have been tracked. Dense direct methods are computationally demanding and are often executed as a final step for estimating a globally consistent dense map after pose tracking with sparse interest-points succeeded. To speed up global optimization, already tracked sparse feature-points can be used as initialization for dense mapping [15]. A recent semi-direct method uses direct motion estimation for initial feature extraction and then continues by using only these features [8].

In contrast to this, we continually combine feature-based and semi-dense direct tracking over time, taking advantage of the fast tracking from the feature-based method and the accurate alignment of image gradients from direct methods. The feature-based tracking result is immediately fed to the direct tracking at runtime as initial guess.

3 Method

Our method is mainly based on the monocular version of LSD-SLAM that we extended to work with stereo cameras. By using stereo cameras instead of a monocular camera, the absolute scale of the scene becomes observable, eliminating scale ambiguity and the need for additional sensors. To ensure a high frame rate, we restrict the semi-dense direct alignment to key frames only and estimate the motion for all other frames by the feature-based method LIBVIS02 [10]. This motion estimate is used as initial estimate for direct alignment of key frames. The semi-dense environment mapping runs in a parallel thread.

3.1 Notation

We follow the notation of Engel et al. [6]. The monochrome stereo images captured at time i are denoted with $I_i^{l/r} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$, with image domain Ω . Each key frame $KF_i = \{I_i^l, I_i^r, D_i, V_i\}$ consists of the left and right stereo images $I_i^{l/r}$, the semi-dense inverse depth map $D_i : \Omega_{D_i} \rightarrow \mathbb{R}^+$, and the corresponding variance map $V_i : \Omega_{D_i} \rightarrow \mathbb{R}^+$. The inverse of the depth z of a pixel is denoted as $d = z^{-1}$. Camera motions are represented as twist coordinates $\xi \in \mathfrak{se}(3)$ with corresponding transformation matrix $T_\xi \in SE(3)$. A 3D point $\mathbf{p} = (p_x, p_y, p_z)^T$ is projected into image coordinates $\mathbf{u} = (u_x, u_y, 1)^T$ by the projection function $\pi(\mathbf{p}) := \mathbf{K} (p_x/p_z, p_y/p_z, 1)^T$ with intrinsic camera matrix \mathbf{K} . Thus, the inverse projection function $\pi^{-1}(\mathbf{u}, d)$ maps a pixel with corresponding inverse depth to a 3D point $\mathbf{p} = \pi^{-1}(\mathbf{u}, d) := ((d^{-1}\mathbf{K}^{-1}\mathbf{u})^T, 1)^T$.

3.2 LSD-SLAM

The processing pipeline of LSD-SLAM [6] consists of the three main components: Tracking, depth map estimation, and global map optimization. Tracking is based on maximizing photo-consistency and thus minimizing the photometric error between the current frame and the most recent key frame using Gauss-Newton optimization:

$$E(\xi) := I_{KF}(\pi(\mathbf{p})) - I(\pi(\mathbf{T}_\xi \mathbf{p})) , \quad (1)$$

where \mathbf{p} is warped from I_{KF} to I by ξ . New frames are tracked towards a key frame and the rigid body motion of the camera $\xi \in \mathfrak{se}(3)$ is estimated. In the depth map estimation, tracked frames are then used to refine the existing depth map of the key frame by many small-baseline stereo comparisons. With each new tracked frame, the depth map of the key frame is refined by either creating new depth hypotheses or improving existing ones. New key frames are created when the distance exceeds a certain threshold and are initialized by propagating depth of the previous key frame towards the new frame. Once a key frame is replaced, it is added to the pose-graph for further refinement and loop closing.

3.3 LIBVISO2

LIBVISO2 [10] is a fast feature-based VO library for monocular and stereo cameras. Similar to other feature-based methods, it consists of feature matching over subsequent frames and egomotion estimation by minimizing the reprojection error. Features are extracted by filtering the images with a corner and blob mask and performing non-maximum and non-minimum suppression on the filtered images. Starting from all feature detections in the current left image, candidates are matched in a circular fashion over the previous left image, the previous right image, the current right image, and back to the current left image. If the first and last feature of such a circle match differ, the match is rejected. Based on all found matches, the egomotion is then estimated by minimizing the reprojection error using Gauss-Newton and outliers are removed using RANSAC.

3.4 Semi-dense Alignment of Stereo Key Frames

We build upon the open source release of monocular LSD-SLAM and extend it with stereo functionality. In contrast to monocular visual odometry, stereo allows to compute absolute depth maps and, thus, does not suffer from scale drift. By extending LSD-SLAM to stereo, we combine the existing depth map computation over time with instant stereo depth from the current image pair. While monocular LSD-SLAM

uses a random initialization and has to bootstrap over the first frames, we take advantage of using stereo cameras and initialize our method with absolute depth values. We use ELAS [9] to compute the depth map of the initial key frame. The following key frames are registered with their previous key frame by minimizing the photometric error as well as the depth error. While in the monocular case, absolute depth is not observable, with stereo cameras absolute depth is observable for every incoming stereo image pair. This allows us to minimize the depth error in addition to the photometric error. Hence, for direct tracking with stereo, we extend the minimization of the photometric residual r_p to take the depth residual r_d into account:

$$\begin{aligned} r_p(\mathbf{p}, \xi) &= \left\| I_{KF_i}(\pi(\mathbf{p})) - I_j(\pi(\mathbf{T}_\xi \mathbf{p})) \right\|, \\ r_d(\mathbf{p}, \xi) &= \left\| D_{KF_i}(\pi(\mathbf{p})) - D_{stereo_j}(\pi(\mathbf{T}_\xi \mathbf{p})) \right\|, \end{aligned} \quad (2)$$

where ξ is the camera motion from the i -th key frame to the new j -th frame and D_{stereo_j} is the initial instant stereo depth map of the j -th frame. The minimization is performed using a weighted least squares formulation and solved with the Gauss-Newton method. The residual is formulated as stacked residual \mathbf{r} and is weighted with a 2×2 weight matrix \mathbf{W} :

$$\mathbf{r}(\xi) = \sum_{p \in \Omega_{D_i}} \begin{pmatrix} r_p(\mathbf{p}, \xi) \\ r_d(\mathbf{p}, \xi) \end{pmatrix}; \quad \mathbf{W}(\xi) = \sum_{p \in \Omega_{D_i}} \begin{pmatrix} w(r_p(\mathbf{p}, \xi)) & 0 \\ 0 & w(r_d(\mathbf{p}, \xi)) \end{pmatrix}, \quad (3)$$

where both residuals are weighted with the Huber norm denoted as $w(\cdot)$.

3.5 Hybrid Odometry Estimation

Our idea is to take advantage of the different strengths of both approaches and, thereby, combine fast feature matching with precise semi-dense image alignment for efficient and reliable state estimation. The modular structure of our approach is illustrated in Fig. 2.

We initialize the first key frame with a dense depth map computed by ELAS. Subsequent frames are then tracked towards the key frame incrementally using feature-based LIBVISO2. The relative poses of the tracked frames are concatenated and form the relative pose of the camera to the key frame:

$$\xi_{feat} = \xi_{in} \circ \xi_{in-1} \circ \dots \circ \xi_{i0}. \quad (4)$$

The current absolute pose of the camera at step j and key frame i can be retrieved by:

$$\xi_{ij} = \xi_{KF_i} \circ \xi_{ij-1}. \quad (5)$$

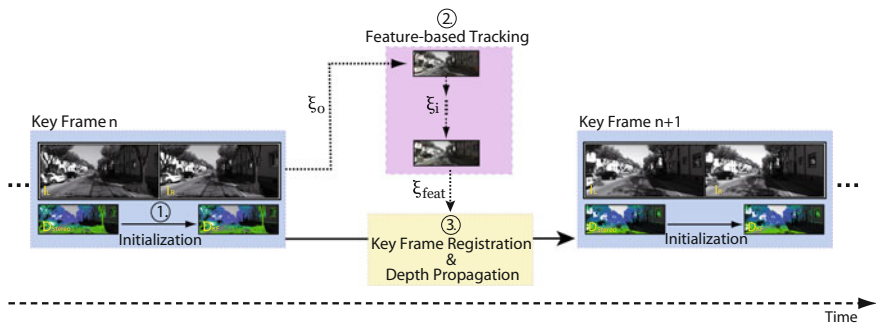


Fig. 2 Overview of our hybrid approach. While direct tracking is only performed on key frames, feature-based tracking is performed for frames in between. The output of the feature-based odometry serves as prior for the direct tracking

We perform feature-based odometry as long as the motion is sufficiently small. As soon as the motion exceeds the motion threshold ϵ_{motion} , we perform direct registration again and the previous key frame is replaced with the new frame:

$$\epsilon_{motion} = \frac{1}{n} \sum_{k=1}^n \sqrt{(u_i^k - u_{i-1}^k)^2}, \tag{6}$$

where n is the number of matched feature points and (u_i^k) and (u_{i-1}^k) are corresponding feature matches between the current and the previous image. The motion ξ_{feat} serves as initial estimate for the direct registration of the new frame towards the key frame:

$$\xi_{KF_{i+1}} = \xi_{KF_i} \circ \xi_{feat} . \tag{7}$$

This allows us to track larger motions faster and more robustly. The depth map of a new key frame is initialized by instant stereo correspondences and then fused with the previous depth map by propagation as described in the next section. Once a new key frame is initialized, we start feature-based matching again.

3.6 Map Update

The depth map of each key frame is updated with instant stereo measurements as well as with propagated depth from the previous key frame. If a new key frame is created, the depth map is computed by instant stereo from the left and right images. For this, we use a simple but fast block matching along epipolar lines. Corresponding pixels are found by minimizing the sum of absolute distances (SAD) error over a 15×15 pixel window. The variance ω for each depth hypothesis is determined as described by Engel et al. [5].



Fig. 3 Computed semi-dense depth maps for KITTI datasets (sequences 00 and 01). Color depicts estimated distance to the sensor

After initializing the depth map with stereo measurements, the depth estimates are refined by propagating depth hypotheses of the old depth map to the new frame:

$$\mathbf{p}_{new}(\mathbf{p}) = \mathbf{R}_{C,KF} \mathbf{p} + \mathbf{t}_{C,KF}, \quad (8)$$

where \mathbf{p} is the 3D point in the old key frame. The rotation $\mathbf{R}_{C,KF}$ and translation $\mathbf{t}_{C,KF}$ describe the coordinate transformation from the key frame coordinate system KF to the candidate coordinate system C . If the residual between the instant and propagated depth is high, the depth value with smaller variance is chosen. Otherwise both estimates— d_{stereo} and d_{prop} —are fused to a new depth estimate d_{new} as a variance-weighted sum:

$$d_{new} = (1 - \omega) d_{stereo} + \omega d_{prop} . \quad (9)$$

Figure 3 shows the resulting semi-dense depth maps for two KITTI sequences.

4 Evaluation

For the evaluation of our hybrid approach, we perform experiments on two challenging stereo datasets: The well-known KITTI-dataset [11] and the EuRoC dataset [2]. The datasets differ in terms of frame rate, apparent motion, and stereo baseline. All experiments have been conducted on an Intel Core i7-4702MQ running at 2.2 GHz with 8 GB RAM. The processing is performed on the original image resolution of the rectified images of 1241×376 and 752×480 , respectively. We compare the quality of our combined approach in terms of accuracy and runtime to LSD-SLAM [6] and LIBVISO2 [10], as well as to two more state-of-the-art methods: S-PTAM [18] and ORB-SLAM [14]. The results of the referred methods have been obtained using the provided default parameters. As ground truth for all sequences is available, we employ the evaluation metrics by Sturm et al. [22] and measure the absolute trajectory error (ATE) by computing the root mean squared error (RMSE) over the whole trajectory. Additionally, for the monocular systems, the scaling factor is estimated to obtain the absolute scale of the camera trajectory. For an intuitively accessible visualization, trajectories are always shown in bird’s eye perspective.

Table 1 ATE Results on KITTI Dataset

KITTI Sequence	Absolute trajectory error RMSE (Median) in m			
	Ours	LIBVISO2	ORB-SLAM	S-PTAM
00	6.15 (5.02)	29.71 (18.49)	8.30 (6.04)	7.83 (6.30)
01	61.74 (55.48)	66.54 (60.46)	335.52 (303.79)	204.65 (157.10)
02	19.47 (15.80)	34.26 (27.36)	18.66 (15.03)	20.78 (17.28)
03	0.67 (0.58)	1.67 (1.54)	11.91 (9.19)	10.53 (10.41)
04	0.72 (0.49)	0.80 (0.66)	2.15 (1.73)	0.98 (0.88)
05	5.78 (4.69)	22.14 (19.07)	4.93 (4.73)	2.80 (2.24)
06	4.37 (3.53)	11.54 (10.26)	16.01 (15.56)	4.00 (4.01)
07	2.63 (1.77)	4.41 (4.37)	4.30 (3.65)	1.80 (1.53)
08	8.75 (7.26)	47.67 (34.84)	38.80 (18.12)	5.13 (4.26)
09	5.55 (4.07)	89.83 (77.57)	7.46 (6.91)	7.27 (4.61)
10	1.87 (1.68)	49.35 (36.00)	8.35 (7.55)	2.08 (1.70)
Mean	10.70 (9.12)	32.54 (26.42)	41.49 (35.66)	25.74 (20.26)
Mean w/o S 01	5.60 (4.49)	29.14 (23.02)	12.09 (8.85)	7.85 (6.57)

4.1 Accuracy

In terms of accuracy, we achieve similar results as current state-of-the-art stereo methods. As our method is a pure odometry method, it accumulates drift over time, especially at large rotations, where direct alignment of key frames becomes more demanding.

The KITTI benchmark is very challenging for direct methods, as it contains fast motions up to 80 km/h in combination with a low frame rate, which causes inter-frame motions up to 2.8 m/frame.

The results of our evaluation on the KITTI dataset are shown in Table 1. Unfortunately, LSD-SLAM fails on all sequences of the KITTI dataset. This is probably caused by too large inter-frame motion for a pure monocular direct method, as sufficient scene overlap is important for successful tracking. Moreover, it can be seen, that all methods lack performance on Sequence 01, resulting in a very high ATE. Sequence 01 contains images from driving on a highway, thus it is hard to find re-occurring feature points in subsequent frames. When averaging over the eleven training sequences, our method ranks first, followed by S-PTAM, ORB-SLAM, and LIBVISO2. However, the bad results from Sequence 01 greatly affect the final average computation, which is why we also show mean values omitting this sequence.

Unfortunately, to our knowledge there is no other publicly available direct method other than LSD-SLAM to compare with. However, as LSD-SLAM fails on the KITTI sequences, we compare our hybrid approach to its fully direct version without feature-based initial estimates. In particular, we compare our hybrid approach to its building blocks—LIBVISO2 and direct stereo tracking—separately. Figure 4 shows

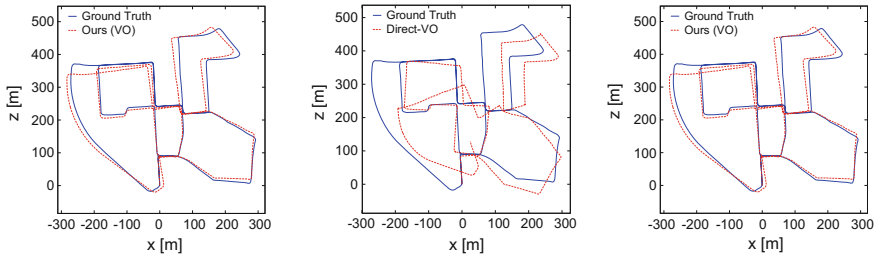


Fig. 4 Comparison of LIBVISO2 (*left*), direct (*middle*) and the proposed hybrid odometry (*right*) on KITTI Sequence 00. Our method accumulates less drift

the estimated trajectories of the three methods exemplary on the KITTI 00 sequence. It can be seen that our hybrid approach accumulates less drift than the pure feature-based or direct method. The direct tracking performs worst on this dataset, because it fails at tracking large inter-frame motions and strong rotations without a good initial estimate.

In addition to the evaluation on the KITTI dataset, we perform further experiments on the recently released EuRoC MAV dataset [2] that contains WVGA stereo images, captured with 20 Hz on an Asctec Firefly hex-rotor. We choose six trajectories with different difficulties from the two Vicon datasets V0 and V1. The data has been collected from flights in a room that is equipped with a Vicon motion capture system, providing 6D ground truth poses. Each dataset contains three trajectories with increasing difficulty: Easy (_01), medium (_02), and difficult (_03). The easy trajectories have good illumination, are feature rich, and show no motion blur, only low optical flow, and low varying scene depth. They capture a static scene. The difficulty increases in the medium trajectories by adding challenging lighting conditions, high optical flow, and medium varying scene depth. However, they still show a static scene and a feature rich environment without motion blur. In contrast, the difficult scene contains areas with only few visual features and more repetitive structures. Moreover, they add motion blur and more challenging lighting conditions. The MAV performs very aggressive flight maneuvers resulting in high optical flow and highly varying scene depth in a non-static scene. The resulting ATE values for this datasets are listed in Table 2. As the difficult datasets V1_03 and V2_03 contain very dynamic movements and fast rotations with an MAV, LSD-SLAM often loses track after a few seconds and is then unable to re-localize for the rest of the trajectory. This is denoted as failure (X). Similarly, S-PTAM and ORB-SLAM lose track for the difficult trajectory V2_03. This dataset shows very challenging conditions with strong motion blur and fast aggressive maneuvers. Moreover, the absence of sufficient visual features makes it hard for the feature-based methods to succeed.

Table 2 shows that our approach outperforms the other methods in terms of accuracy and robustness, and reliably recovers the motion for all test sequences. Additionally, it can be seen, that the results of LIBVISO2 are improved on every trajectory. On average, our hybrid odometry achieves a higher accuracy, with 0.49 m ATE,

Table 2 ATE Results on EuRoC Dataset

EuRoC Dataset	Absolute trajectory error RMSE (Median) in m				
	Ours	LIBVISO2	LSD-SLAM	ORB-SLAM	S-PTAM
V1_01	0.25 (0.18)	0.31 (0.31)	0.19 (0.10)	0.79 (0.62)	0.28 (0.19)
V1_02	0.24 (0.16)	0.29 (0.27)	0.98 (0.92)	0.98 (0.87)	0.50 (0.35)
V1_03	0.81 (0.76)	0.87 (0.64)	X	2.12 (1.38)	1.36 (1.09)
V2_01	0.22 (0.13)	0.40 (0.31)	0.45 (0.41)	0.50 (0.42)	2.38 (1.78)
V2_02	0.31 (0.25)	1.29 (1.08)	0.51 (0.48)	1.76 (1.39)	4.58 (4.18)
V2_03	1.13 (0.97)	1.99 (1.66)	X	X	X
Mean	0.49 (0.41)	0.85 (0.71)	0.53 (0.48)	1.23 (0.94)	1.82 (1.52)

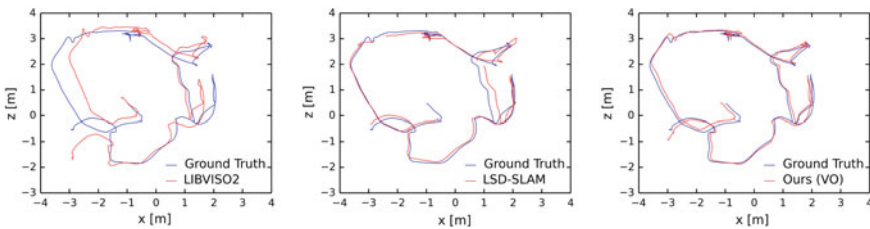


Fig. 5 Comparison of LIBVISO2 (*left*), Mono LSD-SLAM (*middle*) and our hybrid odometry (*right*) on dataset V2_01 with ground truth from a Vicon motion capture system. Our method is much closer to the ground truth

than LSD-SLAM (0.53 m), ORB-SLAM (1.23 m), and S-PTAM (1.82 m). LSD-SLAM, ORB-SLAM and S-PTAM often suffer from fast motions in combination with rotations, and temporarily lose track. In addition, we compare results of our hybrid odometry to fully feature-based and direct methods, shown exemplified on dataset V2_01 in Fig. 5. While the results from LIBVISO2 and LSD-SLAM show ATEs of approximately 40 cm, our method yields an error of only 22 cm.

In summary: We achieve similar or better accuracy on different challenging datasets as current state-of-the-art stereo methods. Moreover, our combined approach performs better than both—feature-based and direct—odometries on their own.

4.2 Runtime

We evaluated the average runtime of our method over all datasets on full resolution. As the tracking of new frames and the map building run in distinct threads, timings are given for each part separately in Table 3. On the KITTI dataset, our average runtime for tracking new frames lies below 30 ms. Hence, our approach is significantly faster than recent direct stereo methods [7] and reaches real-time performance. With

Table 3 Average runtimes of all evaluated methods

Dataset	Method	Tracking (ms)	Mapping (ms)	Total (VO) (ms)
KITTI	Ours	26.5	36.6	63.1
	LSD-SLAM	–	–	–
	ORB-SLAM	30.7	254.0	284.6
	S-PTAM	71.1	5.7	77.4
	LIBVIS02	33.8	–	33.8
EuRoC	Ours	22.6	39.6	62.2
	LSD-SLAM	27.6	85.6	113.2
	ORB-SLAM	17.9	159.2	177.1
	S-PTAM	47.3	1.5	48.8
	LIBVIS02	24.8	–	24.8

higher frame rate—as in the EuRoC dataset—we achieve even better results because we need to do direct tracking less often.

4.3 Qualitative Analysis

A major advantage of using a direct approach for tracking is an accurate semi-dense 3D point cloud, which contains every pixel with sufficient gradient. Thus, our odometry not only estimates the current pose of the camera, but also builds a 3D map of the environment, which can be used for additional tasks, like obstacle avoidance. Figure 6 shows an example of recovered scene depth at near distance and Fig. 1 displays the reconstructed scene of a longer odometry segment from the KITTI dataset 00.



Fig. 6 *Left* Camera image. *Right* Reconstructed semi-dense 3D point cloud

5 Conclusions

In this paper, we proposed a novel hybrid visual odometry method that combines feature-based tracking with semi-dense direct image alignment. Our method fuses depth estimates from motion between key frames with instantaneous stereo depth estimates. The performance of our method has been evaluated in terms of accuracy and runtime on two challenging datasets. Our experiments show that for tracking egomotion between image frames, we achieve accuracies similar to the state-of-the-art at high frame rate without the necessity to reduce the image resolution. Due to the feature-based tracking as prior for semi-dense direct alignment, our method is computationally less expensive than direct methods, but still takes advantage of all image points with sufficient gradient for precise keyframe registration. The distinctiveness of the tracked features makes our method also more robust against large inter-frame motion than direct methods.

In future work, we plan to incorporate high frequency IMU readings and to evaluate other feature-based tracking priors, e.g. ORB features. Moreover, since our method accumulates drift over time, we plan to extend our method by a SLAM backend to enhance accuracy and robustness.

Acknowledgements This work has been supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) in the Autonomics for Industry 4.0 project InventAIRy.

References

1. Achtelik, M., Achtelik, M., Weiss, S., Siegwart, R.: Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments. In: *The International Conference on Robotics and Automation (ICRA)* (2011)
2. Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., Siegwart, R.: The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* (2016)
3. Comport, A., Malis, E., Rives, P.: Accurate quadrifocal tracking for robust 3D visual odometry. In: *The International Conference on Robotics and Automation (ICRA)* (2007)
4. Davison, A., Reid, I., Molton, N., Stasse, O.: Monoslam: Real-time single camera SLAM. *Pattern Anal. Mach. Intell.* **29**(6), 1052–1067 (2007)
5. Engel, J., Sturm, J., Cremers, D.: Semi-dense visual odometry for a monocular camera. In: *The International Conference on Computer Vision (ICCV)* (2013)
6. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-scale direct monocular SLAM. In: *European Conference on Computer Vision (ECCV)* (2014)
7. Engel, J., Stückler, J., Cremers, D.: Large-scale direct SLAM with stereo cameras. In: *International Conference on Intelligent Robots and Systems (IROS)* (2015)
8. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: Fast semi-direct monocular visual odometry. In: *The International Conference on Robotics and Automation (ICRA)* (2014)
9. Geiger, A., Roser, M., Urtasun, R.: Efficient large-scale stereo matching. In: *Asian Conference on Computer Vision (ACCV)* (2010)
10. Geiger, A., Ziegler, J., Stiller, C.: Stereoscan: Dense 3D reconstruction in real-time. In: *Intelligent Vehicles Symposium (IV)* (2011)

11. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: *The Conference on Computer Vision and Pattern Recognition(CVPR)* (2012)
12. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: *The International Symposium on Mixed and Augmented Reality (ISMAR)* (2007)
13. Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P.: Keyframe-based visual-inertial odometry using nonlinear optimization. *Int. J. Robot. Res.* (2014)
14. Mur-Artal, R., Montiel, J., Tardós, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. *Trans. Robot.* **31**(5), 1147–1163 (2015)
15. Mur-Artal, R., Tardós, J.D.: Probabilistic semi-dense mapping from highly accurate feature-based monocular SLAM. In: *Robotics: Science and Systems* (2015)
16. Newcombe, R.A., Davison, A.: Live dense reconstruction with a single moving camera. In: *The Conference on Computer Vision and Pattern Recognition (CVPR)* (2010)
17. Nieuwenhuisen, M., Droschel, D., Schneider, J., Holz, D., Läbe, T., Behnke, S.: Multimodal obstacle detection and collision avoidance for micro aerial vehicles. In: *European Conference on Mobile Robots (ECMR)* (2013)
18. Pire, T., Fischer, T., Civera, J., Cristóforis, P.D., Berles, J.J.: Stereo Parallel Tracking and Mapping for robot localization. In: *International Conference on Intelligent Robots and Systems (IROS)* (2015)
19. Pizzoli, M., Forster, C., Scaramuzza, D.: REMODE: Probabilistic, monocular dense reconstruction in real time. In: *The International Conference on Robotics and Automation (ICRA)* (2014)
20. Stückler, J., Behnke, S.: Multi-resolution surfel maps for efficient dense 3d modeling and tracking. *J. Vis. Commun. Image Represent.* **25**(1), 137–147 (2014)
21. Stückler, J., Gutt, A., Behnke, S.: Combining the strengths of sparse interest point and dense image registration for rgb-d odometry. In: *International Symposium on Robotics (ISR) and 8th German Conference on Robotics (ROBOTIK)* (2014)
22. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: *International Conference on Intelligent Robots and Systems (IROS)* (2012)
23. Weiss, S., Scaramuzza, D., Siegwart, R.: Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments. *J. Field Robot.* **28**(6), 854–874 (2011)

Outdoor Robot Navigation Based on View-Based Global Localization and Local Navigation

Yohei Inoue, Jun Miura and Shuji Oishi

Abstract This paper describes a view-based outdoor navigation method. Navigation in outdoor can be divided into two levels; the global level deals with localization and subgoal selection, while the local level deals with safe navigation in a local area. We adopt an improved version of SeqSLAM method for global-level localization, which can cope with changes of robot's heading and speed as well as view changes using very wide-angle images and a Markov localization scheme. The global level provides the direction to move and the local level repeatedly sets subgoals with local mapping using 3D range sensors. We implemented these global and local level methods on a mobile robot and conducted on-line navigation experiments.

Keywords Outdoor navigation · View-based localization · Mobile robot

1 Introduction

Mobile service robot is an emerging application area in robotics. Such a robot is expected to provide various service tasks like attending, guiding, and searching. One of the indispensable functions of mobile service robots is *navigation*, which makes it possible for the robot to move from one place to another autonomously. Since outdoor environments are important part of human activity, mobile service robots should be able to navigate in outdoor.

Outdoor navigation can be divided into two levels. The global level deals with localization and subgoal selection, while the local level deals with safe navigation in

Y. Inoue · J. Miura (✉) · S. Oishi
Department of Computer Science and Engineering, Toyohashi University of Technology,
Toyohashi, Japan

e-mail: jun@cs.tut.ac.jp; jun.miura@tut.jp

Y. Inoue

e-mail: inoue@aisl.cs.tut.ac.jp

S. Oishi

e-mail: oishi@cs.tut.ac.jp

a local area. This is an analogy to a navigated car driving: a car navigation system tells a driver where the car is and which way to take, and the driver is responsible for safely driving, including following traffic rules and avoiding possible collisions.

Several approaches are possible for outdoor localization. GPS-based systems are usually used, especially in the case of automobiles but could be unreliable or not operational near tall buildings in, for example, usually campus environments. A precise digital map of the environment is also required. Usual mapping and localization approaches might be adopted but making large-scale outdoor maps is often costly. We therefore adopt a simpler way, that is, view-based localization [1–4].

One of the issues in view-based localization is how to cope with view changes. Some of earlier works deal with them using a learning with training data in various illumination conditions [5] or a two-stage SVM-based object/location learning [6, 7]. Yamagi et al. [8] developed a view-based navigation system which uses a robust template matching method. Milford and Wyeth [9] proposed the SeqSLAM method which realizes a very robust image sequence matching even under an extreme view changes. Although this method shows a good performance for image sequences taken from a vehicle, it is not always directly applicable to mobile robot navigation. We therefore use this method with several improvements for realizing a mobile robot navigation in outdoor, combined with a local mapping and path planning capabilities.

The rest of the paper is organized as follows. Section 2 describes an improved SeqSLAM method with several off-line experimental validation. Section 3 describes a local navigation strategy including local mapping, subgoal selection, and path planning. Section 4 shows the navigation experiments. Section 5 concludes the paper and discusses future work.

2 View-Based Localization by SeqSLAM and Its Improvements

2.1 SeqSLAM

SeqSLAM [9] is a view-based localization method which compares a model image sequence with an input one for robust matching. To cope with a large illumination change between a training and a test time, they apply *local contrast enhancement* as follows.

Let \mathbf{D} be a vector of the differences between an input image and the images in the model sequence, which is considered to cover possible range of model images for the input image. Each element D_i in \mathbf{D} is normalized by:

$$\hat{D}_i = (D_i - \bar{D}_l) / \sigma_l, \quad (1)$$

where D_l and σ_l are the mean and the standard deviation of \mathbf{D} . By this enhancement, even if an input image is largely different from the model images and all of the difference values are very large due to a large illumination change, the difference for the true correspondence is expected to be sufficiently small compared to the others. These enhanced vectors are compiled for $d_s + 1$ frames into a matrix \mathbf{M} which has the model and the input image sequence in the row and the column, respectively:

$$\mathbf{M} = [\hat{\mathbf{D}}^{T-d_s}, \hat{\mathbf{D}}^{T-d_s+1}, \dots, \hat{\mathbf{D}}^T] \quad (2)$$

An example matrix is shown in Fig. 3.

Then, assuming a constant velocity during the sequence, a line is searched for which minimizes the following total difference S :

$$S = \sum_{t=T-d_s}^T D_k^t, \quad (3)$$

$$k = s + V(d_s - T + t), \quad (4)$$

where V is the gradient of the line (or a relative velocity in input and model acquisition) and k is the index of the corresponding image in the model sequence for the input image at time t .

SeqSLAM exhibited great performances against drastic view changes, at least for road sequence images. There are, however, rooms for improvements when applied to mobile robot navigation. The following subsections explain our improvements.

2.2 Improvements in Image Matching

The original SeqSLAM uses intensity values normalized within a small window for the feature for image matching. This is simple and fast, but is not very strong for a region with little textures. It is also weak to a large view direction changes. We therefore adopt two improvements: HOG feature matching and the use of a wide angle camera.

HOG feature matching: HOG feature [10] is a histogram of edges in a local region and suitable for representing shape information. The size of training images is 630×420 pixels with 90° FOV (field of view). The cell size for calculating HOG is 35×35 pixels and the number of blocks is 17×11 . Figure 1 shows an example result of HOG calculation. We use a normal SAD (sum of absolute differences) for calculating the dissimilarity between images.

Coping with a variety of robot motion direction: Mobile robots changes their moving directions frequently not only for moving towards a destination but also avoiding collisions with people and obstacles. Since each image in a view sequence captures a scene in a specific direction, it is very much likely to have a different

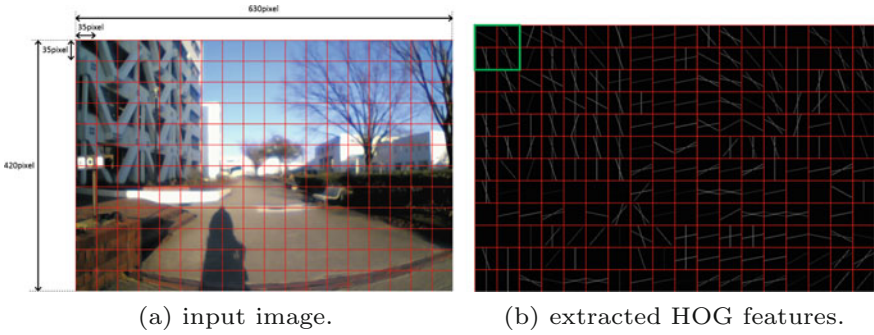


Fig. 1 HOG extraction result

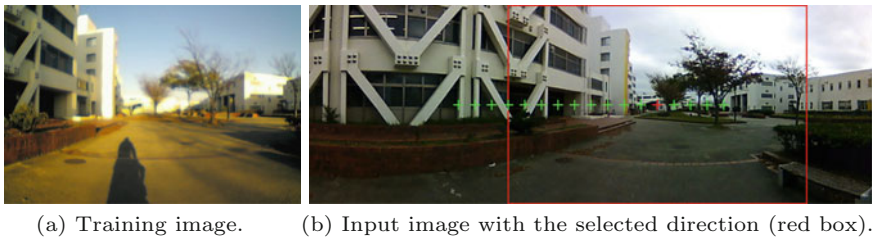


Fig. 2 Selection of moving direction

orientation during navigation, thereby degrading the view-based localization performance.

Morita et al. [11] used an omnidirectional camera to cope with this problem. We also take a similar approach using a wide images (1190×420 pixels with about 180° Horizontal FOV), with which 33×11 blocks are obtained. We scan a learned image horizontally on the wide image within $\pm 40^\circ$ range with 5° interval, and chooses the minimum distance position, which is then used for determining the subgoal direction (i.e., the direction for the robot to move). Figure 2 shows an example of selecting a direction.

2.3 Improvements in Image Sequence Matching

The original SeqSLAM assumes a constant speed during acquisition of training and input image sequences; the matrix is searched for the best line which minimizes the total difference. This assumption is sometimes violated in the case of mobile robots because they need to adjust their speed adaptively to the surrounding situation for, for example, avoid collision and/or threatening to people. We therefore use a DP to cope with such speed variations during image acquisition. We also effectively utilizes the history of movement to increase the reliability and reduces the calculation cost.

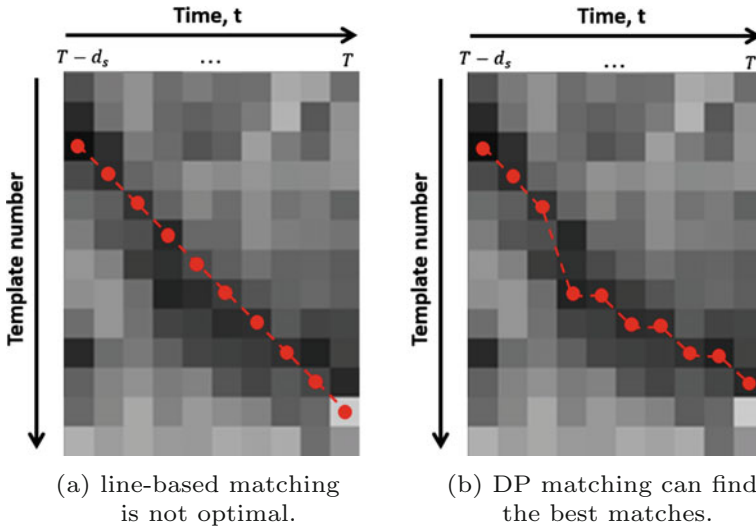


Fig. 3 DP matching between training and navigational image sequences. In the matrix, *darker* elements have smaller differences

DP Matching: DP (dynamic programming) matching [12] is a tool for calculating a match between two data sequences with non-constant interval between data. Figure 3 shows an example of DP matching for image sequences with non-constant robot motions; a linear matching is not suitable for this case. We set a limitation on a speed difference between the training and the navigation phase and apply the DP matching for obtaining the best matched image pairs with an evaluation. In addition, unlike SeqSLAM, we use the latest frame as a representative image of a sequence so that the current location is estimated on-line.

Markov localization: Mobile robot localization often uses a movement history, which is effective to limit the possible robot positions in prediction. Miura and Yamamoto [7] adopted a Markov localization strategy in a view-based localization. In [7], a discrete set of locations are provided for localization and a probabilistic model of transitions between locations was used in the prediction step. This can reduce not only localization failures but also the calculation cost with a limited number of sequence matches.

The Markov localization here is formulated as follows:

$$\hat{Bel}(l) \leftarrow \sum_{l'} P_m(l|l') Bel(l'), \tag{5}$$

$$Bel(l) \leftarrow \alpha P_o(s|l) \hat{Bel}(l), \tag{6}$$

$$P_o(s|l) = \frac{S_{min}}{S_l}, \tag{7}$$

where $P_m(l|l')$ denotes the transition probability from frame l' to l , $Bel(l)$ the belief of the robot being location l , $P_o(s|l)$ the likelihood of location l with sensing s , which is calculated by the minimum matching score of the DP matching divided by the score for location l .

The state transition model $P_m(l|l')$ is determined by considering the image acquisition interval and the robot motion patterns. Currently, the training images are taken with about 1 m interval and the robot takes one image per two seconds with moving at 1 m/s. Since the robot speed changes frequently due to many reasons such as collision avoidance and turning motions, we use a transition model in which the robot may move to locations corresponding to one of the current and the three subsequent location with equal probabilities.

2.4 Off-Line Localization Experiments

Figure 4 shows the route used for the experiments. This is in our campus and about 300 m long. We manually moves the robot on this route and acquired one training robot on this route and acquired one training image set and two testing image sets. The training set and the first testing image set (test1) were taken while the robot moves along the route, while the second testing image set (test2) was taken as the robot did zig-zag motions so that the direction of the robot changes largely from position to position. The image sets are summarized in Table 1.

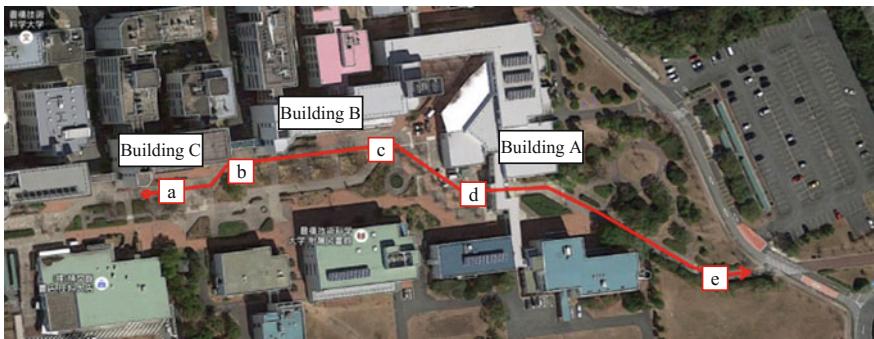


Fig. 4 The route for experiments

Table 1 Training and testing image sets

Camera		Date and weather	# of images	Robot motion
Training	Normal	March 5, 2015, fine	259	Smooth
Test1	Wide	July 11, 2015, cloudy	263	Smooth
Test2	Wide	July 11, 2015, cloudy	282	Zig-zag

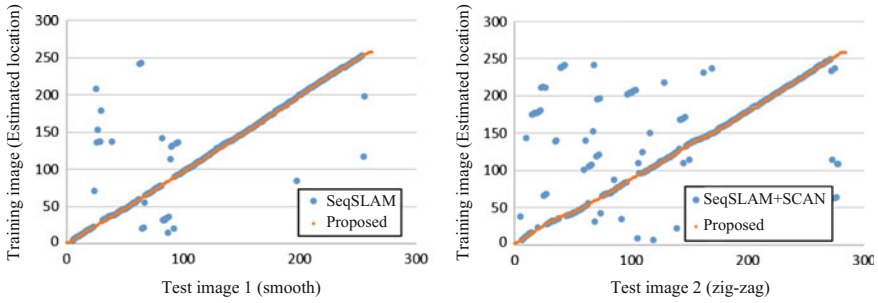


Fig. 5 Localization results

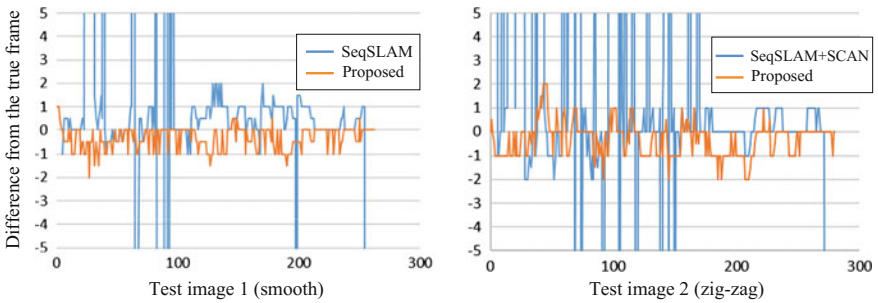


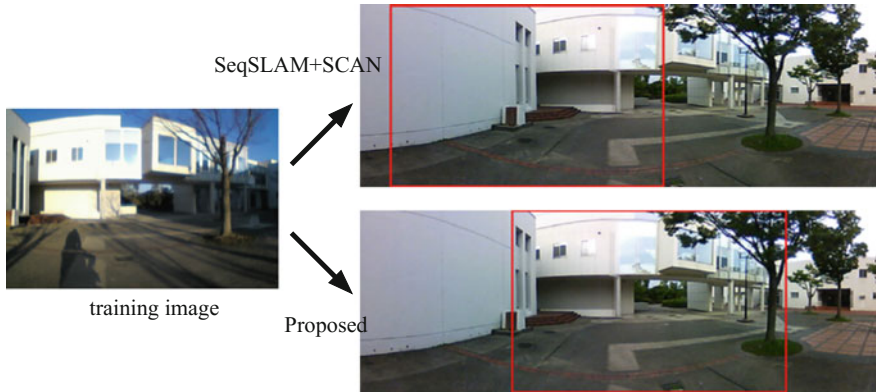
Fig. 6 Localization accuracy

Figure 5 shows the results of localization experiment. We compared the proposed method and SeqSLAM for test1 and test2 image sequences. Since the original SeqSLAM exhibits quite a low performance for the test2 image sequence due to a large variation of the robot heading, we additionally performed a horizontal scanning to find the best matched position in the wide image. Both comparison results show that the proposed method exhibits a much better performance.

Figure 6 shows the evaluation of localization accuracy. The ground truth is determined by manually comparing the training and test images. When an input image is judged to be located between two consecutive training images, the true position is set in the middle of the training images. The maximum frame difference by the proposed method is two for most of frames, meaning the maximum localization error is about 2 m because the training images are acquired with about 1 m interval. Table 2 summarizes the performance in terms of *localization success rate* and *direction selection success rate*. Localization is considered success when the difference is within two frames, while the direction is considered correctly selected when the directional difference is less than 5°. Figure 7 shows a scene where the proposed and the SeqSLAM with scanning suggest different moving directions. Since SeqSLAM does a direct comparison of (normalized) pixel values, it is sometimes weak to scenes with less textures as shown in the figure.

Table 2 Quantitative evaluation results

Test image	Method	Localization success rate (%)	Direction selection success rate (%)
Test1	SeqSLAM	86.1	–
	Proposed	99.6	99.6
Test2	SeqSLAM	74.2	63.1
	Proposed	95.8	95.8

**Fig. 7** Comparison in selecting the moving direction

3 Local Path Planning Using View-Based Localization Results

The proposed view-based localization method provides the direction to move. That information by itself is, however, not enough for guiding an actual robot safely. We therefore develop a local navigation system which includes local mapping, subgoal selection, and path planning.

3.1 Local Mapping

Figure 8 shows our mobile robot. It is based on an electric wheelchair (Patrafour by Toyota Motor East Japan Inc.), equipped with two 3D laser range finders (LRFs) (FX-8 by Nippon Signal Co.) for local mapping and finding free spaces, and a wide-angle camera for view-based localization. Each LRF has about 60° horizontal FOV and two LRFs covers about 100° FOV.

We use two 3D laser range finder for local mapping, which is for finding free spaces. We detect obstacles in two ways. One is to use a height map which detects

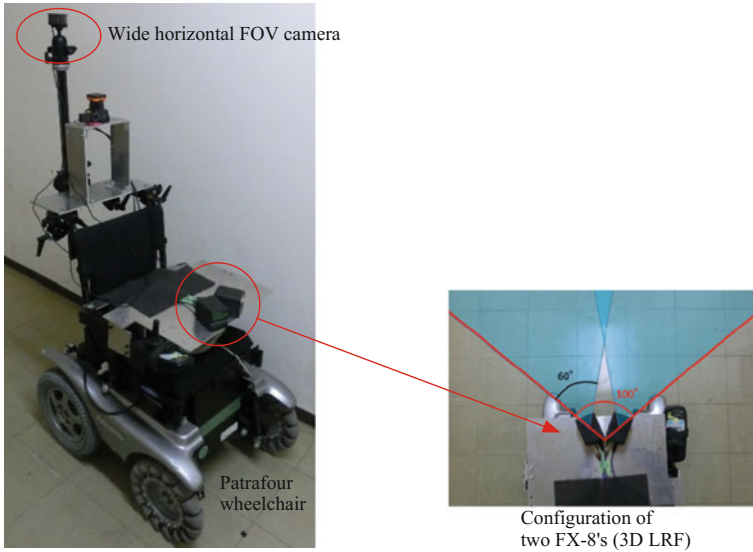


Fig. 8 Our robot with two LRFs and a camera



Fig. 9 Detection of low steps. A local obstacle map centered at the current robot position is shown on the *right*. *Red ellipses on the left* show the detected step locations

regions with relatively high obstacles. We use a polar coordinate with 1° and 50 cm intervals in the angle and the distance axis, respectively, for representing height maps. The pose of the range sensors relative to the ground plane is estimated by fitting a plane to the data points in the region in front of the robot. We set a threshold to 15 cm to detect this kind of obstacles.

The other way is to find low steps. Since it is sometimes difficult to such steps only from the height due to a limited ranging accuracy and the error in ground plane estimation, we examine the differentiation of height data. A region with a large height difference with a certain number of data points is considered to be an obstacle region (i.e., a low step). Figure 9 shows an example of low step detection in a real scene. The steps near the robot are successfully detected.

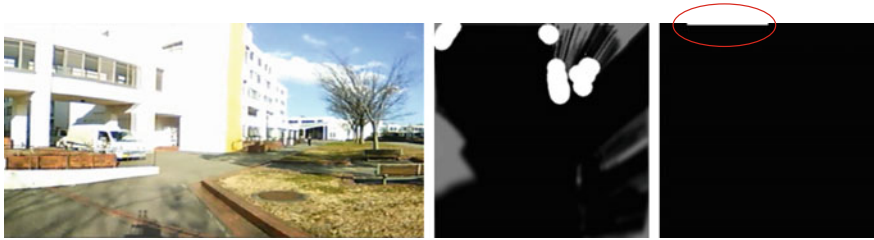


Fig. 10 Subgoal selection example. *Left* scene, *center* local map with enlarged obstacles, *right* frontier whose center point is selected as the subgoal

3.2 Subgoal Selection and Path Planning

The direction to move is suggested by the view-based localization, however, it is not always possible to move in that direction due to obstacles. Since the free spaces are recognized only in the local map, we need to set a subgoal in the local map which is safe and leads the robot to the destination. We here adopt the concept of *frontier* which is often used in exploration planning in an unknown space [13]. A frontier point is a point which is free and adjacent to an unknown point. Such a point either inside the local map or on the edge of the map. All frontier points are partitioned into clustered, among which the ones with line-shape and having a certain size are selected and their center points become the candidates for the subgoal. The most appropriate center point is then selected as the subgoal which has the minimum orientational difference with the suggested moving direction. Figure 10 shows an example selection of subgoal (frontier).

Once the subgoal is set, the path towards it is generated. We use our RRT-based on-line path planner [14]. Since the cycle of view-based localization is about two seconds while the path planner runs in a faster cycle, we use the same subgoal until the next subgoal is set based on the next result from the view-based global localization.

4 Navigation Experiment

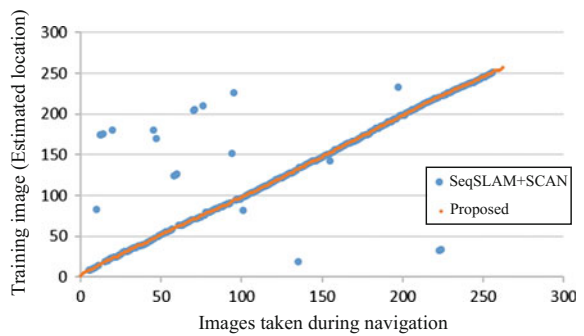
We conducted experiments in the route shown in Fig. 4. The training image set used is the one used in the off-line experiments (see Sect. 2.4). Figure 11 shows snapshots of a run conducted on February 15, 2016. The names of the locations correspond to symbols in Fig. 4. The robot successfully completed the autonomous navigation.

Figure 12 shows an off-line comparison between the proposed method and SeqSLAM with scanning for the image sequence taken during the actual navigation. The figure shows that the proposed global localization method works on-line with



Fig. 11 Navigation experiment. *Left* Robot in motion. *Center* Estimated location (i.e., best matched training image). *Right* Input image and selected moving direction

Fig. 12 Comparison using the image sequence taken during an actual navigation



a high reliability. During other experiments, however, the robot sometimes stuck or almost hit obstacles mainly due to obstacle detection failures. It is necessary to improve the local navigation system for more reliable global navigation.

5 Conclusions and Future Work

This paper described an outdoor navigation system combining a view-based global localization method and a local navigation system. We developed an improved version of SeqSLAM and has shown its reliability in outdoor navigation. We also realized a robot system that can navigate itself given an image sequence of the route to follow.

The system has been validated in a route in our campus. It is necessary to evaluate the system in a more variety of scenes, that is, more variations in weather, season, surrounding objects (buildings or forests), and so on. For a more reliable navigation, it is also necessary to improve the performance of local mapping and navigation is required.

Acknowledgements This work is in part supported by JSPS KAKENHI Grant Number 25280093.

References

1. Matsumoto, Y., Inaba, M., Inoue, H.: Visual navigation using view-sequenced route representation. In: Proceedings of 1996 IEEE International Conference on Robotics and Automation, pp. 83–88 (1996)
2. Katsura, H., Miura, J., Hild, M., Shirai, Y.: A view-based outdoor navigation using object recognition robust to changes of weather and seasons. In: Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2974–2979 (2003)
3. Cummins, M., Newman, P.: FAB-MAP: probabilistic localization and mapping in the space of appearance. *Int. J. Robot. Res.* **27**(6), 647–665 (2008)
4. Valgren, C., Lilienthal, A.: SIFT, SURF and seasons: long-term outdoor localization using local features. In: Proceedings of European Conference on Mobile Robots, pp. 253–258 (2007)
5. Bradley, D.M., Patel, R., Vandapel, N., Thayer, S.M.: Real-time image-based topological localization in large outdoor environments. In: Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3062–3069 (2005)
6. Morita, H., Hild, M., Miura, J., Shirai, Y.: View-based localization in outdoor environments based on support vector learning. In: Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3083–3088 (2005)
7. Miura, J., Yamamoto, K.: Robust view matching-based Markov localization in outdoor environments. In: Proceedings of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2970–2976 (2008)
8. Yamagi, Y., Ido, J., Takemura, K., Matsumoto, Y., Takamatsu, J., Ogasawara, T.: View-sequence based indoor/outdoor navigation robust to illumination changes. In: Proceedings of 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1229–1234 (2009)

9. Milford, M.J., Wyeth, G.E.: SeqSLAM: visual route-based navigation for sunny summer days and stormy winter nights. In: Proceedings 2012 IEEE International Conference on Robotics and Automation (2012)
10. Dalal, N., Briggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of 2005 IEEE Conference on Computer Vision and Pattern Recognition, pp. 886–893 (2005)
11. Morita, H., Hild, M., Miura, J., Shirai, Y.: Panoramic view-based navigation in outdoor environments based on support vector learning. In: Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2302–2307 (2006)
12. Ohta, Y., Yamada, H.: Pattern matching using dynamic programming. *Inf. Process.* **30**(9), 1058–1066 (1989)
13. Yamauchi, B.: A frontier-based approach for autonomous navigation. In: Proceedings of the 1997 IEEE International Conference on Computational Intelligence in Robotics and Automation, pp. 146–151 (1997)
14. Ardiyanto, I., Miura, J.: Real-time navigation using randomized kinodynamic planning with arrival time field. *Robot. Auton. Syst.* **60**(12), 1579–1591 (2012)

Using OpenStreetMap for Autonomous Mobile Robot Navigation

Patrick Fleischmann, Thomas Pfister, Moritz Oswald and Karsten Berns

Abstract In this paper, the integration of OpenStreetMap (OSM) geodata to a robot system which focuses on autonomous off-highway driving is presented. It is shown, how the OSM data is enriched with other data sources and how the map information is processed to generate a path that fits to the capabilities of the robot. Based on the map information, the quality of Global Satellite Navigation System (GNSS) signals is estimated and incorporated into the routing process, e.g. to avoid path with a high probably of GNSS disturbances. Furthermore, it is demonstrated how the robot's localization based on a Carlson filter can be improved by these estimations.

Keywords OpenStreetMap · GNSS navigation · Map-aided localization · Navigation · Digital maps · Robot

1 Introduction

A typical task of autonomous mobile robots is navigating to a predetermined target. In the unstructured outside world this is a challenging mission that can be hardly mastered efficiently over longer distances using the on-board sensors. Although the beeline to the destination can be easily determined using a GNSS, in most cases this direct route is not passable due to buildings, terrain, vegetation or waters. In addition, the use of existing paths is often a safer and faster way to reach the goal.

P. Fleischmann (✉) · T. Pfister · M. Oswald · K. Berns
Robotics Research Lab, Department of Computer Science, University of Kaiserslautern,
Gottlieb-Daimler-Str., 67663 Kaiserslautern, Germany
e-mail: fleischmann@cs.uni-kl.de

T. Pfister
e-mail: pfister@cs.uni-kl.de

M. Oswald
e-mail: m_oswald11@cs.uni-kl.de

K. Berns
e-mail: berns@cs.uni-kl.de

If you pass the problem to a human being, it is likely that he will use a map to solve this task. Those have become so natural for us, particularly through the widespread usage of car navigation systems and smart phones, that only disasters like the earthquake in Haiti in January 2010 make clear how important an up to date map is. Here, it was hardly possible for the aid workers to distribute needed supplies efficiently because infrastructure such as roads and bridges were mostly destroyed or obstructed. Only the provision of up-to-date satellite images from which volunteers of the OpenStreetMap [1] (OSM) project extracted roads and bridges still drivable,¹ were a remedy to this.

That accurate maps and detailed prior information of the environment can be a critical factor for the success of autonomous driving can be seen at the remarkable results that Google regularly demonstrates in their self-driving car project. Here, extremely detailed maps containing information about the exact shape of the road and even on the position of curbs and the height of traffic lights is encoded [2].

In this paper, we summarize the integration of OSM geodata to our robot system which focuses on off-highway driving. Although the OSM data cannot compete with the maps that Google creates for their self-driving cars, the map contains useful information besides the highways usually available in car navigation systems. To mention only a few aspects, OSM includes information on forest and field tracks and paths, cycle ways, steps, the shape of buildings, forests, land use and even individual trees or power poles. Furthermore, it has been shown in multiple evaluations [3], [4] that OSM even outperforms commercial datasets which are often limited to car navigation.

The first part of this paper focuses on the calculation of an optimal path from a given start pose to a certain destination. Therefore, the downloaded OSM data is enriched with information from other sources in a local database and transformed to a topological map represented by a graph. Furthermore, the routing algorithm incorporates desired path characteristics—e.g. small footpaths and highways are not allowed—and vehicle capabilities like maximum slopes. The routing functionality has also been extended to take information about areas into account, where GNSS disturbances like multi-path effects or no-line-of-sight measurements can be expected and thus tries to avoid these paths. In the second part, it is shown, how the result of a localization filter can be improved by estimating the quality of the GNSS signal shadowed by nearby bridges, tunnels and buildings.

2 Related Work

Several authors showed that the OSM geodata is suitable to improve different key components of an autonomous robot such as localization, path planning or to improve the perception. Hentschel and Wagner [5] were one of the first authors who proposed

¹http://wiki.openstreetmap.org/wiki/Humanitarian_OSM_Team.

the integration of OSM data to their robot system. On the one hand they used the street data to calculate a path from a given start to a destination pose. Furthermore, the street network information was used to switch the head lights and the turn signals. On the other hand they extracted the footprint of buildings to match it with 3D laser data to improve their localization system. Pereira et al. [6] proposed the usage of OSM data for global navigation purposes. Between two intersections extracted from the map, they perform a lane following strategy while near the critical intersection points the extracted angles and directions are used to support the local navigation maneuvers.

In [7] a method called OpenStreetSLAM is introduced, where the authors used the OSM information to compensate the accumulated drift of state-of-the-art visual odometry (VO) algorithms. Therefore, they apply chamfer matching to align the detected path with the road network extracted from the map. A related approach by Brubaker [8] combines VO and OSM data without incorporating GPS information. Here, the researchers present an algorithm to match between the VO estimation and the OSM data modeled as a graph. Recently, Ruchti et al. [9] proposed a method also relying on a Monte Carlo filter which uses a 3D laser range finder to match the road network extracted from OSM with range data classified into road and non-road. Similarly, also Mandel et al. [10] use OSM data to fuse it with odometry and GPS measurements to improve the localization of mobility platforms for elderly people care. Using prior knowledge about the devices, e.g. that a wheelchair will not drive on highways, false hypothesis are eliminated.

The detection of GNSS disturbances and the improvement of GNSS-based localization has been investigated by many other research groups. An innovative approach has been demonstrated in [11], where an omnidirectional infrared camera pointing to the sky is used to exclude shadowed satellites from the position calculation. The approach requires a large number of visible GPS and GLONASS satellites and special camera equipment. The latter drawback is improved by [12], where the expensive infrared system was replaced with a normal fisheye camera. In [13] an approach which relies on OSM data in conjunction with digital terrain and elevation data is shown. Using 3D raytracing the pseudoranges of satellites blocked by a building are excluded. This information is used in a Kalman filter where the sensor model was adapted to handle and predict pseudoranges and where the data is fused with odometry measurements. Similarly, the approach proposed by Miura et al. [14] uses 2D OSM data in combination with 3D digital surface models to create a map with 3D buildings. For each position calculation candidates are sampled around this position in a grid-like fashion. In combination with a ray tracing algorithm, the visibility of each satellite is estimated for all candidates and the most likely one is chosen based on the measured signal strength compared with the estimated visibility.

3 OpenStreetMap and Data Representation

The OSM data representation consists of three basic elements, namely *nodes*, *ways* and *relations*. *Nodes* can be used individually to mark single feature points (like bus stops or a fire hydrant) or in a group to define the shape of a road, a path or a building. Thereby, a *node* consist at the minimum of a unique identifier i and a geographic coordinate specified by a latitude φ and a longitude λ attribute. The resolution of both coordinates is limited to 7 decimal places which allows to encode an accuracy of 11.132 mm at the equator but this accuracy cannot be expected. Additional attributes like a timestamp t , a version and information about the creator of the *node* can be existing as well, but only the timestamp is of interest for this work. If the *node* is used to model a point feature, it can be detailed with further information in a child node named *tag*. These tags always consist of two attributes, a key k and a value v , and can also be used to encode altitude information. This elevation data would be very helpful for autonomous mobile robots but is only rarely present in the OSM database.

Except from latitude and longitude, *way* objects share the same attributes as *nodes*. A *way* consist of several *node* objects specified in children named *nd* and referenced by their id in an attribute called *ref*. Here, the order of the *node* references is of importance and has to be kept when the data is interpreted. Possible tags of *way* elements can be classified in two groups. The first group contains paths that actually represent walk- or drivable ways, like streets or footpaths. In the data, these *ways* can be identified by the *highway* key which can have a bunch of values ranging from *motorway* to *path*. The second group represents polygonal shapes which are used to model areas such as land uses, parks or building footprints. Both types of paths can be further detailed with tags, e.g. the number of levels which was used to estimate the height of the buildings in this work.

The last class *relation* represents groupings of multiple *nodes*, *ways* and/or other relation objects. For example, several *ways* together can form a bus line which is represented by a relation. Several of these bus line *relations* can then form a public transport network. Additionally, several *way* objects which represent polygons can be combined or cut to a relation, which allows to create multi-polygons. These, for example, can be used for buildings with a courtyard or to create coherent objects for complex areas like the Palatinate Forest. All types of relations are further specified using tags.

4 Acquisition and Preparation of OSM Data

The mission area can be specified manually using a bounding box or automatically concerning the robot's current position and the goal point which should be approached. For the first option, a plugin for the graphical tooling of the Finroc²

²<http://finroc.org>.

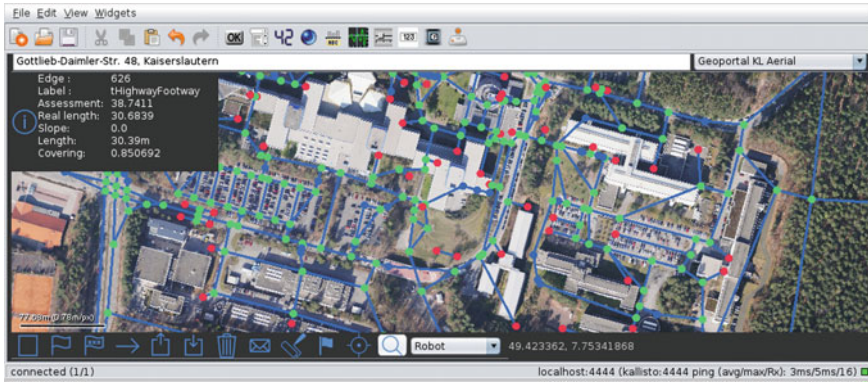


Fig. 1 A topological graph deduced from the OSM data superimposed on an aerial image of the campus of the University of Kaiserslautern. Intersections are shown as *green* nodes, dead ends are depicted in *red*, and *blue* nodes represent positions where the way type changes

robotic framework has been developed which allows to select the area of interest based on aerial images. An example of this extension is shown in Fig. 1. The extracted boundaries of the selected map area specified by two pairs of geographic coordinates are then passed to a download module which requests the data from the OSM servers. The default OSM-API only allows download queries up to a fixed amount of data which can result in the need of multiple requests for a certain map area which causes additional delays. Therefore, the data is downloaded using the alternative overpass-API³ which provides the advantage of less restricted queries combined with the opportunity to formulate more precise requests resulting in a more performant download process.

Downloaded XML data is parsed, filtered, cleaned up automatically and stored in a local MySQL database for different reasons. Firstly, the data can be accessed in a performant way and more important, it is available offline, as an Internet connection can not always be expected in all regions of our test ground, the Palatinate Forest. Secondly, the database can be extended with own data collected during tests and missions and updates from the OSM project can be easily merged by comparing the timestamps provided with the data.

The main data of interest are all *ways* which are labeled as *highways* since they are used to represent all possible kinds of drivable or walkable roads. This information is used to create a graph of the provided road network in which the robot is deployed. To speed up routing, the network is reduced to a topology as shown in Fig. 1. Therefore, nodes which are referenced by more than one *way* are identified by a SQL statement. These intersections (green nodes) are stored in the database together with identified dead ends (red nodes) to accelerate future routing tasks. Furthermore, the edges connecting these navigation nodes are also extracted and stored persistently.

³http://wiki.openstreetmap.org/wiki/Overpass_API.

Additionally, *ways* tagged as buildings, land use and amenity are of special interest for an autonomous robot, as they provide information about footprints of buildings, which can be used for localization purposes as well as surroundings of ways which can be considered for local path planning. Moreover, altitude data is also beneficial for the routing process, e.g. to avoid steep hill climbs. Whereas OSM datasets only rarely provide altitude data for *nodes*, this information is obtained from other data sources. The altitude information is stored permanently in the MySQL database as an additional field for the *node* objects and is extracted after processing the downloaded OSM XML file. Our implementation uses two different data sources to obtain this information for each *node*. The primary source is a non-free digital terrain model (DTM) with a grid size of $1\text{ m} \times 1\text{ m}$. This dataset has been created using aerial laser scanning and is offered by the State Office of Surveying and Geospatial Basic Information of Rhineland-Palatinate. The data is provided as a list of equally spaced triples where each entry is composed of an easting and a northing coordinate of the Universal Transverse Mercator (UTM) coordinate system as well as the height above the sea level in meters. To extract the altitude for a given OSM-*node*, the geographic coordinates (φ, λ) have to be converted to the UTM coordinate system first. Afterwards, the altitude is calculated by a distance weighted average of the four closest coordinates provided in the DTM dataset. If the robot is deployed outside the boundaries of this high resolution DTM, the Google Elevation Service is employed to deduce a *node*'s altitude. Google bases this service on different sources like the NASA's Shuttle Radar Topography Mission (SRTM)⁴ and interpolates the requested elevation from the four nearest points. Since the current SRTM dataset (2014) provides a resolution of 1-arc second (respectively 30 m), the obtained elevation data quality varies and therefore can only be seen as rough criteria which has to be handled with care.

In addition to the topological graph and the data stored in the MySQL database, a local polygonal map is build based on the OSM information. As demonstrated in Fig. 2, the polygons are used to represent objects of special interest like buildings, tunnels, bridges and forest areas. The polygonal representation allows to extract different properties which are used within the routing process but also to support the localization system (see Sect. 6). Additionally, also properties for path execution and for the perception system can be deduced from this map, e.g. reducing the velocity if the robot is driving close to a building or to handle increasing noise and exposure problems when driving through a tunnel. The map also allows for conclusions whether a pose lies on a parking lot where potentially parked cars have to be taken into account, and as well in which scenic surrounding the pose is located, e.g. in a forest, a field or a housing area.

⁴<http://www2.jpl.nasa.gov/srtm/>.

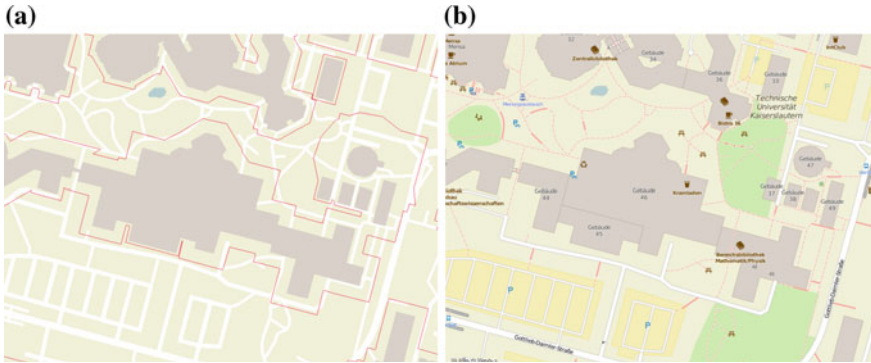


Fig. 2 **a** Visualization of the local polygonal map with calculated “GNSS shadows” using the building’s height shown as *red lines*, **b** corresponding OSM representation (*right*)

5 Routing with GNSS Visibility Constraints

Routing from a start to a goal point is founded on the extracted topology. Therefore, the edges of this graph are weighted according to a modular weighting function and then used as base for an state of the art A*-search algorithm determining the cheapest way. The weighting function consists of different components which weight an edge according to its length, way type, maximum slope, and height differences. All components are allotted with different significances by the user and can easily be extended. This allows to fit the features of the computed route to different robots in varying scenarios and user preferences. Additionally, a complete way class, e.g. steps, can be excluded from the A*-search to adapt the route to the robot’s capabilities by setting the weight to ∞ .

As a novel extension, also the expected quality of the GPS signals and thus the reliability of the localization system has been integrated into the routing process. To estimate this measure for each edge of the topological graph, the polygonal map is employed to calculate “GPS shadows”. Therefore, the building and bridge polygons are enlarged using an offset algorithm as a first step. Each margin m_i used as the offset to buffer a polygon is calculated based on the given levels l_i of a building or a bridge provided by the OSM-project as shown in Eq. 1.

$$m_i = \frac{l_i \cdot h_{avg}}{\tan(\alpha)} \quad (1)$$

Here, h_{avg} is the average level height which was set to 3 m for buildings and 4.5 m for bridges. α describes the elevation angle of the lowest satellite which should be included in the estimation. This value can either be set to a fixed value or extracted from the \$GPGSV-NMEA messages. Afterwards, buffered polygons which intersect or which are contained by others will be merged. Secondly, the so called “coverage” of all edges is calculated based on the merged shadows. To do so, the part of the

way which lies within the shadow polygon is compared to the overall length to get a percentage covering measure c_i . Similarly, parts of a way which cross a tunnel or which course below a building are assumed to be completely shadowed. As introduced above, the covering is incorporated into the weighting function as a further component.

6 Improving the Localization Using OSM Data

While the previous explained routing approach tries to avoid areas with a high risk of GNSS corruptions, it is not always feasible to bypass these. Instead of avoiding these paths completely, the already extracted information about the expected quality can be used to improve the localization estimations in such situations. Therefore, a modified square root localization filter is employed. In principle, the filter fuses the measurements from the wheel and steering encoders, an inertial measurement unit and the GNSS receiver to a common localization estimation. Typically, a conventional Kalman filter is used for this task, but a square root filter is known to provide a higher numerical robustness. In the case of the localization application, this is especially useful as the variances of the state vector may differ by several magnitudes. The localization filter implements the square root filter formulation presented in [15]. In the filter, the movement of the kinematic center, modeled by a state space representation in three dimensions, is tracked. Multiple tasks are solved by the localization filter in this application. First, the sensors are combined to generate a consistent common estimation of the most important localization states. None of the single sensors can provide measurements of all these states with a sufficient accuracy. Furthermore, the position measurements of the GNSS are smoothed, so that the resulting trajectory does not contain any unrealistic jumps. This is a necessary property, required by the local obstacle mapping and navigation tasks. Finally, the filter approach can be used to bridge GNSS dropouts.

For the filter, all scalar measurements of each sensor i have to provide a sensor model to be incorporated into the common estimation. These sensor models include the measurement noise variance $R_{k,i}$, the actual measurement value $z_{k,i}$ and a measurement sensitivity matrix H_i . The measurements have to satisfy the following condition:

$$z_{k,i} = H_i x_k + v_{k,i} \quad (2)$$

Thereby, k denotes the time point and x_k is the true localization state containing the vehicle position, the orientation, the attitude and the velocity. The sensor uncertainty is represented by $v_{k,i}$, which is assumed to be a Gaussian distributed random number with a variance $R_{k,i}$. According the Kalman filter assumptions, $v_{k,i}$ should be independent over time. The influence of a sensor i on the fused output mainly depends on the variance $R_{k,i}$. When some satellites, used by the GNSS receiver, are shadowed by buildings, the determined position by the receiver will start to drift from the true position. This behavior is caused by multipath interference and no-line-of-sight mea-

surements. A drift, however, significantly violates the Kalman filter assumptions of independent measurement errors. Consequently, also the filter position will follow this drift and the estimated state will get corrupted. To avoid such a behavior the influence of the GNSS measurements is reduced stepwise. In the first step, the variance $R_{k,\text{GNSS}}$ is increased by a factor f_k when the vehicle approaches a building:

$$f_k = f_{\max} - f_{\max} \frac{d_{\min} - d_k}{d_{\min} - d_{\max}} + 1 \quad (3)$$

The factor f_{\max} denotes the maximum factor applied to the variance of the GNSS sensor model. The influence starts to decrease at a distance of d_{\max} to the nearest building and has the lowest influence at a distance of d_{\min} . If the vehicle is closer to a building than d_{\min} , the measurements are excluded completely. As soon as the vehicle moves away from the building, the influence is increased again by the factor f_k . This procedure results in a smooth estimated trajectory which avoids unsteadiness caused by the drifts of the GNSS.

7 Experiments

In this chapter, two selected experiments which demonstrate the results of the implemented approaches are shown. The first experiment shows the routing based on OSM data and GNSS constraints while the second one focuses on the localization improvements that have been archived by integrating the OSM data into the localization filter as introduced in the previous chapter. Both experiments have been conducted on a modified John Deere XUV 855D Gator vehicle. The platform features a four-wheel drive system, a diesel engine with 17 kW, an automatic gearbox and 4 fully independent dual a-arm suspensions which makes it especially suitable for off-road driving in rough terrain. To turn the series vehicle into an autonomous car, a Drive-by-Wire system has been installed. Thus, the vehicle is manually drivable as well as completely controllable by a computer, ranging from the steering, the brakes, and the gear shift to the lights and the ignition. The localization sensor suite consist of a u-blox NEO-7P GNSS module, a MicroStrain 3DM-GX3-25 inertial measurement unit and absolute encoders to measure wheel speed and angular velocity. Furthermore, a John Deere StarFire 3000 DGPS receiver with SF2 corrections has been mounted to generate ground truth data.

7.1 Routing

This experiment covers two routes from the same starting point to the same goal. The first possibility is a result of just using the length l_i as a weight $w_i = l_i$ in the A* algorithm which generates an overall path of an estimated 270.71 m length.

For the second route, the estimated GNSS signal quality was incorporated into the weighting function: $w_i = (1 + c_i) \cdot l_i$, where c_i is the percentaged covering measure as described in Sect. 5. Thus, the routing algorithm finds another possibility which has a lower coverage compared to the first one. This second path is slightly longer (275.84 m) but only 17.4 % of the path are lying within the calculated GNSS shadows instead of 59.21 % of the first variant.

To compare these estimations with the real world, both routes have been driven with the test vehicle talking the trajectory predefined by the OSM data into account and keeping the car on the paved roads of the campus. The maximum speed was limited to 3 m/s which is automatically reduced in sharp curves. To estimate the quality of the GPS signal, the signal-to-noise ratio (SNR) of all satellites used for position estimation has been compared. The results are as follows: the measured length of the first route (shown in red color) is 261.6 m while the alternative (green) is about 5.6 m longer. According the GPS quality, the first variant (red) has an average SNR which is 2.01 dB lower than on the second path (green), mainly caused by a tunnel in the first part of the route and the proximity to high buildings in the last section. Detailed plots of the measurements are given in Fig. 4 while traces of both drives are shown in Fig. 3.

7.2 Localization

For the localization experiment, a path of 120 m length has been selected which includes a tunnel of 23.8 m length which goes through a building. To better demonstrate the drift effect of the GNSS receiver—which seems to be smoothed due to internal filtering at higher speeds—the vehicle was stopped for 30 s inside the tunnel. The scenario is depicted in the right part of Fig. 5, where the position with the

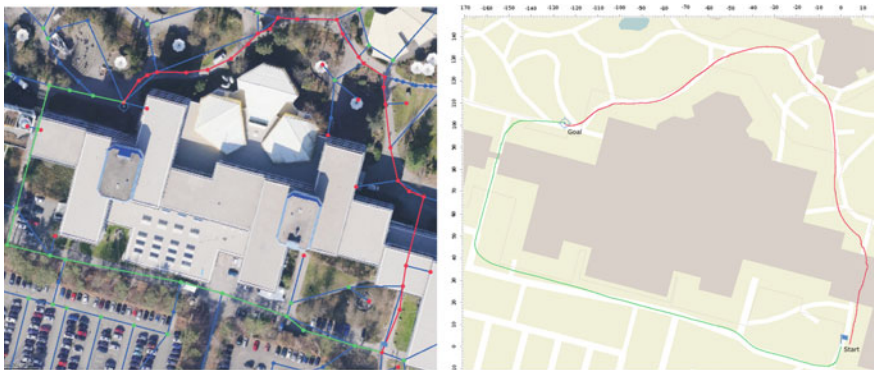


Fig. 3 Two alternative routes from a given start to a given goal. The red variant is expected to have more than 3 times more areas with GPS disturbances than the green route. The right drawing shows traces of both routes driven with the Gator vehicle

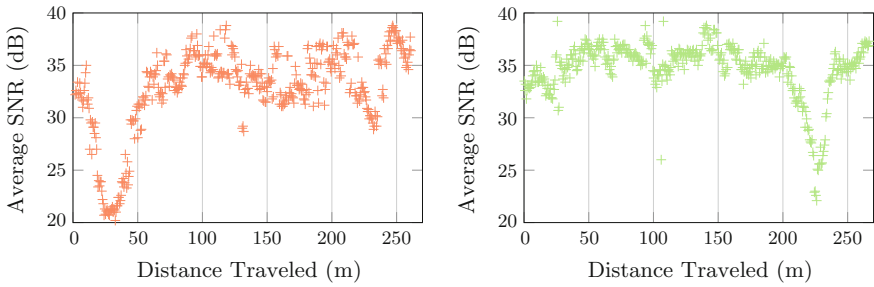


Fig. 4 Averaged signal-to-noise ratio (SNR) of all satellites used for the position calculation: (left) of the red route shown in Fig. 3, (right) the signal quality of the green one



Fig. 5 The vehicle drove from point 1 to point 3 with an intermediate stop at point 2. As the stop is lying inside a tunnel, the GPS-measurements (red line) are heavily drifting. The output of the filter is segmented in multiple stages: no influence part (green), influence part with an increased variance in the localization filter (blue) and a yellow part, where the GPS measurements are not incorporated at all

label 1 marks the start position, 2 the intermediate stop and 3 the end position. The start and the end positions have been measured by the StarFire DGPS receiver and the intermediate point was calculated using a cadastral building map. On the left side of Fig. 5, the red line shows the GPS measurements (10 Hz sampling rate) with a huge drift after stopping in the tunnel. The second trace (green, blue and yellow) shows the result of the proposed filtering approach using OSM information. For this experiment, the factor to increase the variance f_{max} was set to 50. Furthermore, the GNSS receiver input was disabled, if the vehicle was closer than $d_{min} = 3$ m to the building. For distances larger than $d_{max} = 30$ m, no influence of the buildings to the GPS measurements was assumed. To clarify the filtering steps, the different sections are color coded in the image. For the green part, the variance of the GNSS sensor model uses a fixed parameter set which was determined for the u-blox receiver. In the yellow areas, the GNSS sensor was completely excluded from the position determination—as at this point just odometry and the IMU data was used—as the algorithm detected that the

vehicle is driving through a building or a tunnel. For the parts which are coded in blue, the distance between the current position and the closest obstacle based on the OSM data was used to increase the variance of the GPS sensor model with the exponential function given in Eq. 3. The plot of the traces shows that the accuracy of the localization system could be significantly improved by the proposed usage of OSM geodata as the filter does not follow the heavy GPS drift inside the tunnel and the GPS measurements do not distort the localization when driving close to a building.

7.3 Conclusion

In this paper, a simple approach to estimate GNSS disturbances based on OSM data and to incorporate this information to the weighting function of a routing algorithm has been demonstrated. The algorithm determines completely different routes compared to the traditional time or length based weights which can lead to a more precise localization and thus safer robot navigation. Furthermore, an approach is introduced how the OSM data can be used to adaptively modify the influence of GNSS measurements to a localization filter used on our autonomous robot. Using the map information, the localization can be significantly improved, especially if the GNSS based positioning is heavily affected by no-line-of-sight measurements, e.g. close to buildings or inside tunnels.

References

1. Haklay, M., Weber, P.: Openstreetmap: User-generated street maps. *IEEE Pervasive Comput.* **7**(4), 12–18 (2008)
2. Madrigal, A.C.: The trick that makes google's self-driving cars work. *The Atlantic* **5** (2014)
3. Over, M., Schilling, A., Neubauer, S., Zipf, A.: Generating web-based 3d city models from openstreetmap: the current situation in germany. *Comput. Environ. Urban Syst.* **34**(6), 496–507 (2010)
4. Neis, P., Zielstra, D., Zipf, A.: The street network evolution of crowdsourced maps: openstreetmap in germany 2007–2011. *future Internet* **4**(1), 1–21 (2011)
5. Hentschel, M., Wagner, B.: Autonomous robot navigation based on openstreetmap geodata. In: 2010 13th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE, pp. 1645–1650 (2010)
6. Pereira, E.C., De Lima, D.A., Victorino, A.C.: Autonomous vehicle global navigation approach associating sensor based control and digital maps. In: IEEE International Conference on Robotics and Biomimetics, 2014. ROBOTICS'14 (2014)
7. Floros, G., van der Zander, B., Leibe, B.: Openstreetslam: global vehicle localization using openstreetmaps. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 1054–1059 (2013)
8. Brubaker, M., Geiger, A., Urtasun, R., et al.: Lost! leveraging the crowd for probabilistic visual self-localization. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 3057–3064 (2013)

9. Ruchti, P., Steder, B., Ruhnke, M., Burgard, W.: Localization on openstreetmap data using a 3d laser scanner. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2015)
10. Mandel, C., Birbach, O.: Localization in urban environments by matching sensor data to map information. In: 2013 European Conference on Mobile Robots (ECMR), IEEE, pp. 50–55 (2013)
11. Suzuki, T., Kitamura, M., Amano, Y., Hashizume, T.: High-accuracy gps and glonass positioning by multipath mitigation using omnidirectional infrared camera. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 311–316 (2011)
12. Marais, J., Meurie, C., Attia, D., Ruichek, Y., Flancquart, A.: Toward accurate localization in guided transport: combining gnss data and imaging information. *Transp. Res. Part C: Emerg. Technol.* **43**, 188–197 (2014)
13. Obst, M., Bauer, S., Wanielik, G.: Urban multipath detection and mitigation with dynamic 3d maps for reliable land vehicle localization. In: Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION, IEEE, pp. 685–691 (2012)
14. Miura, S., Hisaka, S., Kamijo, S.: Gps multipath detection and rectification using 3d maps. In: 2013 16th International IEEE Conference on Intelligent Transportation Systems-(ITSC), IEEE, pp. 1528–1534 (2013)
15. Carlson, N.A.: Fast triangular formulation of the square root filter. *AIAA J.* **11**(9), 1259–1265 (1973)

A Lane Change Detection and Filtering Approach for Precise Longitudinal Position of On-Road Vehicles

Tianyi Li, Ming Yang, Xiaojun Xu, Xiang Zhou and Chunxiang Wang

Abstract This paper presents a lane change detection and filtering approach for precise longitudinal localization. Maps, the road which is traveled along and the trajectory of the vehicle are used as the only inputs. Straight-road transformation is proposed for lane change detection on the curve, and the filtering algorithm is designed for online positioning process. Experiments demonstrate the improvement of longitudinal position precision by lane change detection and filtering, and show the application of this approach on terrain localization.

Keywords Lane change detection · Inertial measurement unit (IMU) · Longitudinal localization · Terrain localization

1 Introduction

A precise localization of on-road vehicles is a prerequisite in many advanced vehicle applications of Intelligent Transportation System (ITS). The localization accuracy for tasks such as parking should be within the decimeter range which cannot be achieved by low-cost GPS. Many techniques have been developed for precise localization, including lateral position estimation [1] and longitudinal position estimation [2].

T. Li · M. Yang (✉)

Key Laboratory of System Control and Information, Department of Automation,
Processing Ministry of Education of China, Shanghai Jiao Tong University, Shanghai, China
e-mail: MingYang@sjtu.edu.cn

X. Xu · X. Zhou

School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University, Shanghai, China

C. Wang

Research Institute of Robotics, Shanghai Jiao Tong University, Shanghai, China

© Springer International Publishing AG 2017

W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_65

Drivers have to change lanes daily because of congestion, navigation or preference, during which the ego-vehicle moves from its current lane to an adjacent lane, either on the left or right side. Lane change actions increase the travel distance and induce the estimation error of longitudinal position and road shape. As a result, changing lanes deteriorates the performance of map-relative positioning systems, e.g., localization systems using road shape information [3] or road slope information [2, 4].

These effects can be avoided by the detection and filtering of lane changes. Many studies have focused on lane change detection, and state-of-the-art approaches include inferring driver intention [5–7] and scene understanding [8–10]. As those approaches are not particularly for better positioning accuracy, they do not eliminate the effect of lane change actions. Meanwhile, inferring driver intention approaches may not work well when the vehicle drives on the winding road because of the shortage of road shape information, and scene understanding approaches, which often use vision sensors, are sensitive to poor weather and illumination conditions.

In this paper, we present a lane change detection and filtering approach for vehicle longitudinal position. It does not only detect and eliminate the lane change action on the straight road, but also on the curves and winding roads. We also do experiments to demonstrate the effectiveness of this approach.

This paper is composed of the following sections. The lane change detection approach is proposed in Sect. 2. In Sect. 3, we filter the lane change trajectory for better position precision. We discuss results of the experiments in Sect. 4 and conclude this paper in Sect. 5.

2 Lane Change Detection

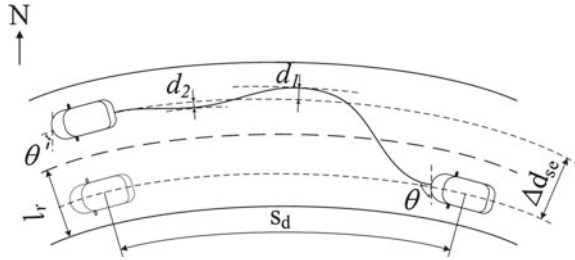
2.1 Problem Description

Given the trajectory of a vehicle, we would like to remove all the lane changes in it and output a new trajectory without the noise of lane change. Specifically, we take the following things as input:

- A sequence of discrete points, say $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, which represents the given trajectory of a vehicle.
- A map. This would give us some information of the roads we're driving on, such as road length, road curvature, etc.

For convenience, we would also use the “direction angle” of the vehicle at each point, θ_i , directly, which could be easily obtained by calculating the slope $(y_{i+1} - y_i)/(x_{i+1} - x_i)$.

Fig. 1 Lane change trajectory: the characters are used to describe the feature of lane change



After lane change filtering, the desired output would be:

- A list of time intervals, recording which parts of the trajectory are lane changes.
- A sequence of discrete points, $(x'_1, y'_1), (x'_2, y'_2), \dots, (x'_n, y'_n)$, the new trajectory which removes all lane changes from the old one.

2.2 Lane Change Definition

Lane change is an action that a vehicle moves from its current lane to an adjacent lane, either on the left or right side. The typical trajectory of a whole lane change is depicted in Fig. 1, in which a vehicle travels from the left lane to the right one on a curve. In general, continuously changing lanes is not recommended in driving rules, so changing several lanes at a time is beyond our consideration.

2.3 Feature Selection

We select three features to represent a part of the trajectory and determine whether it is a lane change.

2.3.1 The Lateral Displacement

The lateral displacement is the projection of the displacement between the start point and the end point on the direction vertical to the driving direction. As is shown in Fig. 1, Δd_{se} is the lateral displacement of a vehicle during the lane change. The lateral displacement of a lane change trajectory must satisfy the inequality: $\Delta d_{se} \geq l_r$, where l_r is one-lane width. It is difficult to tell the difference between the lane change trajectory and a turning trajectory if the lateral displacement is regarded as the only judgment.

2.3.2 The Yaw Angle Displacement

The yaw angle displacement means the angle transition during lane change, and its symbol is $\gamma = |\theta' - \theta|$. $\gamma \approx 0$ if lane change happens in the straight road, and $\gamma \approx \alpha s_d$ if the roadway is curved. In the paper, the road curvature of one possible location can be obtained from the map.

2.3.3 The Lane Change Duration

A whole lane change usually lasts for 2–5 s [11], so the feature of the lane change duration is $2 \leq \Delta t \leq 5$.

2.4 Lane Change Detection Based on Maps

We adopt a bidirectional graph to represent the map data illustrated in Fig. 2, in which nodes contain the shape and slope information of street segments, and edges represent the connectivity of roads [12]. The road shape information can be extracted from the existing digital map, e.g. OpenStreetMap (OSM). One street is broken into several road segments, in which there is no intersection, and the shape of each segment should be arc or line.

If the vehicle is traveling on a straight road, we can calculate the features as following. Suppose the beginning coordinate of the segment is (x_1, y_1) with beginning direction is θ_1 , while the end ones are (x_2, y_2) and θ_2 , then the lane change detection features are:

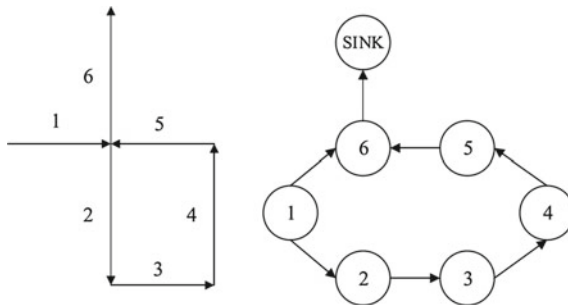


Fig. 2 The map structure is shown in the figure. The *left* graph is the road network map and the *right* one is a transformed bidirectional graph. In the *left* graph, the directed lines represent the street segments and their driving directions. If there are two-way roads, every two-way road is divided into two overlapped one-way roads. The nodes are common points of street segments. In the right graph, the node means a unit which contains the information of a street segment, and the directed line means the connectivity of street segments

$$\gamma = |\theta_1 - \theta_2| \tag{1}$$

$$d_{se} = |d_{12} \sin(\arctan(\frac{y_1 - y_2}{x_1 - x_2}) - \theta_1)| \tag{2}$$

where $d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

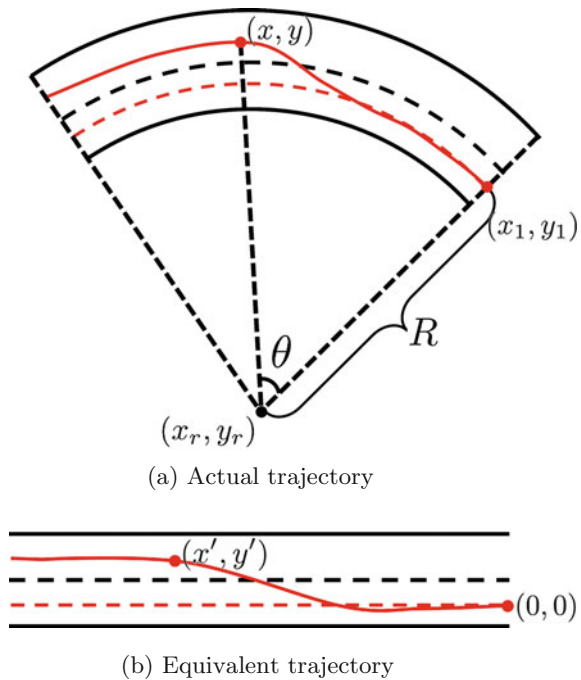
The segment is regarded as lane change trajectory on a straight road when $\gamma < \gamma_0$ and $d_{se} > d_0$, where γ_0 and d_0 are parameters chosen according to the road situation. If the vehicle is traveling on a curve, the case will be different from that on a straight road. We assume that a curve is in the shape of a circular arc, and the radius and center of the circle can be extracted from maps. Our main idea is to convert the trajectory into the one on straight roads. That is, for the trajectory on curves, we choose to map it into a new equivalent trajectory on straight roads. Given the center (x_r, y_r) and the radius R , the function of straight-road transformation is

$$x' = R\theta \tag{3}$$

$$y' = \sqrt{(x - x_r)^2 + (y - y_r)^2} - R \tag{4}$$

where $R = \sqrt{(x_r - x_1)^2 + (y_r - y_1)^2}$ and $\theta = |\phi - \phi_1|$. The meaning of symbols in (3) and (4) is shown in Fig. 3.

Fig. 3 The lane change at a curve and the transformed result



Thus, we can use the method of straight roads on the equivalent trajectory of curves. The experimental result is shown in Sect. 4.

3 Lane Change Filtering Algorithm

After the process of lane change detection, lane change filtering algorithm is designed to eliminate the impact of lane change on vehicle longitudinal position. As a result, our purpose is to remove the lateral displacement from the trajectory and derive a new trajectory. For example, the vehicle’s location after lane change filtering is adjusted to the dotted vehicle in Fig. 1.

For a lane change on the straight road, we define data obtained at the start time and the end time of the trajectory as p_{stt} and p_{end} respectively. Then the lane change-removed trajectory is generated by:

- (a) translate the location of p_{stt} (and for every point before p_{stt}) in the lateral direction by d_{se} , forming a new point $p_{stt'}$;
- (b) for every point between p_{stt} and p_{end} , translate it in the lateral direction to form a line between $p_{stt'}$ and p_{end} .

For process (a), the point (x, y) will be transformed as:

$$x' = x + d \sin \alpha \tag{5}$$

$$y' = y - d \cos \alpha \tag{6}$$

$$\theta' = \alpha \tag{7}$$

where

$$\alpha = \frac{\theta_{stt} + \theta_{end}}{2} \tag{8}$$

$$d = (y_{end} - y_{stt}) \cos \alpha + (x_{end} - x_{stt}) \sin \alpha \tag{9}$$

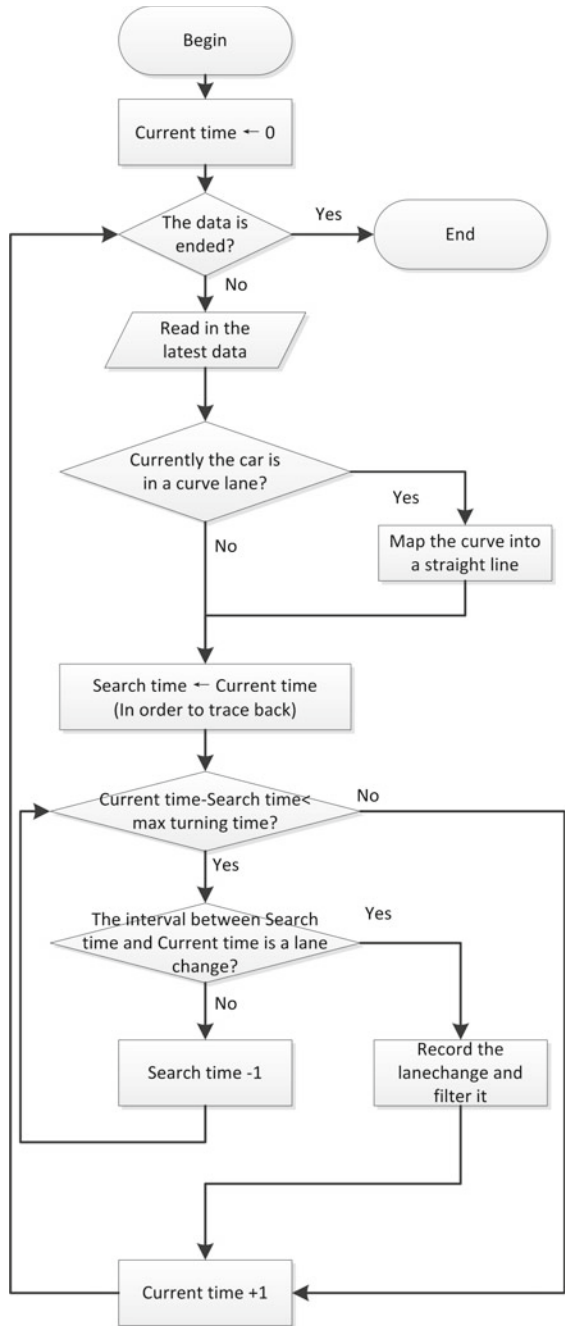
While for process (b), the transforming formula is almost the same except that Eq. (9) becomes:

$$d = (y - y_{stt}) \cos \alpha + (x - x_{stt}) \sin \alpha \tag{10}$$

Given a stream of sensor data, we can calculate the location (x, y) and direction θ of the vehicle. When a new tuple of data is read in, we’ll trace back and see whether there exists a lane change trajectory that ends at current time. If there exists one, then use the filter algorithm to remove the lane change. The process is shown in Fig. 4.

The lane change filter for improving the map precision can perform off-line, using the vehicle’s trajectory and the road shape map as the priori knowledge. While localization, the trajectory should eliminate the influence of lane change on-line

Fig. 4 The flow chart for the algorithm of lane change detection and filtering



to improve the position accuracy. When the lane change action is found out, the sequence of this segment will be modified and the vehicle should be relocated.

It is important to note that the lane change detection and filtering approach should be shut down when the vehicle drives at the intersection corner in case of false detection. It's hard to determine the lane with the turning trajectory especially after the curve to straight transformation, which is in various shapes related to the number of lanes. Fortunately, the vehicle isn't allowed to change lane at the intersection, so the operation will not lead to missed detections.

4 Experimental Evaluation

To evaluate the approach in realistic situation, experiments were performed in the region of Shanghai Jiao Tong University (SJTU) Minhang Campus which covers an area of about 4 km². Raw data were collected in an instrumented vehicle which is equipped with one RTK-GPS, a Xsens MTi-28A53G35 IMU and two high-precision encoders mounted on two rear wheels respectively. RTK-GPS data are used as the reference. Biases and high frequency noises from IMU are estimated and removed from the raw data with the statistical method. The experiment program ran on a computer with Intel (R) Core (TM) i7-4710 MQ CPU @ 2.50 GHz, 8G Memory and Ubuntu 14.04 64 bit operating system.

The vehicle was driven at nearly 30 km/h and sampled four trajectories in the campus at the rate of 10 Hz. The trajectories are shown in Fig. 5, in which the vehicle changed lanes about fifty times totally.

In the experiments, the vehicle changed lane nearly 40 times not only on the straight roads, but also on the curves. The approach can detect all times of lane change successfully and we adjusted the recorded sequences for better estimate of longitudinal position. With map data, the curve to straight projection can be done when the vehicle is driven on the curve, which is shown in Fig. 6. The zoomed filtering result on the straight road and the curve is shown in Fig. 7.

It is noted that the lane change detection approach can hardly run normally based on wrong road information or inaccurate maps. Fortunately, it cannot get any worse to filter the lane change trajectory when the vehicle is localized on the incorrect road or the maps are inaccurate. Nevertheless, the aim to filter lane change is to improve the accuracy of vehicle localization based on maps, so we could only detect lane change with the right road. For fine positioning, the map should be very accurate for map-relative localization. In a word, the lane change filter module will perform well if we know the road where the vehicle is located and the maps are with enough resolution.

It is noted that longitudinal distance estimation is of vital importance to the accuracy of road slope information which is the main feature of terrain localization. However, changing lane increases travel distance and leads to the descent of road slope estimation. To validate the necessity of lane change filtering, we conducted an experiment for terrain localization. The lane change filter was performed based on the

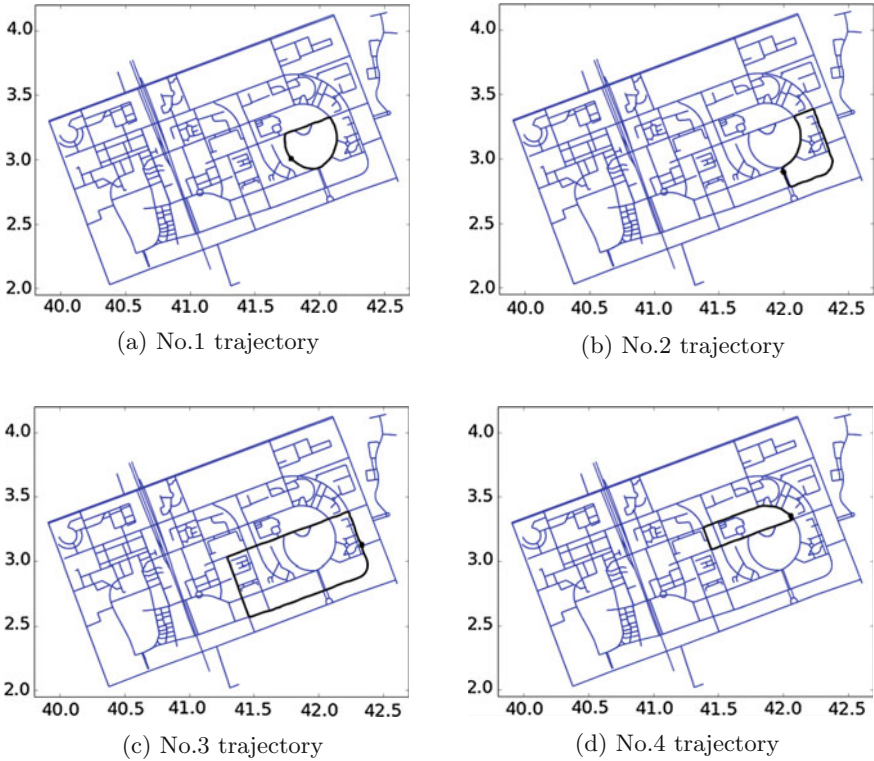


Fig. 5 Travel trajectory in the road shape map. The trajectories are all closed cycles, and the starting point is shown as a *circle*

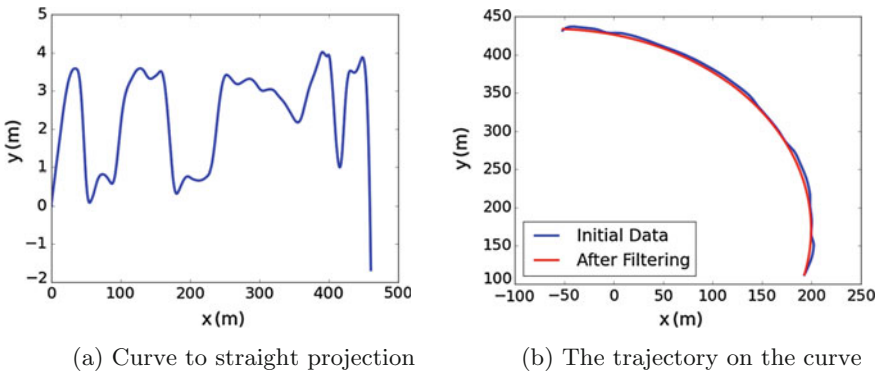


Fig. 6 The lane change filter on the curve. The origin of coordinate is the starting point of the filter

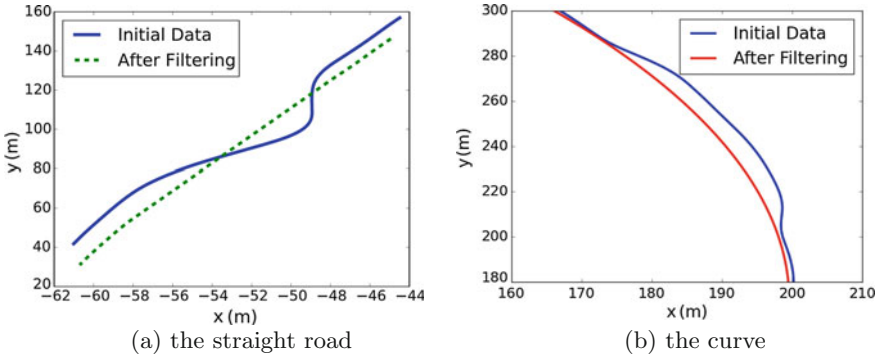
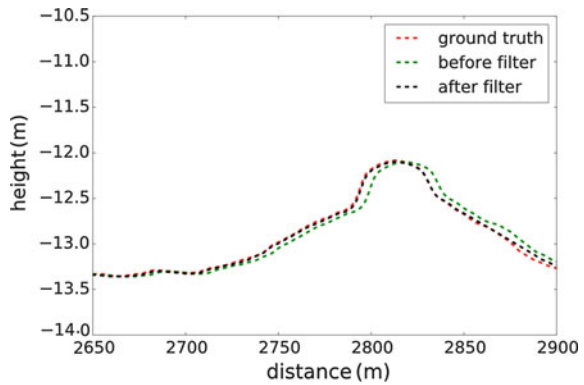


Fig. 7 The detailed view of lane change filter on the curve

Fig. 8 The slope result of lane change filtering



assumption that road slope only changes in the longitudinal orientation of the road. As is shown in Fig. 8, lane change filtering is necessary to reduce the error of longitudinal distance and road slope estimation. In the figure, the vertical axis means the height relative to the starting point and the horizontal axis means travel distance from the starting point. After 2500 m driving, the trajectory is longer than longitudinal distance. The truth road slope information of the maps were also collected and shown for performance evaluation.

5 Conclusion

A lane change detection and filter approach is presented in this paper which uses maps, the localized road and the driving trajectory to detect not only lane changes on the straight roads, but also on the curves. Experiments validate that lane changes can be detected using maps, rear wheel encoders and IMU. And the lane change detection and filtering approach can improve longitudinal position precision using

equivalent trajectory and help terrain localization by eliminate the error of road slope estimation.

Future work will focus on lane change prediction by matching the driving trajectory with maps, and the efficient integration with network map-matching algorithm.

Acknowledgements This work was supported by the National Natural Science Foundation of China (91420101), International Chair on automated driving of ground vehicle, and National Magnetic Confinement Fusion Science Program (2012GB102002).

References

1. Rose, C., Britt, J., Allen, J., Bevly, D.: An integrated vehicle navigation system utilizing lane-detection and lateral position estimation systems in difficult environments for gps. *IEEE Trans. Intell. Transp. Syst.* **15**(6), 2615–2629 (2014)
2. Dean, A.J.: Terrain-based road vehicle localization using attitude measurements. ProQuest (2008)
3. Li, T., Yang, M., Deng, L., He, Y., Wang, C.: An extended probabilistic self-localization algorithm using hybrid maps. In: 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), IEEE, pp. 81–86 (2014)
4. Dean, A.J., Vemulapalli, P.K., Brennan, S.N.: Highway evaluation of terrain-aided localization using particle filters. In: ASME 2008 Dynamic Systems and Control Conference, American Society of Mechanical Engineers, pp. 1139–1145 (2008)
5. Salvucci, D.D.: Inferring driver intent: A case study in lane-change detection. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, SAGE Publications, Vol. 48, pp. 2228–2231 (2004)
6. Salvucci, D.D., Mandalia, H.M., Kuge, N., Yamamura, T.: Lane-change detection using a computational driver model. *Human Factors: J. Human Factors Ergonomics Soc.* **49**(3), 532–542 (2007)
7. Mandalia, H.M., Salvucci, M.D.D.: Using support vector machines for lane-change detection. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting. SAGE Publications, Vol. 49, pp. 1965–1969 (2005)
8. Kasper, D., Weidl, G., Dang, T., Breuel, G., Tamke, A., Wedel, A., Rosenstiel, W.: Object-oriented bayesian networks for detection of lane change maneuvers. *IEEE Intell. Transp. Syst. Mag.* **4**(3), 19–31 (2012)
9. Schleichriemen, J., Wedel, A., Hillenbrand, J., Breuel, G., Kuhnert, K.D.: A lane change detection approach using feature ranking with maximized predictive power. In: Intelligent Vehicles Symposium Proceedings, 2014 IEEE, IEEE, pp. 108–114 (2014)
10. Liu, W., Wen, X., Duan, B., Yuan, H., Wang, N.: Rear vehicle detection and tracking for lane change assist. In: Intelligent Vehicles Symposium, 2007 IEEE, IEEE, pp. 252–257 (2007)
11. Yao, W., Zhao, H., Bonnifait, P., Zha, H.: Lane change trajectory prediction by using recorded human driving data. In: Intelligent Vehicles Symposium (IV), 2013 IEEE, IEEE, pp. 430–436 (2013)
12. Brubaker, M.A., Geiger, A., Urtasun, R.: Lost! leveraging the crowd for probabilistic visual self-localization. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, IEEE, pp. 3057–3064 (2013)

Part XV
Robot Vision

Synchronous Dataflow and Visual Programming for Prototyping Robotic Algorithms

Sebastian Buck, Richard Hanten, C. Robert Pech and Andreas Zell

Abstract Robots perceive their environment by processing continuous streams of data, which can be very naturally modelled as a dataflow graph. The development of new perception algorithms is often an iterative process, involving the investigation of a set of parameters and their influence on the system. The amount of immediate feedback available to the developer can make these influences more obvious and can therefore speed up development. We present a framework based on synchronous dataflow and event-based message passing that forms the basis of a visual programming language for rapid prototyping of robotic perception systems. We explicitly model algorithmic parameters in the dataflow graph, which results in a more expressive feature set. We provide an open-source implementation, consisting of a user interface for immediate feedback and interactive manipulation of dataflow algorithms and an independent execution framework that can be directly used on any robot.

Keywords Perception · Prototyping · Visual programming · Dataflow · Robotics

1 Introduction

Robots interact with their environment and therefore need to solve a variety of perception tasks. Information has to be extracted from the raw data recorded by the robot's sensors, before any action can be performed. These tasks, which can be

S. Buck (✉) · R. Hanten · C.R. Pech · A. Zell
University of Tübingen, Sand 1, 72076 Tübingen, Germany
e-mail: sebastian.buck@uni-tuebingen.de

R. Hanten
e-mail: richard.hanten@uni-tuebingen.de

C.R. Pech
e-mail: robert.pech@uni-tuebingen.de

A. Zell
e-mail: andreas.zell@uni-tuebingen.de

© Springer International Publishing AG 2017
W. Chen et al. (eds.), *Intelligent Autonomous Systems 14*,
Advances in Intelligent Systems and Computing 531,
DOI 10.1007/978-3-319-48036-7_66

collectively seen as a part of a robot's cognitive functions, interpret streams of data for an indefinite amount of time.

Designing and implementing perception algorithms for a robotic system often begins with prototyping and experimentation, followed by iterative refinement. This is especially the case in research and education, where existing functionalities should be reused and improved. Reusability is an important characteristic of a well designed system. To create a reusable piece of software, it is important to maximize modularity, such that modules can be shared by different projects.

Many modern robotic systems and frameworks, such as ROS [1], are implemented using message passing. By only depending on the types of messages coming in and going out, this approach results in a very low coupling between modules, which simplifies team work. The ROS graph can be viewed as *dataflow* programming, where information flows from sources, like camera drivers, through various other processes into sinks, for example the motor drivers. Different processes in such a system can read and produce messages at different frequencies and the overall *asynchronous* dataflow is thus hard to describe and control.

Synchronous data flow (SDF), on the other hand, models the flow of data explicitly and is widely used in signal processing. In SDF it is known a priori how many messages a node will consume and produce in each iteration. Processes that read one message on each incoming channel and produce one message on each outgoing channel are called homogeneous and allow an efficient execution without needing message queues, as shown by Lee and Messerschmitt in [2].

The asynchronous approach is well suited for modelling an entire robotic system, since different subsystems have different requirements: Low-level control has to be executed at very high frequencies, whereas high-level decision making might only be necessary relatively infrequently. Perception is a subsystem that is situated in between the two extremes, preferably running at frequencies equal to the data acquisition rate. The sequential nature of most perception algorithms is ideally implemented using SDF and derivations thereof. There already exist many approaches to model perception as a pipeline of processes, to which we will compare our approach in Sect. 5.

The main idea of this paper is to define a pragmatic dataflow model that can be directly used for prototyping perception algorithms with visual programming:

- We develop an implementation of coarse-grain synchronous data flow models without message queuing. Our model is based on SDF and is tailored to allow interactive dataflow graph manipulation. We explicitly model computation parameters in the data flow and introduce aspects of control flow. We show how the scheduling of such graphs can be controlled by the user (Sect. 2).
- We provide an open-source implementation of the presented dataflow framework, called the *Cognitive Systems Algorithm Prototyper and EXperimenter* (CS::APEX), which consists of a graphical user interface and an execution back end. The user interface allows direct interaction with the structure of the dataflow graph and introspection into the flow of data (Sect. 3).

- We share the lessons we learned using the framework in research projects, educational courses and robot competitions. Additionally we show how the introduction of control flow aspects makes this model more expressive for more complex applications than pure dataflow models (Sect. 4).

2 Graphical Model for Coarse-Grained Dataflow

The idea of this section is to give a useful and expressive model for the implementation of a dataflow-based visual programming framework with the following requirements:

- Dataflow graphs can be created and modified at run-time.
- Parameter values can be changed through the dataflow and by the user.
- Data in the flow can be inspected at any step in the process.
- The user can influence the scheduling policy of the processing nodes.
- Irregular events are made visible to the user and can be handled coherently.

2.1 Homogeneous Synchronous Data Flow

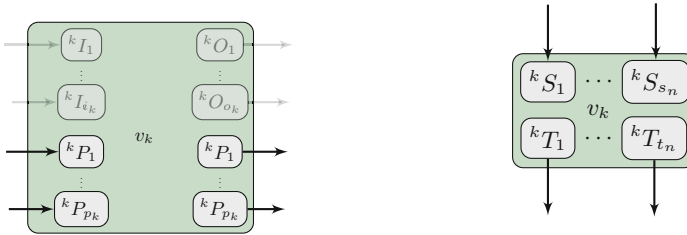
Flow-based programming has been extensively investigated in literature. Our approach is derived from *homogeneous synchronous data flow* (SDF) [2], which is widely used for signal processing pipelines.

At first we will look at a general dataflow. Let $G = (V, E)$ be a directed graph of processing nodes $v_k \in V$ and message connections E . For each process v_k we define a set of inputs \mathcal{I}_k and outputs \mathcal{O}_k

$$\mathcal{I}_k = \{ {}^k I_1, \dots, {}^k I_{i_k} \}, \mathcal{O}_k = \{ {}^k O_1, \dots, {}^k O_{o_k} \},$$

as visualized in Fig. 1a. We add an edge $({}^k O, {}^l I)$ to E if an output ${}^k O$ of node v_k is sending messages to an input ${}^l I$ of node v_l . Inputs and outputs are typed and can be connected if their types are *compatible*. An output can be connected to arbitrarily many inputs, but inputs can only be connected to one output. We allow for an input to be *optional*, which means that it is ignored, if it is not connected to an output. It is treated as a normal input, otherwise.

When the process v_k is executed, it will read the message from each $I \in \mathcal{I}_k$ and write a message to some of the $O \in \mathcal{O}_k$. After the execution of v_k , the messages for $O \in \mathcal{O}_k$ will be forwarded to all the connected inputs. If there are no messages to be sent, we propagate a special `Nothing` token instead.



(a) Inputs and outputs of node v_k . Parameters ${}^k P_i$ are both inputs and outputs, where messages are automatically forwarded. (b) Triggers and Slots on a dual graph structure. Signals sent from Triggers represent *events* and do not behave like flowing data.

Fig. 1 A node consists of inputs and outputs (*left*), as well as slots and triggers (*right*). These entities implement data-driven and control-driven flow respectively

2.2 Parameters

For each computational node $v_k \in V$ we define a set of parameters \mathcal{P}_k , which are treated both as inputs, as well as outputs of v_k by adding additional inputs $\mathcal{I}_k^P \subset \mathcal{I}_k$ and outputs $\mathcal{O}_k^P \subset \mathcal{O}_k$ (Fig. 1a). Parameters are declared for each node type and control the internal processing of each instance of that node type. A node v_k can read its parameters at any time and is allowed to change them.

A parameter’s value can also be changed by other means: An incoming message at a parameter input port causes the value of that parameter to be updated. At every firing of v_k , all the parameters’ values are sent as messages on the corresponding output ports. Both mechanisms together allow values of different parameters to be synchronized without their nodes knowing about each other.

Parameters behave the same as regular input or output ports and can be connected to any other port, making the parameter accessible to the network. Values computed by a node can be manipulated using further processing nodes and then assigned to another node’s parameter, for example. At the same time, this explicit modelling enables a user interface to present control panels to adjust the parameter values, giving the user a more direct control over the data flow.

2.3 Event-Based Message Passing

Pure dataflow is ideal for processing indefinite streams of information. A robot’s perception can be modelled using multiple subsystems that are based on dataflow. There are, however, stimuli the system has to respond to, which are more irregular and often not predictable. These can be both triggered by external means or detected within the data stream. We call these stimuli *events* and introduce means to handle them in a coherent framework with the dataflow itself.

To realize such non-dataflow communication between nodes in G , we define sets S_k and \mathcal{T}_k representing *Slots* and *Triggers* of node v_k analogously to I_k and O_k (see Fig. 1b). Triggers can be used to signal *events* to another node by connecting them to slots, contributing an edge $({}^i T, {}^j S)$ to the edges E . Triggers can only be connected to slots and outputs only to inputs. Every node is therefore a composition of inputs, outputs, signals and slots, as visualized in Fig. 1.

In contrast to the dataflow, events are more irregular and should be handled asynchronously once they are triggered, so that not all slots have to receive a message at the same time. This means that slots can be connected to multiple triggers and vice versa. By not using the dataflow to send events between nodes, we avoid sending special marker messages. Disjoint dataflow subgraphs can run at different frequencies but can still communicate via events. A useful application of events is to control aspects of the execution and to send commands between nodes independently of the dataflow, for example disabling currently unneeded nodes, or resetting internal state of more complex, stateful nodes.

2.4 Scheduling

Scheduling the execution of G requires a policy to decide, when to execute the different nodes. Node v_k becomes *enabled* once each of its inputs has received a message and each of its outputs can send a new message. Nodes without inputs, also called *sources*, are enabled whenever their outputs can send messages. *Sinks*, that is nodes without outputs, are enabled when all inputs have received a message. Enabled nodes can be executed whenever processing resources are available. Once executed, v_k reads messages from the inputs, processes them and (possibly) generates outputs.

An output can send a new message once all previously sent messages have been read down-stream. Messages are read before they are processed, so that earlier nodes in the network can already be executed again, even if their previously generated messages are still being processed. This allows the scheduler to perform pipeline execution of sequentially connected nodes.

There can be multiple enabled nodes at any time, which allows concurrent execution. The order in which these nodes are executed does not matter, they can run in a sequence or fully in parallel. To allow dynamic modification of the dataflow graph at run-time, a static scheduling scheme, as originally developed for SDF graphs [2], cannot be employed.

Events are also managed by the scheduler: If a slot has received an event, the node will be required to handle the event as soon as possible. The scheduler can execute event handling routines at any time, if the node is not currently being executed. If a node is enabled and also has pending events to handle, the scheduling policy can decide which will happen first.

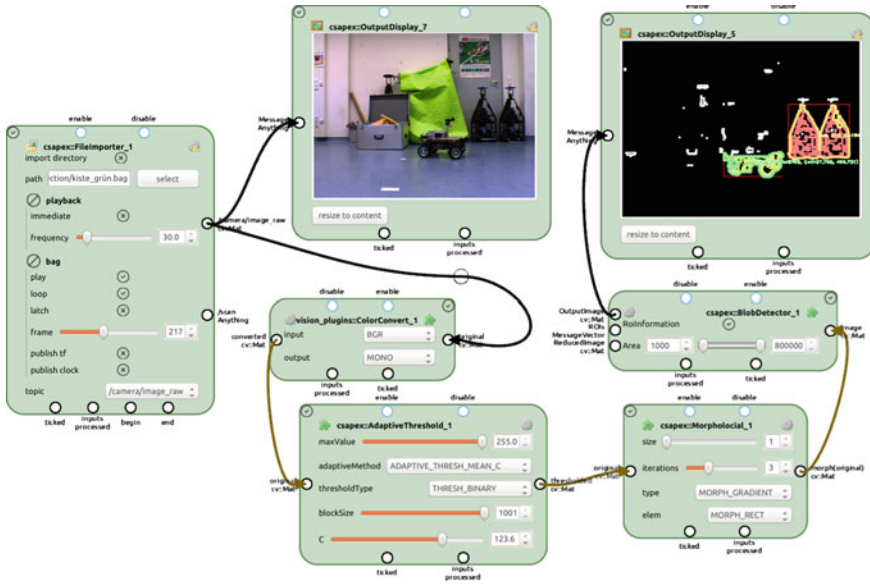


Fig. 2 Exemplary workflow that imports a ROS bag, converts the camera image to gray values, performs adaptive thresholding, morphological operations and blob detection

3 Implementation in CS::APEX

The aim of CS::APEX¹ is to be a user-centred platform for developing and experimenting with flow-based algorithms for robots and other cognitive systems, encouraging modularity, extensibility and accessibility. We follow a pragmatic approach to dataflow programming, focused on providing a user-friendly interface, concentrating on speeding up the prototyping experience, providing useful user feedback and making parameters of the system more easily accessible. Resulting dataflow networks can be directly deployed on a robot (Sect. 4).

The framework consists of two components: A graphical user interface (see Fig. 2) based on Qt5 and a computation back end library for scheduling and maintenance.² We achieve modularity by implementing the flow-based graph structure presented in Sect. 2, encouraging users to implement component-based solutions that only depend on message types and can thus be easily reused. Extensibility is accomplished by a plug-in system, which makes modification of the main components unnecessary and simplifies the distribution of implemented computing nodes among collaborators.

¹CS::APEX and its documentation are available for download at <http://www.ra.cs.uni-tuebingen.de/software/apex/>.

²An overview video can be seen at <http://youtu.be/weFZZrQ1BeE>.

The user interface allows the user to dynamically add and delete computation nodes at run-time. Nodes can also be disabled and enabled, moved and copied. Furthermore, the user can add and delete connections between nodes and inspect the transmitted values in these connections. No scripting or manual configuration file editing is required. The user interface is used to generate a network and to provide feedback during the prototyping process. Once the configuration is done, the UI is no longer needed and the graph can be executed in a headless fashion. In this way, a prototype configuration can be used on a robot, without a screen attached.

Adding custom functionality is possible by implementing new node types and providing parameters to allow fine tuning. Computation nodes are written in C++11 and dynamically linked once they are needed. We provide multiple ways to add new processing nodes: Nodes can be derived from a base class `Node`, or from specialized base classes, like image filters. Furthermore, we provide a utility class that can automatically generate nodes from a given C++ function by analyzing the function signature using template meta programming techniques.

Packages can also provide custom message types. Other plug-ins can use these messages without having to worry how to do I/O with them. Authors of a custom message can provide functions to serialize the message using YAML, to read them from a file, to visualize them and publish them via ROS. We have implemented messages for integral types, strings, images, laser scans and more. We have also implemented support for OpenCV³ and the point cloud library (PCL)⁴ via independent libraries.

Parameters are central to our approach, since the direct feedback from changing parameters can speed up rapid prototyping. As described in Sect. 2, we create a pair of one input and one output per parameter. Since this would lead to a lot of ports for nodes with many parameters, we only add inputs and outputs for parameters that are marked as *interactive* by the user.

We provide different types of parameters: Boolean, integral and floating point values, ranges, intervals, pairs and more specialized variants like angles, file paths and color values. To enable fast changes to parameter values, we wrap each parameter into a specialized UI widget that allows the user to quickly modify the parameter's value. UI elements and the parameters themselves are strictly separated, so that a graph can be used on a robot completely without a graphical interface. We render these controls directly into the graphical representation of the node, as can be seen in Fig. 3. This way, the controls are physically located where they are used, which makes it easy to find the right parameter to change and allows us to provide helpful information for parameters via tool-tips.

Executing a node does not have any side effects on other nodes in the graph. For this reason, concurrency can be achieved without any effort on the client side, merely the scheduler has to deal with the details of concurrent programming. This takes away the burden on inexperienced programmers, who can make full use of par-

³OpenCV is available at <http://www.opencv.org/>.

⁴PCL is available at <http://www.pointclouds.org/>.

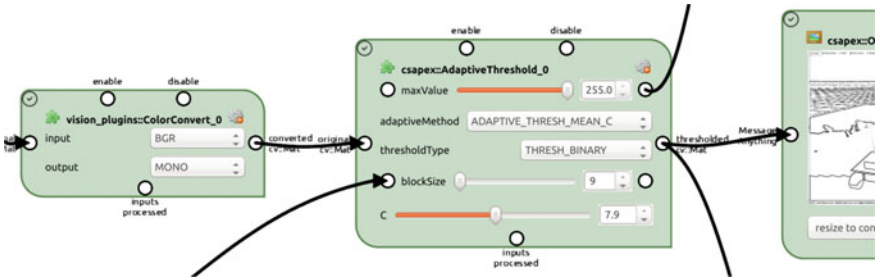


Fig. 3 A node that performs adaptive thresholding on an image with one image input and one output. There are five parameters: two floating point ranges (maxValue, C), one integer range (blockSize) and two sets (adaptiveMethod, thresholdType). maxValue and blockSize are connected to the data flow. The node has automatically generated slots to enable and one to disable it and a trigger that fires once the input is processed

allel architectures with thread-agnostic modules. Parallel execution is not necessary though, all nodes can be executed sequentially as well.

Even though users should not have to be concerned about matters of concurrency, we want to enable them to take control of the scheduling process. For this reason, we employ a distributed scheduling scheme using thread pools. Every node is assigned to a thread group and all the nodes in a thread group are managed by one scheduler. By default, every node is managed by a single scheduler, as to not fully utilize every CPU core, if that is not desired. The interface allows users to define their own thread groups and assign nodes to them.

4 Applications

Since the development of CS::APEX began in 2012, we have developed more than 300 plug-ins to solve a variety of perception problems for research projects and robotics competitions: We achieved the second place in the SICK Robot Day 2014 [3], which was an object-delivery competition. We also deployed the framework for the perceptions tasks at the SpaceBot Camp 2015,⁵ which was hosted by the national aeronautics and space research centre of Germany (DLR).

Additionally, we are using the framework in research projects: We developed a person-recognising autonomous transportation system in the BMBF-funded project PATSY,⁶ using dataflow graphs to detect obstacles and people in point clouds. In the project IZST IOC 104,⁷ founded by the state of Baden-Württemberg, we developed vision algorithms for laparoscopic surgery.

⁵http://www.dlr.de/dlr/desktopdefault.aspx/tabid-10081/151_read-15747.

⁶<http://www.ra.cs.uni-tuebingen.de/forschung/patsy/>.

⁷http://www.ra.cs.uni-tuebingen.de/forschung/Chirurgische_Navigation.

In the remainder of this section we present two use cases in more detail.

4.1 SICK Robot Day 2014

The SICK Robot Day 2014 [3] was an object-delivery competition, in which robots had to autonomously detect and approach filling stations where they received labelled objects. The objects then had to be delivered to delivery stations determined by the object label. Delivery stations were marked with large number signs, as shown in Fig. 3a. We split the detection task into four separate dataflow graphs: Cross-hair detection, number sign detection, bar code reading, and environment map analysis. Here we focus on the specific task of number sign detection and demonstrate the benefits of using our framework.

Many necessary processing steps had already been implemented as nodes before and could be reused without modification: First we had to rotate the image, because the camera was mounted on its side. Then we performed color conversion from YUV to BGR. Next we applied an adaptive thresholding step (see Fig. 3) to separate light from dark areas. After that we performed connected component analysis to find dark image areas.

The only missing component is defined by the following interface: A node that takes a vector of image components, classifies the components and outputs the classified result as vector of regions of interest. We implemented a plug-in to communicate with the Java-based neural network framework JANNLab developed by Otte et al. [4], using an MLP to perform the classification. By keeping a generic message passing interface, the resulting node is easily reusable. For training of the neural network we were able to reuse already existing nodes.

The interfaces between the different nodes had to be changed multiple times, so that the initial interface specification was soon outdated. Being able to clearly see what kind of data a node consumes and produces, these changes had nearly no impact on productivity and could be accounted for with minimal or no changes to other nodes. We also benefited from fast parameter tuning and debugging due to the immediate visual feedback. This was especially helpful during the final moments of preparation for the competition.

We were able to use the resulting graphs on our robot without modification. Running the framework without a user interface entailed minimal overhead compared to a hand crafted program, since feedback and debug information (as seen in Fig. 4b) are not computed without a UI. Experiments show an average of $k \times 1.2$ ms overhead, where k is the longest distance from a source to a sink.



(a) The robot's camera image, showing an occluded bull's eye target and the number sign of station 1. (b) Overlay of the detections of two dataflow graphs, one detecting numbers and one detecting bull's eye targets.

Fig. 4 SICK robot day 2014: Sensor input and visual feedback generated

4.2 Evolutionary Optimization of a ROS Node

The second use case we want to share is the application of evolutionary algorithms to optimize the parameters of a ROS node called `laser_scan_matcher` using the *Differential Evolution* algorithm. Using a plug-in to communicate with the optimization framework Eva2 developed by Kronfeld et al. [5], we were able to find parameters that minimized the trajectory error on our dataset.

The scan matching process was running in a separate ROS node, whose parameters were set using a generic ROS parameter interface. The values of these parameters were determined by the optimization framework for each iteration and were propagated through the dataflow. A trigger was executed every time a run of the scan matcher was complete. This trigger was connected to a slot in the optimizer, which caused a new parameter set to be generated.

The communication with the optimization framework is generic and can be applied to other problems. The user selects an arbitrary set of parameters using the user interface, which adds it to the optimization process. This is possible due to the generic integration of the parameters into the dataflow and can be implemented purely on a plug-in basis, since it does not require any modifications to the framework. The only additional functionality, that has to be implemented, is a node to calculate a fitness value for the current parameter set. This value is needed by the optimizer to assess different parameter combinations.

Furthermore, this example demonstrates the usefulness of the additional event-based mechanism. Events can be used to control the execution of different dataflow subgraphs. This way, processes like these can be automated and controlled based on the dataflow. To the best of our knowledge, such an approach to parameter optimization is not possible in related frameworks.

5 Comparison to Related Work

Many tools and frameworks utilizing flow-based programming have been published in related domains, such as Ptolemy II by Eker et al. [6] and the Ptolemy-based Kepler by Ludätscher et al. [7]. Special purpose frameworks include the Robot Task Commander by Hart et al. [8], the Konstanz Information Miner (Knime) [9] for data mining, the Waikato Environment for Knowledge Analysis (WEKA [10]) and Orange [11] for machine learning and MeVisLab [12] for medical image processing. There are also commercial products based on data flow processing, for example LabVIEW and MATLAB Simulink.

Biggs et al. [13] provide a pipeline based approach, and show its potential with an example for point cloud processing. They impose requirements similar to the ones presented in Sect. 2, yet focus more on inter-node communication aspects and less on interaction. Their framework implements asynchronous dataflow, meaning that there is a need for message queues between components. They provide a user interface with which the graph can be modified at run-time and parameters can be adjusted, but they do not model parameters in the dataflow and don't seem to feature event-based functionality in their user interface.

Although our implementation can be used independently from ROS [1], we provide our user interface as a ROS node, in which ROS topics can be subscribed and published to. ROS itself can be seen as a flow-based framework, where different nodes are separate processes and can run on different machines in a local network. ROS is implemented using the publish-subscribe pattern, where each node is publishing messages onto topics that are subscribed to by other nodes, which is another case of asynchronous dataflow. Our visual programming approach allows users to construct processing graphs on the fly via a graphical user interface instead of using text based configuration files.

More relevant for our work is ecto [14], which grew out of the ROS scene and also represents computer vision and perception tasks as a directed acyclic graph. In contrast to ecto, our approach explicitly handles node parameters, allowing them to be used as data sources or sinks. Graphs in ecto are meant to be specified and configured using python programs. There exists a web-based graphical user interface for ecto which allows users to create nodes and connect them. This approach has the advantage that the graph is accessible via the network, yet the implementation does not allow for a high level of interactivity. In our approach, the user interface is the key part of the framework and we focus on interactive graph manipulation and immediate feedback.

6 Discussion and Conclusions

We describe a graphical model based on synchronous dataflow and event-based message passing, as well as an implementation of this model in the form of a visual programming language framework called CS::APEX. Extensions to the well known

SDF model are motivated by the application to prototyping perception algorithms in a scientific setting: We model program parameters directly in the dataflow, which we can use to perform automatic parameter optimization. Furthermore, we use signal-based mechanisms to model irregular events and allow users to manipulate both data flow and event handling in a coherent interface.

In other fields, graphical prototyping tools have become commonplace, yet in robotics there do not exist such standard tools. We think this is partly due to the fact, that robotics is a broad and multidisciplinary field. We aim to provide a user-friendly graphical interface that lowers the barrier to entry into robotics, especially robotic perception. The interface provides immediate feedback, allowing to visually construct new dataflow graphs at run-time, to get insight into the dataflow in real-time and to learn the effects of different parameters on the behaviour of the algorithm. Although we provide a library of reusable nodes, custom algorithms have to be implemented eventually. In contrast to fine-grained visual programming, where individual instructions are composed in a graphical interface, we rely on a plug-in based system which allows users to program custom processing nodes in C++ and then compose them visually. This approach simplifies using unknown modules, since inputs, outputs and parameters are directly displayed and can be connected visually.

On its own, homogeneous synchronous data flow seems to be a limiting factor, due to a uniform execution of the individual nodes, whereas robotic systems are composed of many subsystems requiring different update rates. Our approach, however, is meant to implement individual subsystems, such as perception modules. Many perception problems naturally show synchronous characteristics: If an online image classifying algorithm, for example, cannot operate in real-time, images have to be dropped, synchronizing the update rate of the pipeline.

Our framework can be fully applied to problems that can be separated into modules. Highly optimized algorithms that cannot be subdivided, however, have to be implemented as single nodes. This is not unusual in a coarse-grain data flow framework such as the presented approach. Even though large nodes are less reusable, they can still profit from the parameter system and other UI features such as execution profiling and data visualization.

Acknowledgements This work is funded by the German Federal Ministry of Education and Research (BMBF Grant 01IM12005B). The authors would like to thank Sebastian Otte and Fabian Becker for providing their implementations of artificial neural networks and evolutionary optimization algorithms, as well as the students, who are using CS::APEX in their research, for providing constructive feedback.

References

1. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y.: Ros: an open-source robot operating system. In: ICRA workshop on open source software. Vol. 3, p. 5 (2009)

2. Lee, E., Messerschmitt, D.G., et al.: Synchronous data flow. *Proc. IEEE* **75**(9), 1235–1245 (1987)
3. Buck, S., Hanten, R., Huskić, G., Rauscher, G., Kloss, A., Leininger, J., Ruff, E., Widmaier, F., Zell, A.: Conclusions from an object-delivery robotic competition: sick robot day 2014. In: *The 17th International Conference on Advanced Robotics (ICAR)*, Istanbul, TR, pp. 137–143 (2015)
4. Otte, S., Krechel, D., Liwicki, M.: Jannlab neural network framework for java. In: Perner, P. (ed.) *MLDM Posters*, pp. 39–46. IBAI Publishing (2013)
5. Kronfeld, M., Planatscher, H., Zell, A.: The eva2 optimization framework. In: *Learning and Intelligent Optimization*, pp. 247–250. Springer (2010)
6. Eker, J., Janneck, J.W., Lee, E.A., Liu, J., Liu, X., Ludvig, J., Neuendorffer, S., Sachs, S., Xiong, Y.: Taming heterogeneity-the ptolemy approach. *Proc. IEEE* **91**(1), 127–144 (2003)
7. Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E.A., Tao, J., Zhao, Y.: Scientific workflow management and the kepler system. *Concurrency Comput.: Practice Exp.* **18**(10), 1039–1065 (2006)
8. Hart, S., Dinh, P., Yamokoski, J., Wightman, B., Radford, N.: Robot task commander: A framework and ide for robot application development. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pp. 1547–1554 (2014)
9. Berthold, M.R., Cebon, N., Dill, F., Gabriel, T.R., Kötter, T., Meinel, T., Ohl, P., Sieb, C., Thiel, K., Wiswedel, B.: KNIME: the Konstanz Information Miner. In: *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. Springer (2007)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. *SIGKDD Explor. Newsl.* **11**(1), 10–18 (2009)
11. Demšar, J., Curk, T., Erjavec, A.: Črt Gorup, Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., Štajdohar, M., Umek, L., Žagar, L., Žbontar, J., Žitnik, M., Zupan, B.: Orange: data mining toolbox in python. *J. Mach. Learn. Res.* **14**, 2349–2353 (2013)
12. Bitter, I., Van Uitert, R., Wolf, I., Ibanez, L., Kuhnigk, J.M.: Comparison of four freely available frameworks for image processing and visualization that use itk. *IEEE Trans. Visual. Comput. Graphics* **13**(3), 483–493 (2007)
13. Biggs, G., Ando, N., Kotoku, T.: Rapid data processing pipeline development using openrtm-aist. In: *2011 IEEE/SICE International Symposium on System Integration (SII)*. pp. 312–317 (2011)
14. Ethan Rublee, V.R., et al.: Ecto - A C++/Python Computation Graph Framework (2015)

Depth-Based Frontal View Generation for Pose Invariant Face Recognition with Consumer RGB-D Sensors

Giorgia Pitteri, Matteo Munaro and Emanuele Menegatti

Abstract In this work, we propose to exploit depth information to build a pose-invariant face recognition algorithm from RGB-D data. Our approach first estimates the head pose and then generates a frontal view for those faces that are rotated with respect to the frame of the camera. Then, some interest points of the face are detected by means of a Random Forest applied to the RGB image and they are used as key-points where to compute feature descriptors. Around these points and their 3D counterpart, we extract both 2D and 3D local descriptors, which are then concatenated and classified by means of a Support Vector Machine trained in “one-versus-all” fashion. In order to validate the accuracy of the system with data from consumer RGB-D sensors, we created the *IAS-Lab RGB-D Face Dataset*, a new public dataset in which RGB-D data are acquired with a second generation Microsoft Kinect. The reported experiments show that the depth-aided approach we propose allows to improve the recognition rate up to 50 %.

Keywords 3D face recognition • Pose-invariance • RGB-D • Frontalization

1 Introduction

Face recognition is an important application in video surveillance, mobile robotics, ambient assisted living and human-robot interaction. The problem of identifying or verifying a face in a video sequence or in an image based on a face database is

G. Pitteri (✉) · M. Munaro (✉) · E. Menegatti
Department of Information Engineering, University of Padova,
Via Gradenigo 6/A, 35131 Padova, Italy
e-mail: giorgia.pitteri@dei.unipd.it
URL: <http://robotics.dei.unipd.it>

M. Munaro
e-mail: matteo.munaro@dei.unipd.it

E. Menegatti
e-mail: emg@dei.unipd.it

usually solved by following three steps: face detection, feature extraction from face regions and face recognition. In many cases, a pre-processing phase is needed before feature extraction, e.g. to normalize the illumination, and a dimensionality reduction algorithm is applied on the extracted features, e.g. by means of Principal Component Analysis. There are several works in literature concerning face recognition, but most of them only deal with 2D data [1, 2]. However, 2D-based methods could suffer from problems with respect to different light conditions, pose variations and facial expressions.

To cope with pose variations, face frontalization techniques have been recently developed that work on 2D data [3]. These methods are able to estimate the face pose and perform a morphing to a frontal view. However, when the face is highly rotated, the morphing effect is not very robust and 3D information could probably help on this. The influence of precise 3D information provided by professional scanners has been recently studied [4], but it is unclear if low cost and less accurate RGB-D sensors, such as the Microsoft Kinect, can still provide an advantage over 2D face recognition.

In this work, we propose a face recognition algorithm that exploits a depth-based frontalization technique to make face recognition more invariant to pose. The generation of a frontal view is done by means of a pose estimation method that works on 3D data and a face mirroring technique that fills the part of the face with missing data. This procedure helps the subsequent step of facial feature detection. Another contribution of this paper is the release of a new face recognition dataset,¹ that has been collected with the new consumer RGB-D sensor from Microsoft, the Kinect v2.

The remainder of the paper is organized as follows: Sect. 2 reviews the state of the art of face recognition techniques; Sect. 3.1 gives an overview of our pipeline for face recognition, while Sect. 3.2 explains how we obtain a frontal view to achieve pose invariance and Sect. 3.3 shows the mirroring technique needed to fill the missing data caused by face frontalization. In Sect. 4, we report experiments on the new dataset we released and we show results in terms of faces properly recognized. Finally, Sect. 5 draws the conclusions of this work.

2 Related Work

A wide set of works in literature rely on 2D images alone, such as *Eigenfaces*, a popular 2D face recognition algorithm based on Principal Component Analysis (PCA) [5], or the similar *Fisherfaces* algorithm based on Linear Discriminant Analysis (LDA). Currently, descriptor-based approaches [6, 7] have been proven to be effective in face representation. LBP [8], SIFT [9], SURF [10] and POEM [11] are some

¹IAS-Lab RGB-D Face dataset: <http://robotics.dei.unipd.it/reid>.

effective descriptors used for solving the face recognition problem. However, as we mentioned before, when dealing with complex surveillance or service robotics scenarios, such as people not in frontal pose or in bad light conditions, two dimensional information is not always sufficient to obtain reliable results of face detection and recognition. 2D face frontalization methods have been proposed for correcting limited face rotation [3, 12] and accurate 3D information from laser scanners has been exploited to increase recognition accuracy by combining 2D and 3D descriptors [4, 13]. The advent of active and low cost 3D sensors, such as Microsoft Kinect, significantly increased the interest in using 3D information to overcome the limitations of 2D methods. An important goal has been achieved in [14], where a pose estimation algorithm that exploits depth information acquired with low cost sensors is proposed. For the purpose of whole-body person re-identification, [15] overcomes the problem of the different poses people can have by exploiting 3D data acquired by a Microsoft Kinect to transform a person point cloud to a standard pose before applying a 3D matching algorithm. To the best of our knowledge, depth information from consumer RGB-D sensors has not still be used for helping face frontalization techniques and thus improving pose-invariance in face recognition algorithms. This work goes in that direction.

3 Methodology

3.1 System Overview

The framework we have designed allows recognition of a subject standing in front of a RGB-D camera in different poses by taking into account a single test frame. To achieve this goal, our approach (Fig. 1) is divided into four phases:

- (1) preprocessing of point cloud data to generate a frontal view;
- (2) 2D projection of face points onto the image plane and face mirroring;
- (3) extraction of facial keypoints and descriptors;
- (4) descriptor classification.

Exploiting depth information, the algorithm generates frontal views of the subjects by means of a head pose estimation and correction algorithm as it will be explained in Sect. 3.2. 3D frontal data are then projected onto a 2D image plane. A face mirroring technique is then applied to the transformed data. The images are fed as input to a facial keypoint detection algorithm developed in [16] that we integrated in our system. The keypoint positions in the 2D space are mapped to the 3D space in order to extract both 2D and 3D feature descriptors around them. The set of descriptors we adopt to recognize a specific person are:

- Speeded Up Robust Feature (SURF [10]): to detect interest points, the SURF algorithm uses an integer approximation of the determinant of Hessian blob detector, which can be computed with 3 integer operations using a precomputed integral

Fig. 1 System overview. The proposed approach is divided into four phases: (1) preprocessing of point cloud data to generate a frontal view, (2) face points projection onto 2D space and application of a face mirroring technique, (3) extraction of facial keypoints and descriptors and (4) descriptor classification

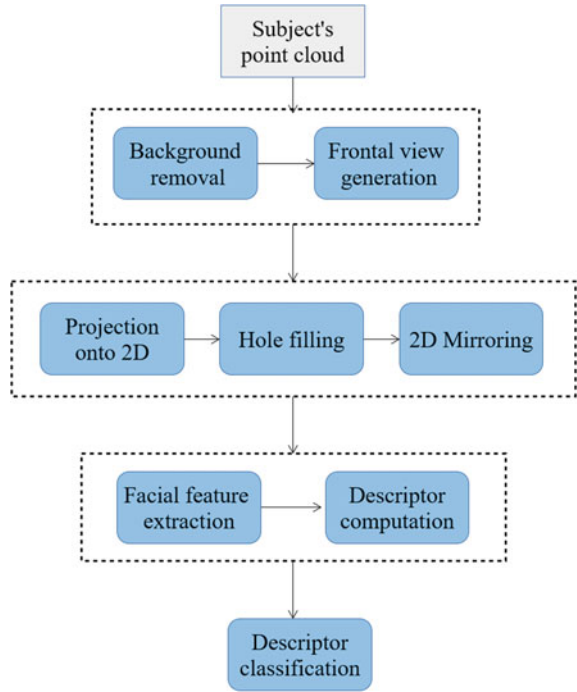


image. Its feature descriptor is based on the sum of the Haar wavelet response around the point of interest that can also be computed with the aid of the integral image;

- Fast Point Feature Histogram (FPFH [17]): a lower complexity version of the PFH descriptor. It is characterized by a histogram of values that represents the relationships between pairs of 3D points belonging to the considered neighborhood;
- Euclidean distances between all pairs of keypoints in the 3D space;

These descriptors are local, that is they extract local information from facial regions and not global features of the entire face. To take the best from all these features, we exploit a fusion technique at the feature extraction level, which consists in concatenating input feature vectors. The choice of the most discriminative features is based upon experiments carried out in cross validation on a training dataset. For classification, we exploit linear Support Vector Machines (SVMs), that are fast at testing time and effective for recognition tasks. SVMs were originally designed for binary classification, but, in this work, we use their extensions to multi-class classification problems, that are called “one-versus-all” and “one-versus-one” SVMs [18]. The “one-versus-all” method trains k SVM models where k is the number of classes to recognize. For each subject, the descriptors of that subject are used as positive examples for its SVM, while the descriptors of all the other subjects are considered as negative examples. At testing time, the k decision functions are used to obtain a

confidence value for the test descriptor from each classifier. The class with the higher confidence value is chosen as the estimated class. The other method, “one-versus-one”, builds $\frac{k(k-1)}{2}$ classifiers where each one is trained on data from only two classes, that provide positive and negative examples. During the testing phase, all these pairwise SVMs are evaluated and a voting strategy is applied in order to choose the best class with respect to two parameters, here listed in descending order of priority: (i) wins, times a class is estimated as membership class, and (ii) confidence max, the maximum value of confidence obtained in the wins.

3.2 Frontal View Generation for Pose Invariance

Accurately localizing the head and all keypoints is a necessary preprocessing step for face recognition. Most of the algorithms in literature that are based on RGB data alone work better with frontal subjects [16], as we can see in Fig. 2. As illustrated in Fig. 3, our approach exploits depth information to transform subjects’ point clouds to a frontal pose and then project them onto 2D, making the algorithm in [16] to work better and our face recognition system to be more pose-invariant. In order to apply the correct transformation, an automatic and robust method for head pose estimation that computes the original pose is needed. In this work, we make use of the head pose and orientation estimation algorithm, developed in [14, 19], that exploits depth data acquired by a consumer depth sensor. The approach is based on discriminative random regression forests: ensembles of random trees trained by splitting each node so as to simultaneously reduce the entropy of the class labels distribution and the variance of the head position and orientation. That is, the forest first discriminates which parts of the image belong to a head, and use only those patches to cast votes

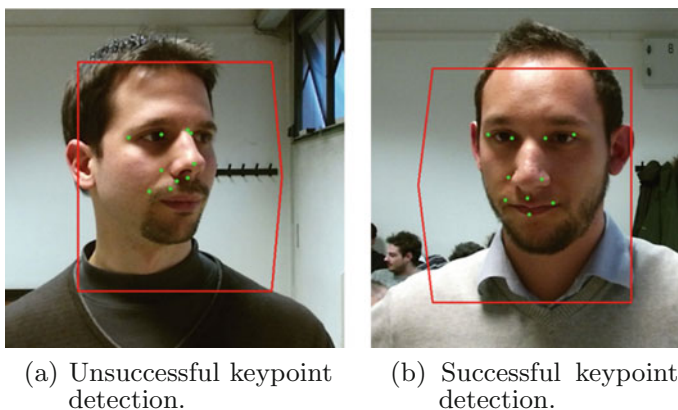


Fig. 2 Difference between the result of the facial feature detection algorithm in [16] for side and frontal poses

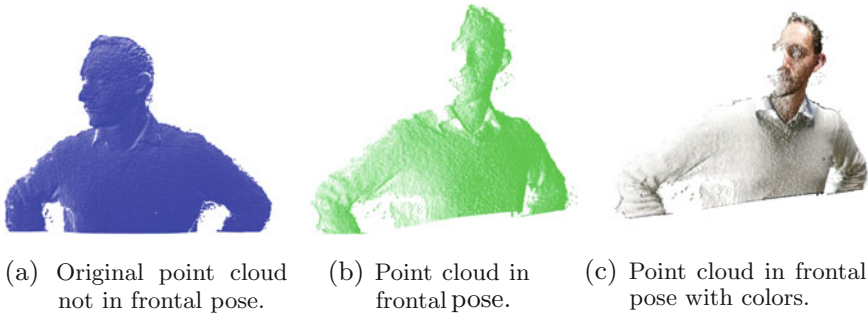


Fig. 3 Frontal view generation process. From the original cloud to the transformed one

for the final estimate. This algorithm computes the center of the head and the orientation expressed in roll, pitch and yaw angles. From these outcomes, we compute the inverse transformation matrix that rototranslates the point cloud of the subject and generates a frontal view of the face. Given the output of [19], that is the face centroid $\mathbf{c} = (c_x, c_y, c_z)$ and the face rotation vector $\mathbf{r} = (\theta_{roll}, \theta_{pitch}, \theta_{yaw})$ with respect to the camera reference frame, we compute the rototranslation matrix \mathbf{RT} to apply to the face point cloud as

$$\mathbf{RT} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad \mathbf{t} = \mathbf{c} - (\mathbf{R}^T \mathbf{c}), \quad (1)$$

where \mathbf{R} is the rotation matrix derived from the rotation vector \mathbf{r} and \mathbf{t} is the translation to apply for keeping fixed the position of the face centroid.

The transformed point clouds are then projected onto the 2D space to get a RGB image of the face in frontal view.

3.3 Face Mirroring for Improving Facial Feature Detection

It is worth noting that the process of rotating each subject's point cloud and then projecting it onto 2D can have two negative effects: the part of the head that was originally occluded to the sensor is now devoid of points and, since point clouds are not completely dense, the obtained RGB images present some holes in correspondence (Fig. 4b). The first problem means that the keypoint detection algorithm would not work well with just half of the face as input data. In this work, we overcome this problem with an easy but effective symmetrization technique called face mirroring, shown in (Fig. 4c). Once found the symmetry axis of the face passing by the head centroid, we fill the missing part of the face with points that are symmetrical to the existing ones with respect to this axis.

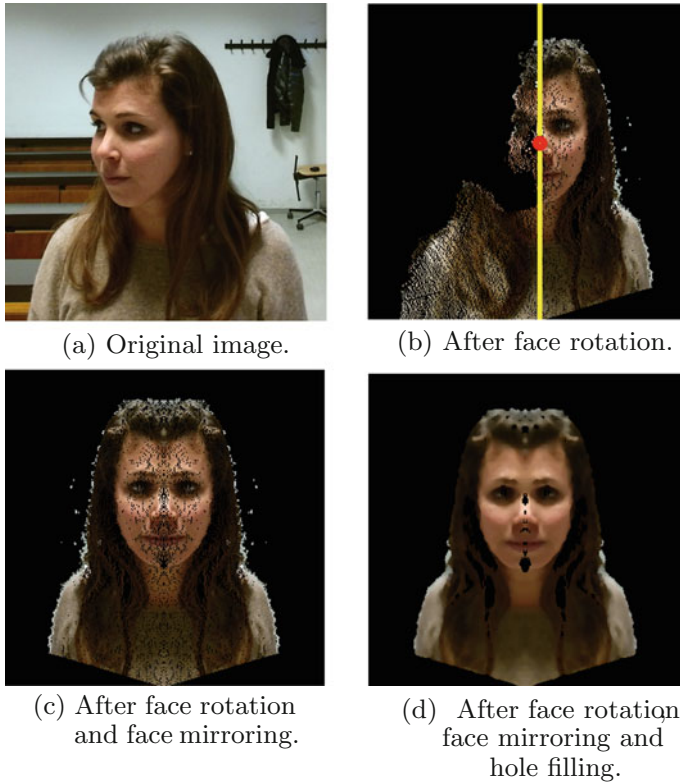


Fig. 4 The main steps of the face frontalization procedure: **a** original image, **b** image after 3D face rotation and symmetry axis, **c** image after 3D face rotation and face mirroring, **d** final result after rotation, mirroring and hole filling

Regarding the problem of holes, we propose to fill them up by computing the depth and RGB median values over a neighborhood around each hole. When computing the median, the points with invalid depth are not considered. Moreover, the dimension of the kernel of the neighborhood chosen for the median calculation iteratively increases until at least a valid point (with valid depth) is found. A result of this process is shown in Fig. 4d. The filling of invalid points in the point cloud is also important for having a better mapping of 2D facial features obtained with [16] to 3D points, that are then considered as keypoints around which computing 3D descriptors.

4 Experiments and Results

4.1 IAS-Lab RGB-D Face Dataset

The new face recognition dataset we collected, the *IAS-Lab RGB-D Face* dataset, has been created to measure accuracy and precision of 2D and 3D face recognition algorithms based on data coming from consumer RGB-D sensors. The training dataset consists of 26 subjects captured in 13 different conditions (with pose, light and expression variations), standing 1 or 2 m from the sensor. In order to represent a typical service robotics scenario, where few people have to be recognized and many others have to be classified as unknown, the testing dataset contains 19 subjects and just four of them were also present in the training dataset. The other testing subjects are thus considered as unknown. The testing set is furthermore divided into five subsets, as explained in Table 1. Samples from the training and the testing datasets are reported in Fig. 5. It can be noticed how this dataset is challenging for mainly two reasons: the limited number and the high variability of examples available for every person. However, this is often the case in service robotics scenarios.

In Fig. 6, we compare our depth-based face frontalization method with the one proposed in [3], that is based on 2D data, on two images extracted from this dataset. In presence of large rotations, as in the second row, the method in [3] creates a double face, that could cause problems in extracting descriptors around the facial features, while our method obtains a very good result.

4.2 Recognition Results

For evaluation purposes, we computed two recognition indices:

- True Positive Rate (TP): percentage of correct recognition over all the classified examples

$$TP\% = \frac{\#correct_matches}{\#test_samples} \cdot 100 \quad (2)$$

Table 1 Testing sets of the *IAS-Lab RGB-D Face* dataset

	Test1	Test2	Test3	Test4	Test5
Frames per subject	2	1	2	1	1
Pose	Frontal	Frontal	Frontal	Not frontal	Frontal
Lighting	Good/quite good	Good	Good	Good	Dark
Distance	1 m	2 m	1 m	1 m	1 m
Expression	Standard	Standard	Smile/serious	Standard	Standard



Fig. 5 Selection of frames from the **a** training and **b** testing sets of the *IAS-Lab RGB-D Face* dataset. The *first four rows* contain frames of the four known persons of the dataset, that are persons present in both the training and the testing dataset, while the *other rows* exemplify the “unknown” class

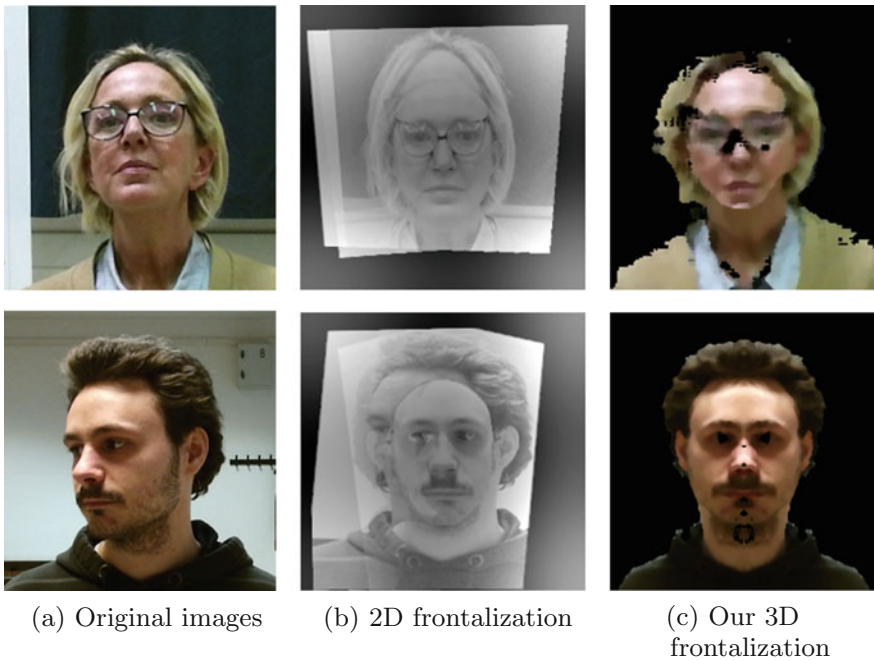


Fig. 6 Comparison between the 2D frontalization method in [3] and the proposed depth-based frontalization. The difference is evident in presence of large face rotations (*second row*)

Table 2 Valid frames (frames for which it was possible to extract features)

	Test1	Test2	Test3	Test4	Test5
Examples loaded	38/38	18/19	38/38	16/19	12/19
Known subjects examples	8/8	4/4	8/8	3/4	3/4
Unknown subjects examples	30/30	14/15	30/30	13/15	9/15

- False Positive Rate (FP): percentage of times an unknown subject is recognized as a known person

$$FP\% = \frac{\#false_positives_unknown}{\#unknown_samples} \cdot 100 \tag{3}$$

It should be considered that the effective number of frames for each subject that can be used for training and testing can be different with respect to the number of frames per subject of the datasets. This is due to the fact that the facial feature detection algorithm sometimes fails, thus the corresponding frame cannot be exploited for testing. Table 2 reports the number of valid frames obtained when applying our complete pipeline. For these frames it was possible to extract features.

The first test we carried out had the purpose of showing that generating the frontal view before extracting the facial features and applying a mirroring technique can improve performance. In Table 3, we report the recognition indices of the system in three cases: (i) using the original images without face frontalization, (ii) using our face frontalization technique and (iii) with both face frontalization and face mirroring. For this test, we extracted three descriptors: SURF from the 2D image and FPFH and 3D distances between facial feature points in the 3D space. We then classified them by means of SVMs trained in “one-versus-all” fashion.

From Table 3, in correspondence of the *Test 4* testing set, that contains non-frontal faces, we can notice that the TP rate is very low on the original images, but it increases when we apply our frontalization and mirroring techniques. In particular, the latter allows to detect facial keypoints more accurately than in the other cases. A second test, whose results are reported in Table 4, compares the three chosen

Table 3 Evaluation results obtained with three different variants of the followed approach: (i) without frontal view generation, (ii) with frontal view generation, (iii) with frontal view generation and mirroring

	Test1		Test2		Test3		Test4		Test5	
	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)
Original images	20	4	30	60	37	46	11	54	30	50
Frontalization	85	23	79	7	87	13	33	0	75	0
Frontalization and mirroring	85	23	79	7	87	13	59	0	75	0

Table 4 Evaluation results obtained by changing the extracted features. We tested each type of descriptor alone and also their concatenation

	Test1		Test2		Test3		Test4		Test5	
	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)
SURF	56	20	79	7	56	17	23	8	42	34
FPFH	83	34	57	14	84	30	48	8	72	11
3D distances	20	0	20	0	20	0	25	0	25	0
SURF+FPFH	92	40	75	21	84	30	48	8	67	11
All	85	23	79	7	87	13	59	0	75	0

Table 5 Comparison of the two types of multi-class SVMs

	Test1		Test2		Test3		Test4		Test5	
	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)
“One-versus-all”	85	23	79	7	87	13	59	0	75	0
“One-versus-one”	75	33	53	36	79	54	39	42	50	0

descriptors, proving the usefulness of 3D information also at the feature extraction level. For this test, we applied frontalization and mirroring and we performed classification with “one-versus-all” SVMs. The highest value of TP rate is obtained when all the three types of descriptor are concatenated, thus showing their complementarity for the face recognition task. Finally, Table 5 reports the recognition results with two types of multi-class SVMs: “one-versus-all” and “one-versus-one”. The better results are obtained with the “one-versus-all” type.

4.3 Runtime Performance

The face recognition algorithm described and tested in the previous sections has been implemented in C++ on a standard notebook with an Intel Core *i7 - 5500U* @ 3.0 GHz processor. In Table 6, the runtime of the single operations in our face recognition pipeline are reported. The most demanding operation is the descriptor computation. The overall frame rate is then of about 0.7 s, which suggests that this approach could be used in a real-time scenario with further optimization and with a limited number of people in the dataset.

Table 6 Runtime analysis

	Time (s)
Pose estimation and correction	0.14
Projection to 2D	0.08
Mirroring	0.01
Facial feature extraction	0.05
Descriptor computation	0.4

5 Conclusion

In this work, we presented a face recognition system that exploits depth information acquired with a Microsoft Kinect v2 to improve the performance in case of non-frontal faces. We proposed a depth-based face frontalization algorithm that estimates and corrects the pose of the face from the point cloud of the subjects. To overcome the problem of missing points, we also tested a mirroring technique to improve the results of facial feature detection algorithms. We tested the proposed algorithm on a newly created dataset, the IAS-Lab RGB-D Face Dataset, that contains subjects in different poses, with various expressions and with different light conditions. Experimental results show that even 3D information from consumer RGB-D sensors could play an important role in improving facial recognition results in challenging scenarios. In particular, exploiting depth information for frontal view generation allows the keypoint detection algorithm to fail in a lower number of frames. Furthermore, the 3D information is useful also at the feature extraction level, allowing to extract more robust descriptors. Indeed, the best results have been obtained using both 2D and 3D descriptors. In conclusion, the reported experiments show that the depth-aided approach we propose allows to improve the recognition rate up to 50 %.

References

1. Lu, X.: Image analysis for face recognition. *Personal notes, May*, vol. 5 (2003)
2. Delac, K., Grgic, M., Grgic, S.: Statistics in face recognition: analyzing probability distributions of PCA, ICA and LDA performance results. In: *Image and Signal Processing and Analysis*, pp. 289–294 (2005)
3. Hassner, T., Harel, S., Paz, E., Enbar, R.: Effective face frontalization in unconstrained images. In: *CVPR*, pp. 4295–4304 (2015)
4. Phillips, P.J., Flynn, P.J., Scruggs, T., Bowyer, K.W., Worek, W.: Preliminary face recognition grand challenge results. In: *7th International Conference on Automatic Face and Gesture Recognition*, pp. 15–24 (2006)
5. Turk, M., Pentland, A.: Eigenfaces for recognition. *J. Cogn. Neurosci.* **3**(1), 71–86 (1991)
6. Hua, G., Akbarzadeh, A.: A robust elastic and partial matching metric for face recognition. In: *IEEE 12th ICCV*, pp. 2082–2089 (2009)
7. Taigman, Y., Wolf, L., Hassner, T. et al.: Multiple one-shots for utilizing class label information. In: *BMVC*, pp. 1–12 (2009)

8. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Trans. Image Process.* **19**(6), 1635–1650 (2010)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
10. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **110**, 346–359 (2008)
11. Vu, N.-S., Dee, H.M., Caplier, A.: Face recognition using the POEM descriptor. *Pattern Recogn.* **45**(7), 2478–2488 (2012)
12. Chu, B., Romdhani, S., Chen, L.: 3D-aided face recognition robust to expression and pose variations. In: *CVPR*, pp. 1907–1914. IEEE (2014)
13. Bowyer, K.W., Chang, K., Flynn, P.: A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition. *Comput. Vis. Image Underst.* **101**(1), 1–15 (2006)
14. Fanelli, G., Gall, J., Van Gool, L.: Real time head pose estimation with random regression forests. In: 2011 *CVPR*, pp. 617–624 (2011)
15. Munaro, M., Fossati, A., Basso, A., Menegatti, E., Van Gool, L.: One-shot person re-identification with a consumer depth camera. In: *Person Re-Identification*, pp. 161–181. Springer (2014)
16. Dantone, M., Gall, J., Fanelli, G., Van Gool, L.: Real-time facial feature detection using conditional regression forests. In: *CVPR*, pp. 2578–2585. IEEE (2012)
17. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: *ICRA*, pp. 3212–3217. IEEE (2009)
18. Hsu, C.-W., Lin, C.-J.: A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* **13**(2), 415–425 (2002)
19. Fanelli, G., Dantone, M., Gall, J., Fossati, A., Van Gool, L.: Random forests for real time 3D face analysis. *Int. J. Comput. Vis.* **101**(3), 437–458 (2013)

Lighting- and Occlusion-Robust View-Based Teaching/Playback for Model-Free Robot Programming

Yusuke Maeda and Yoshito Saito

Abstract In this paper, we investigate a model-free method for robot programming referred to as view-based teaching/playback. It uses neural networks to map factor scores of input images onto robot motions. The method can achieve greater robustness to changes in the task conditions, including the initial pose of the object, as compared to conventional teaching/playback. We devised an online algorithm for adaptively switching between range and grayscale images used in view-based teaching/playback. In its application to pushing tasks using an industrial manipulator, view-based teaching/playback using the proposed algorithm succeeded even under changing lighting conditions. We also devised an algorithm to cope with occlusions using subimages, which worked successfully in experiments.

Keywords Robot programming · View-based approach · Neural network · Principal component analysis

1 Introduction

Conventional teaching/playback for robot programming is model-free, depending only on internal sensors, such as encoders for joint angles, and humans do not have to prepare explicit models for tasks and manipulated objects. Therefore, conventional teaching/playback for robot programming is very simple and versatile, as long as task conditions, e.g., the initial poses of manipulated objects, do not change. However, if we have to deal with nontrivial variations in the initial pose of the object or unexpected fluctuations in the pose of the object during manipulation, the conventional scheme is insufficient.

Y. Maeda (✉)

Faculty of Engineering, Yokohama National University, 79-5 Tokiwadai,
Hodogaya-ku, Yokohama 240-8501, Japan
e-mail: maeda@ynu.ac.jp

Y. Saito

Ricoh Corp., Tokyo, Japan

Vision-based robot programming can be used in such cases. In general, model-based image processing, such as feature extraction and pattern matching, is performed in order to detect and localize an object. Robot programming with model-based image processing is powerful because the model is used as prior knowledge. However, the human operator must provide the model in advance and perform camera calibration, which is not necessary in the model-free conventional teaching/playback scheme.

Thus, we have developed a new model-free robot programming method referred to as view-based teaching/playback [7, 8, 11]. Rather than model-based image processing, the proposed method uses view-based (or appearance-based) image processing [9] to move robots. Due to a lack of models, the proposed method cannot handle substantial changes in task conditions. However, the proposed method can deal with nontrivial variations in task conditions to some extent with the generalization ability of artificial neural networks. The proposed view-based teaching/playback worked successfully in experiments conducted in virtual and actual environments [7, 8, 11]. However, because it is view-based, disturbances to camera images by lighting and occlusions may cause serious problems.

In this paper, in order to make our view-based teaching/playback more robust to such disturbances, we present a method to use range images in addition to grayscale images, and a method to switch neural networks online for appropriate robot control. Experimental results obtained using an industrial manipulator are shown.

2 Related Research

A number of previous studies have tackled robot programming through human demonstrations with computer vision [2–4]. While model-based image processing is used in most of the previous studies, adoption of view-based image processing is increasing. Zhang et al. presented a view-based robot control system for fine positioning to grip objects [14]. Zhao et al. also presented a method of appearance-based robot visual servoing [15], which can be used for view-based grasping. Shibata and Iida examined box pushing by a mobile robot using an artificial neural network in a view-based approach [13]. Kato et al. proposed a view-based method for reinforcement learning of robot motion with adaptive image resolution adjustment [5]. They applied their method to pushing using a manipulator. Noda et al. presented a framework for multi-modal integration learning [12], which is applicable to view-based generation of robot motions. Recently Levine et al. achieved view-based grasping of novel objects based on massive learning [6].

3 View-Based Teaching/Playback

3.1 Overview

In the following, we briefly introduce view-based teaching/playback [7, 11]. The method consists of two parts, similar to conventional teaching/playback, namely, a teaching phase and a playback phase.

In the teaching phase, a human operator commands a robot to perform a manipulation task. It is referred to as *demonstration*. All movements of the robot are recorded. All images of the teaching scenes are also recorded by a camera. The task can be repeated two or more times under different task conditions for generalization.

Then, a mapping from the recorded images to the movements is obtained as an artificial neural network. The network is trained with the scene images and movements recorded in the teaching phase. The input of the neural network is a scene image, and its output should be the desirable robot motion corresponding to the image.

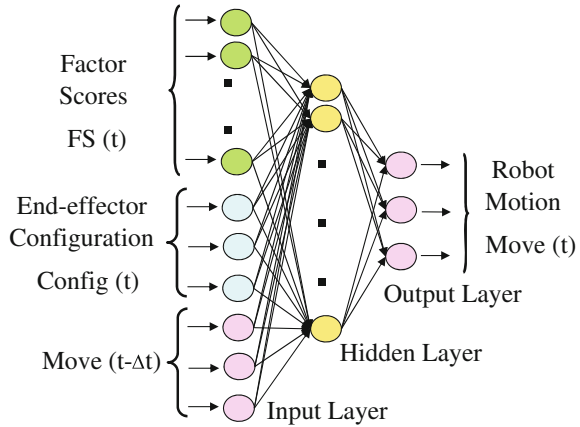
In the playback phase, the motion of the robot is determined by the output of the neural network calculated from scene images. If a neural network is constructed properly, it can reproduce the original robot motion demonstrated in the teaching phase, as long as the task condition remains unchanged. Moreover, even if the condition is changed (for example, the initial pose of the object varies), the neural network may be able to drive the robot to complete the task due to its generalization ability based on multiple demonstrations under different task conditions.

3.2 Mapping

We use a feedforward neural network, as shown in Fig. 1 for mapping from images to robot motion. The output of the neural network is the one-step movement of the manipulator at time t , $\text{Move}(t)$. The input of the neural network consists of factor scores of the current image for principal components, $\text{FS}(t)$, current robot configuration, $\text{Config}(t)$, and the movement of the manipulator in the previous time step, $\text{Move}(t - \Delta t)$, where Δt is the sampling time. The factor scores computed by principal component analysis (PCA) [1] are used instead of the entire pixel data of the camera image, as in [14]. The movement in the previous step is added to make the neural network recurrent so that the network can recognize the “context” of the task. Even if the camera image does not change, the neural network can change the motion of the manipulator.

The weights of the neural network can be computed from image and motion data in demonstrations by backpropagation with momentum (BPM) [10].

Fig. 1 Neural network for robot motion



3.3 Playback Termination

We need to define a termination condition to finish the playback at the goal. Our method is view-based and therefore we cannot obtain the object pose explicitly. Thus we have to implement a view-based condition to terminate the playback. For this purpose, we use the RMS error between factor scores of the current image and those of the goal image. The error, ΔE_S , can be written as follows:

$$\Delta E_S = \sqrt{\frac{1}{N} \sum_{i=1}^N |\text{FS}(t)_i - \text{FS}_{\text{Goal}i}|^2}, \tag{1}$$

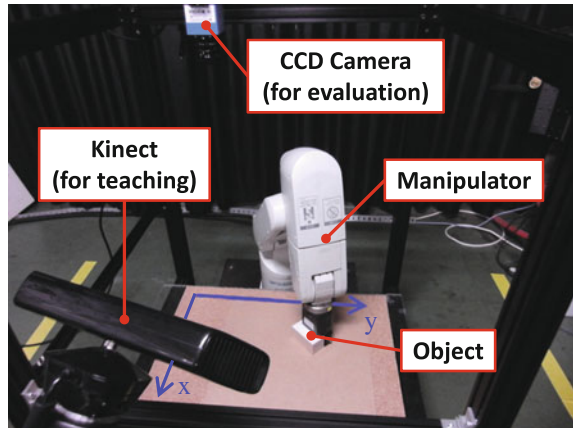
where $\text{FS}(t)_i$ and $\text{FS}_{\text{Goal}i}$ are the factor scores of the current and goal image for the i -th principal component, respectively, and N is the number of principal components used in PCA-based image compression. It should be noted that the factor scores of the current image include information on the current pose of the object implicitly.

Additionally we use the Cartesian distance between the current and goal position of the hand of the manipulator, ΔE_P , which can be obtained from encoders of the manipulator, to terminate the playback.

Now the termination condition can be written as follows:

$$\Delta E_S < T_S \text{ and } \Delta E_P < T_P, \tag{2}$$

where T_S and T_P are threshold values. However, immediate termination of playback when Eq. 2 holds may cause termination on the fringe of the goal region. To avoid this, we continue playback until $\Delta E_S(t) \geq \Delta E_S(t - \Delta t)$.

Fig. 2 Experimental setup

4 Application to Pushing

4.1 Experimental Setup

As in our previous studies [7, 11], we applied the proposed view-based teaching/playback to pushing tasks by an industrial manipulator. We prepared an experimental setup for teaching experiments, as shown in Fig. 2. The manipulated object was a wood block having dimensions of 80 [mm] \times 40 [mm] \times 40 [mm].

We used an RV-1A six-axis industrial manipulator (Mitsubishi Electric). The manipulator accepts position commands via Ethernet from a controller PC with a Core i7 860 CPU (2.8 GHz). Another back-end PC with a Core i7 2600 CPU (3.4 GHz) was used for offline computation, including PCA and BPM.

A Kinect camera (Microsoft) was used to obtain 640 pixel \times 480 pixel RGB and range images at 30 [Hz]. The RGB images were converted to grayscale images for view-based teaching/playback. Note that the Kinect camera requires neither intrinsic nor extrinsic calibration because the proposed method is view-based. Another monochrome CCD camera, DMK21F04 (Imaging Source), was installed only for evaluation of playback errors.

4.2 Teaching Phase

4.2.1 Demonstration

Let us consider pushing of an object on a plane as a target task. This pushing manipulation requires three-degree-of-freedom planar motion of the end-effector of the manipulator, which is used to push the object to the goal. We used a gamepad with

thumbsticks to command the 3-DOF motion of the end-effector. In order to avoid sudden changes in motion, a first-order lag filter was applied.

4.2.2 Mapping with a Neural Network

The input of a neural network includes the top 50 factor scores of the camera images (grayscale or range images) in the teaching phase.

The output of the neural network is the $[0, 1]$ -normalized 3-DOF motion of the end-effector in one time step, $\text{Move}(t) = (\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{\theta})$, as follows:

$$\Delta\hat{x} = 0.5(\Delta x / \Delta x_{\max} + 1) \quad (3)$$

$$\Delta\hat{y} = 0.5(\Delta y / \Delta y_{\max} + 1) \quad (4)$$

$$\Delta\hat{\theta} = 0.5(\Delta\theta / \Delta\theta_{\max} + 1), \quad (5)$$

where Δx , Δy , and $\Delta\theta$ are the relative motions of the end-effector along the x -, y -, and θ -axes, respectively, and Δx_{\max} , Δy_{\max} , and $\Delta\theta_{\max}$ are the upper limits of their absolute values, respectively. We set $\Delta x_{\max} = \Delta y_{\max} = 0.5$ [mm] and $\Delta\theta_{\max} = 0.0085$ [rad].

4.2.3 Neural Network Learning

We use the three-layered feedforward neural network shown in Fig. 1. The network has three neurons for its 3-DOF output, 56 neurons for its input (50 neurons for $\text{FS}(t)$, three neurons for $\text{Config}(t)$, and three neurons for $\text{Move}(t - \Delta t)$), and 50 neurons in its hidden layer. All of the weights of the neural network are calculated by BPM using a momentum factor of 0.9. Backpropagation with momentum is iterated until the residual converges or the computation time reaches its limit (1,200 [s] per one demonstration in this study).

4.3 Playback Phase

In the playback phase, robot motion is determined by the neural network shown in Fig. 1 using camera images (grayscale or range ones). Position commands are sent to the manipulator at 30 [Hz]. We set $T_S = 2.0$ and $T_P = 2.0$ [mm].

5 View-Based Teaching/Playback Using Range Images

We compared view-based teaching/playback with range images and that with grayscale images.

We changed the lighting conditions in order to confirm the robustness of the proposed teaching/playback. Specifically, we turned off a fluorescent lamp on the ceiling just before the playback phase (Fig. 3). Obviously, grayscale images are affected by such lighting changes, but range images should not from an idealistic viewpoint.

In the teaching phase, a human operator demonstrated a pushing operation with the ceiling lamp turned on. Some grayscale and range images obtained in the demonstration are shown in Fig. 4. Note that the range images shown in the present paper are colored for ease of comprehension. Neural networks were trained based on the demonstration for 485 range or grayscale images with 640 pixels \times 480 pixels.

The paths of the end-effector of the manipulator in the playback phase with the lamp turned off are shown in Fig. 5. Playback using grayscale images became

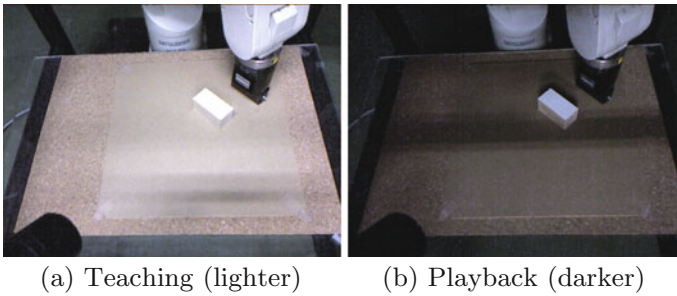


Fig. 3 Change in the lighting conditions

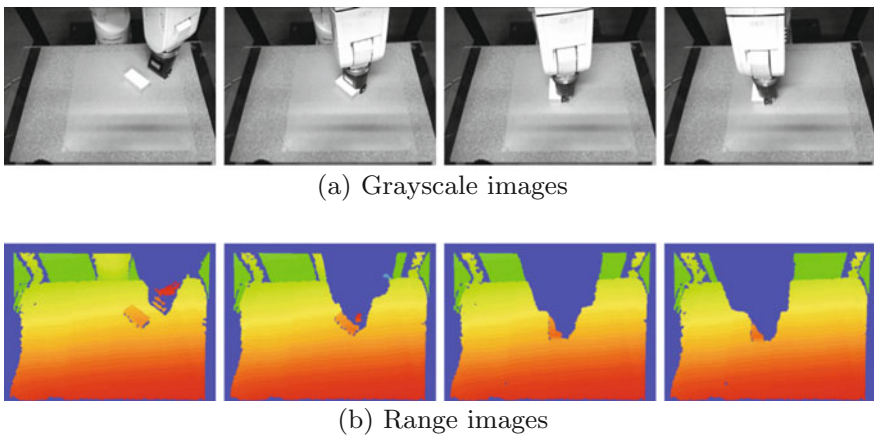


Fig. 4 One demonstration

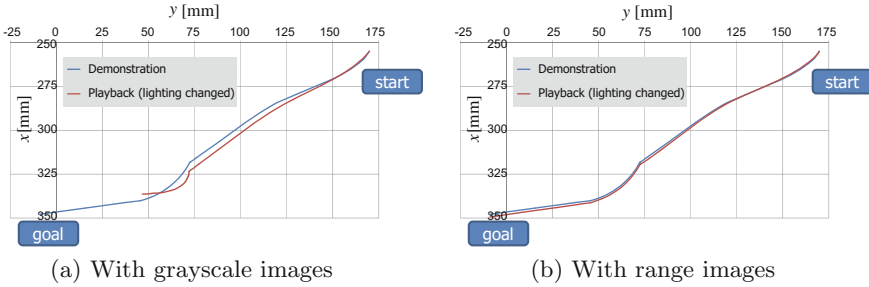


Fig. 5 End-effector paths in teaching and playback

stuck before reaching the goal. On the other hand, playback using range images successfully reproduced the demonstration, as expected. This indicates that view-based teaching/playback using range images is robust to changes in lighting conditions. When the lighting condition is stable, however, using grayscale images is preferable because of larger noise in range images.

6 Teaching/Playback Using Grayscale and Range Images

In this section, we present teaching/playback using both grayscale and range images in order to simultaneously achieve robustness and accuracy.

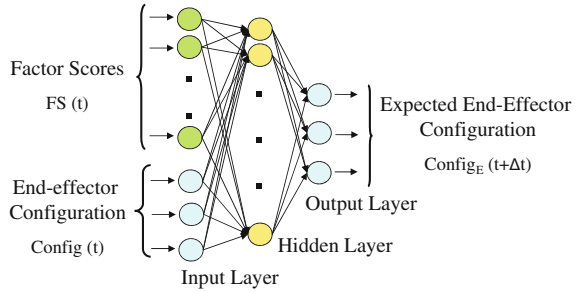
6.1 Detection of Unexpected Situations

We applied an algorithm for the detection of unexpected situations [11] to view-based teaching/playback. Here, we briefly describe the detection algorithm.

When the termination condition (Eq. 2) does not hold, the robot continues to move. If the robot encounters unexpected situations due to disturbances, it is difficult to achieve the manipulation goal, and continuation of the movement may be dangerous. Thus, we developed a view-based method to detect unexpected situations for safety [11].

In the proposed method, we use the neural network shown in Fig. 6. The output of this neural network, $Config_E(t + \Delta t)$, is the expected absolute pose of the end-effector in the next time step, whereas that of the neural network for motion control (Fig. 1) is the relative robot motion. The outputs of the two neural networks should be consistent for the neighborhood of the inputs used in the teaching phase, but inconsistent away from the neighborhood of the inputs. Thus, the inconsistency of the outputs of the two neural networks indicates that the robot is in an unexpected situation.

Fig. 6 Neural network used to detect unexpected situations



Concretely, the condition for the detection of unexpected situations can be written as follows:

$$\Delta E_A = \|\text{Config}(t) + \text{Move}(t) - \text{Config}_E(t + \Delta t)\| > T_A, \quad (6)$$

where T_A is a threshold, and the norm of the robot configuration is defined as follows:

$$\|(\Delta x, \Delta y, \Delta \theta)\| = \sqrt{|\Delta x|^2 + |\Delta y|^2 + |r\Delta \theta|^2}, \quad (7)$$

where r is a weight coefficient. If Eq. (6) holds, we terminate the robot motion urgently as an emergency stop.

6.2 Switching Neural Networks

We apply the above algorithm to the switching of two neural networks: one uses range images and the other uses grayscale images. The networks are used to control the robot as follows:

1. In the teaching phase, both range images and grayscale images are obtained. A neural network with range images and that with grayscale images are trained. Additionally, a neural network to detect unexpected situations is also trained for grayscale images.
2. In each time step of the playback phase, if Eq. (6) holds, the neural network with grayscale images cannot cope with the current situation. In this case, the neural network with range images, which is more robust, is used for robot motion. Otherwise, the neural network with the grayscale images is used.

6.3 Experiments

We demonstrated manipulation, as shown in Fig. 7, to test the above switching algorithm. We set $r = 42.1$ [mm] and $T_A = 25$ [mm] in the experiments. In the playback phase, we turned the fluorescent lamp off and on, as shown in Fig. 8a. Switching result in the playback is shown in Fig. 8b. This indicates that the proposed method switched images for robot control online and carried the object to the goal successfully.

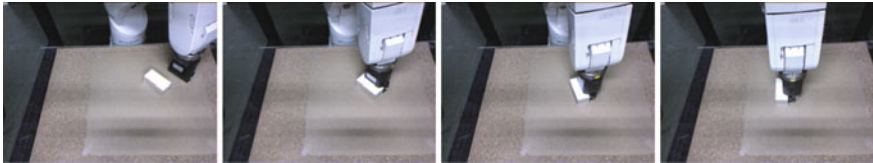
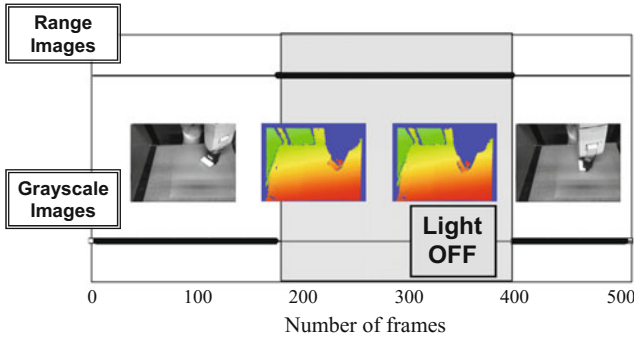


Fig. 7 Demonstration with recording range and grayscale images



(a) Playback



(b) Images Used in Playback

Fig. 8 Playback using range and grayscale images

7 Occlusion-Robust Teaching/Playback

View-based image processing is affected by occlusion. In fact, not only foreground disturbances, but also background disturbances may cause serious problems. In this section, in order to make our view-based teaching/playback occlusion-robust, we introduce neural networks with subimages.

First, in the teaching phase, not only a neural network with full images (Fig. 1), but also neural networks with subimages are generated. Here, we consider a full image (640 pixels \times 480 pixels), a left-side subimage (448 pixels \times 480 pixels), and a right-side subimage (448 pixels \times 480 pixels). When occlusion occurs, the neural network with full images will fail to generate appropriate robot motion. However, if the occlusion is local, a neural network with subimages may not be affected by the occlusion and, therefore, may work well. Thus, switching neural networks would make the proposed view-based teaching/playback occlusion-robust.

7.1 Switching Neural Networks

As in the previous section, we switch neural networks as follows:

1. In the teaching phase, neural networks for robot motion with subimages are generated, in addition to that with full images. Moreover, a neural network to detect unexpected situations is also generated for each of the neural networks for robot motion.
2. In each time step of the playback phase, if Eq. (6) holds, the neural network with full images cannot cope with the current situation. In this case, ΔE_A for each subimage is computed, and a subimage with the minimum ΔE_A is used for robot motion.

The above online switching algorithm enables us to use the neural network that is least affected by occlusion for robot motion control.

7.2 Experiments

We demonstrated manipulation as shown in Fig. 9. We set $T_A = 10$ [mm] in the experiments. In addition to full grayscale images, left- and right-side subimages were used to generate neural networks. During the playback phase, we generated occlusions on the right and left sides, as shown in Fig. 10. The proposed method switched neural networks online in order to avoid occlusion effects and successfully carried the object to the goal. By using more subimages, we would be able to deal with a larger variety of occlusions.

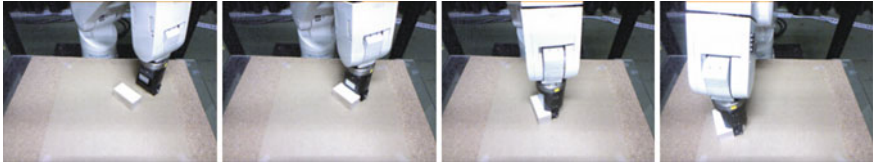
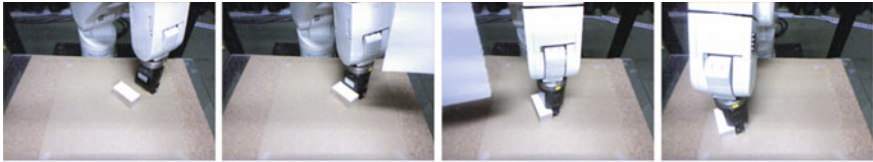


Fig. 9 Demonstration for occlusion-robust programming



(a) Playback



(b) Images Used in Playback

Fig. 10 Playback using full images and subimages

7.3 Integration of Lighting-Robust Teaching/Playback

The above algorithm can be easily integrated to enable switching of grayscale and range images shown in the previous section. The integrated switching algorithm can be summarized as follows:

- Use full grayscale images when $\Delta E_{Ag} < T_{Ag}$,
- Use full range images when $\Delta E_{Ag} \geq T_{Ag}$ and $\Delta E_{Ar} < T_{Ar}$,
- Use range subimages with the minimum ΔE_{Ar} when $\Delta E_{Ag} \geq T_{Ag}$ and $\Delta E_{Ar} \geq T_{Ar}$,

where ΔE_{Ag} and ΔE_{Ar} are ΔE_A calculated with grayscale and range images, respectively, and T_{Ag} and T_{Ar} are thresholds for grayscale and range images, respectively. We set $T_{Ag} = 5.0$ [mm] and $T_{Ar} = 8.0$ [mm] in the experiments.

We investigated the performance of the proposed method for changing lighting conditions (turning off the fluorescent lamp) and occlusions, as shown in Fig. 11. The proposed method successfully used range images when the lighting condition changed and range subimages when occlusions occurred. Consequently, the object was carried to the goal successfully.

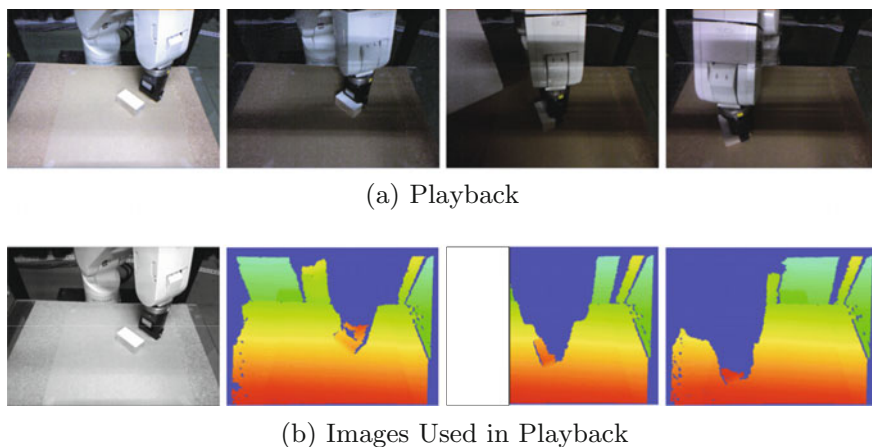


Fig. 11 Playback using grayscale images, range images, and range subimages

8 Conclusion

In the present paper, we presented view-based teaching/playback using range images in addition to grayscale images. Playback using range images leads to robustness to changes in lighting conditions, while playback using grayscale images leads to good positioning accuracy for unchanged lighting conditions. Using appropriate switching of neural networks, the proposed view-based teaching/playback successfully combined both advantages. Moreover, we proposed subimage-based teaching/playback to achieve robustness to occlusions. The experimental results revealed that the proposed view-based teaching/playback is lighting- and occlusion-robust.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number JP24560286 and JP15K05890.

References

1. Abdi, H., Williams, L.J.: Principal component analysis. Wiley Interdiscipl. Rev.: Comput. Stat. **2**(4), 433–459 (2010)
2. Argall, B.D., Chernovab, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. *Robot. Auton. Syst.* **57**(5), 469–483 (2009)
3. Billard, A., Calinon, S., Dillman, R., Schaal, S.: Robot programming by demonstration. In: Siciliano, B., Khatib, O. (eds.) *Springer Handbook of Robotics*, chap. 59.2, pp. 1371–1394. Springer, Berlin (2008)
4. Dillmann, R., Asfour, T., Do, M., Jäkel, R., Kasper, A., Azad, P., Ude, A., Schmidt-Rohr, S.R., Lösch, M.: Advances in robot programming by demonstration. *Künstliche Intelligenz* **24**(4), 295–303 (2010)

5. Kato, M., Kobayashi, Y., Hosoe, S.: Optimizing resolution for feature extraction in robotic motion learning. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, pp. 1086–1091 (2005)
6. Levine, S., Pastor, P., Krizhevsky, A., Quillen, D.: Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. arXiv preprint [arXiv:1603.02199](https://arxiv.org/abs/1603.02199) (2016)
7. Maeda, Y., Moriyama, Y.: View-based teaching/playback for industrial manipulators. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 4306–4311 (2011)
8. Maeda, Y., Nakamura, T.: View-based teaching/playback for robotic manipulation. *ROBOMECH J.* **2**(1) (2015), doi:[10.1186/s40648-014-0025-4](https://doi.org/10.1186/s40648-014-0025-4)
9. Matsumoto, Y., Inaba, M., Inoue, H.: View-based navigation using an omniview sequence in a corridor environment. *Mach. Vision Appl.* **14**(2), 121–128 (2003)
10. Meireles, M.R.G., Almeida, P.E.M., Simões, M.G.: A comprehensive review for industrial applicability of artificial neural networks. *IEEE Trans. Ind. Electron.* **50**(3), 585–601 (2003)
11. Moriyama, Y., Maeda, Y.: View-based teaching/playback for manipulation by industrial robots. *Trans. Jpn. Soc. Mech. Eng. Series C* **79**(806), 3597–3698 (2013) (in Japanese)
12. Noda, K., Arie, H., Suga, Y., Ogata, T.: Multimodal integration learning of robot behavior using deep neural networks. *Robot. Auton. Syst.* **62**(6), 721–736 (2014)
13. Shibata, K., Iida, M.: Acquisition of box pushing by direct-vision-based reinforcement learning. In: Proceedings of SICE Annual Conference, pp. 1378–1383 (2003)
14. Zhang, J., Knoll, A., Schmidt, R.: A neuro-fuzzy control model for fine-positioning of manipulators. *Robot. Auton. Syst.* **32**(2–3), 101–113 (2000)
15. Zhao, Q., Sun, Z., Sun, F., Zhu, J.: Appearance-based robot visual servo via a wavelet neural network. *Int. J. Control Autom. Syst.* **6**(4), 607–612 (2008)

Part XVI
Robotic Arm

Development of a Portable Compliant Dual Arm Robot

Zhifeng Huang, Chingszu Lin, Ping Jiang, Taiki Ogata and Jun Ota

Abstract In this research, we aimed at designing a portable and safe dual arm robot to help people, especially for the elder or disable people in their daily life. With the consideration of portable, we limited the weight of the robot to be lower than 7 kg. The weight enables a person to lift the robot, even though by single arm. To overcome this challenging, most of the parts of the robot was designed to be manufactured by plastic. On the other hand, for the safety, we design a new type of passive compliant unit to sense the torque of the joint and buffer the impact of collisions. In each arm of the robot, there were six degrees of freedom, three in the shoulder and one in the elbow. In order to detect and buffer the impact of collision which occurred in any position of the arm, passive compliant unit was installed in both shoulder and elbow. In addition, there was also two degree of freedoms in the wrist joints for the robot to adjust the orientation of the end effector. In order to examine our design, the whole robot was manufactured by 3D printer with ABS material, except the motor, bearing and the screw. Finally, an experiment was conducted to test the proposed dual arm robot's basic performance. The result showed that the payload of the robot was up to 500 g and the maximum reach is up to 400 mm. In addition, utilized the passive compliant units of the shoulder and elbow, the robot arm was able the buffering impact.

Keywords Robot manipulator · Housework robot · Compliance design · Passive compliance · Portability

Z. Huang (✉)

School of Automation, Guangdong University of Technology, Guangzhou, China
e-mail: zhifeng@gdut.edu.cn

C. Lin · P. Jiang · T. Ogata · J. Ota

Research into Artifacts, Center for Engineering (RACE), The University of Tokyo,
Chiba, Japan

1 Introduction

In daily life, we all have the experiences that we were doing different housework in different areas. For an example, we cut the vegetable or fruit in the kitchen, while we might fold our clothes in the bedroom. Different with the industry tasks which were required the robot to be fixed on the working station and to process the tasks with high positional accuracy and powerful payload, housework required more force control rather than high positional accuracy, and much less payload. In addition, we should notice that most of the housework need dual arm to cooperate to finish the tasks. Since the robot might be close to the human beings or directly interact with them in most of the house work, the safety of the robot should be carefully ensured. One of the basic safe specifications of the robot should be the capacity of buffering the impact of the collision between the human and robot.

Based on the naturally observation and consideration above, we can conclude that a robot which was suit for housework for human beings in their daily life should include four basic specifications as follows:

- (i) Mobility, working space can be changed easily;
- (ii) Capacity of force control, the end effector's operation force can be controlled;
- (iii) Capacity of dual arm task;
- (iv) Safety, the robot should be able to buffer the impact of collision between the robot and human.

In previous study, researchers have developed many dual arm robots for the housework, such as PR2 [1], Baxter [2], HRP2 [3], and Honda's ASIMO Robot [4]. Many methods were proposed to enable these robots to perform various housework, such as cloth folding [5, 6], assisting human being in clothing [7, 8], opening the refrigerator [9], and cooking [10] so on. However, until now, we still cannot see any robot be used in our real daily life to do the housework. Considered with the basic specifications defined above, one of the problems which prevent the dual arm robots from assisting human being in real daily life might be the low mobility. Most of these robots had a mobile platform. The HRP2 and ASIMO even can walk by foot. Unfortunately, their mobility was still insufficient when working in the real house environment, since of the varied conditions and obstacle. The robot move itself from one area to another area might cost more time than processing a task of housework. To support our viewpoint, we compared the dual arm robot including the robots developed in laboratory and which already has been commercialized. We used two indexes to evaluate the robot performance including the weight and payload. The weight was related to robot's size and the cost of mobility, while the payload is basic ability in processing task. As shown in Fig. 1, most of the dual arm robot system's weight was large than 29 kg. The lightest robot was KAWADA's NEXTAGE Robot [11] which was 29 kg. A great part of the robot's weight was

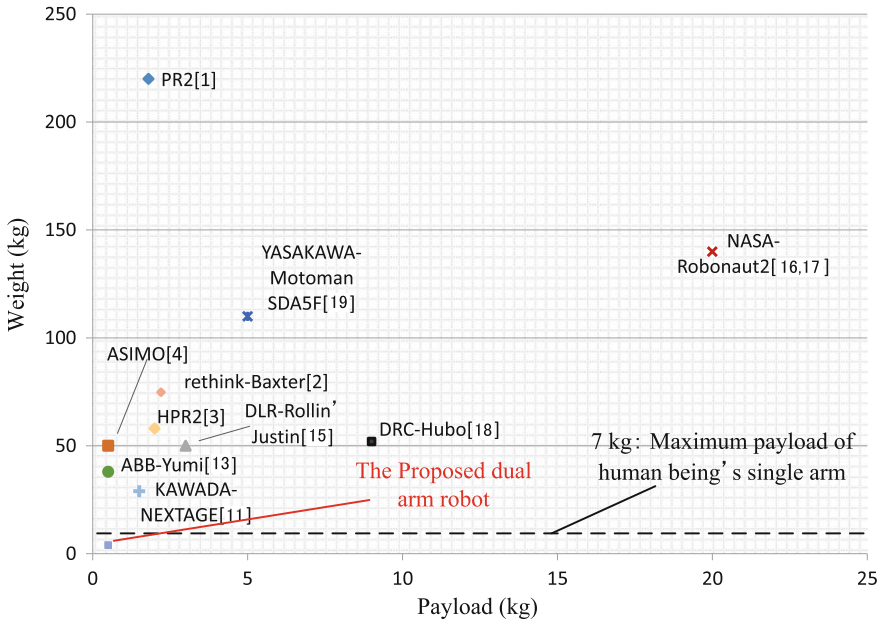


Fig. 1 Comparison of dual arm robot system

contributed by the mobile platform. However, the mobile platform did not improve the robot’s mobility. Huge size and heavy weight prevent the robot from working efficiently in the real daily life for housework. With reverse thinking, we might take out mobile platform and make the robot to be light enough to be moved by manual easily for changing working space. A light weight dual arm robot which can be lift by single arm of an adult might be much more suit for the application in housework. For example, the robot can be taken to the kitchen room to assist people in cooking, and then be taken to the laundry for folding the clothes.

In this paper, we target on developing a portable dual arm robot system, with the consideration of basic specification of housework discussed above. The robot’s weight was limited to enable to be lift by an adult with single arm. In order to make the robot be light weight, most of the robot’s parts were design to can be manufactured by plastic. In addition, a new type of passive compliant unit was design to ensure the robot’s safety. The passive compliant units enable the robot not only to buffer the impact of collision between human and robot but also to sense force of the end effector. In addition, the passive joints was designed carefully to make its parts can be made by plastic.

The remainder of the paper is structured as follows. Section 2 describes the robot’s specification. Section 3 details the hardware of the robot including joints configuration and the mechanical structure of the compliant unit. Section 4 presents the results of an experiment carried out to examine the dual arm robot’s basic performance. Section 5 concludes the paper.

2 Specifications

In order to ensure the dual arm robot to be portable, the weight of the robot should not exceed an adult's single arm's capacity. According to the NASA's report of human performance capacities [12], the limitation of is 7.7 kg. In our design, to ensure the robot can be lift easily by single arm, we set the weight limit to be 6 kg, which is about 80 % of the maximum capacity.

The payload of each robot arm was set to 0.5 kg. This specification was referred to the ASIMO [4] and ABB's YUMI [13]. The payload enables the robot arm to manipulate most of the daily life object, such as a dish, an apple, a cloth and so on.

The reach of the arm was set to be 400. This reach range enable to dual arm robot to manipulate the object which was within 800 mm × 400 mm × 400 mm. This specification can fulfill most of the housework tasks.

Finally, in order to buffer the impact of collision between human and robot, the robot arm should have inherent compliance.

3 Mechanical Design

3.1 Configuration of the Robot

The configuration of the robot was shown in Fig. 1. The dual arm robot has 12 degrees of freedom (DoF) totally. Each arm has 6 DoFs. There were 3 DoFs in the shoulder and 1 Dof in the elbow. In order to ensure the whole robot to buffer the impact of collision in any position and orientation, passive compliant unit was installed in all the joints of the shoulder and the elbow. By this way, no matter the collision was occurred in the upper arm or forearm, the passive compliant unit would enable to robot to buffer the impact. These passive compliant units also enabled the robot to sense the force and orientation of the impact. In addition, there were 2 DoFs in the robot's wrist.

Actuators were installed in all DoFs of the robot. We chose the RC servo (Futaba Co., Ltd) as the actuators of the joints. The RC servos were command by serial communication which could simplify the cable of the robot. More detail specification of each joint was shown in Table 1 (Fig. 2).

3.2 Mechanical Design of the Passive Compliant Unit

In order to ensure safety of the dual robot arm when collision was occurred with human, we design the passive compliant unit to make the robot to be inherent compliance. The principle of the passive compliant unit was shown in Fig. 3a. Spring was installed between the output flanges of the actuators and the joint's link.

Table 1 Dynamics parameters of the dual-arm robot

Joint ^a	Max. Torque (Nm)	Max. velocity (°/s)	Motion range (°)
1	4.8	100	-150 to 150
2			-110 to 110
3			-150 to 150
4			-123 to 123
5	0.7	200	-135 to 135
6			-130 to 50

^aThe number of the joints was indicated in Fig. 2

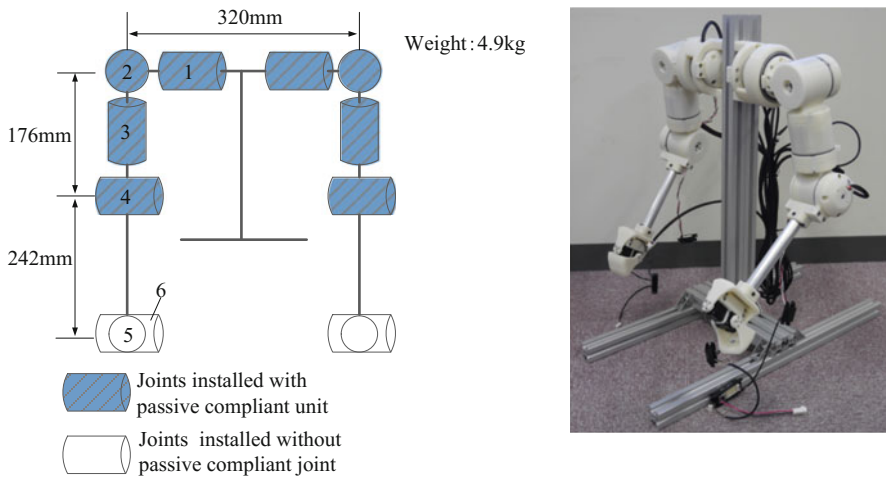
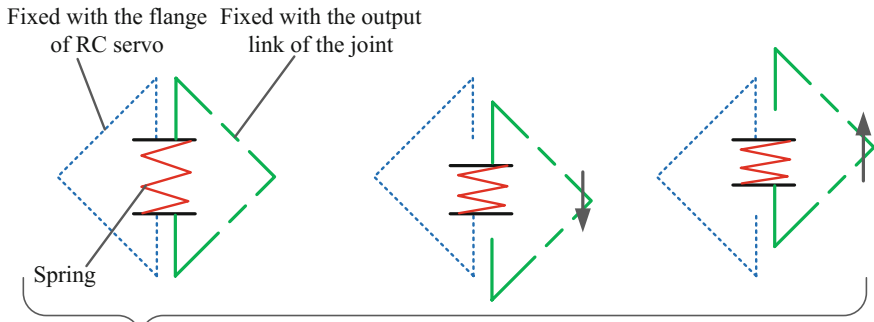


Fig. 2 Joint configuration of the robot

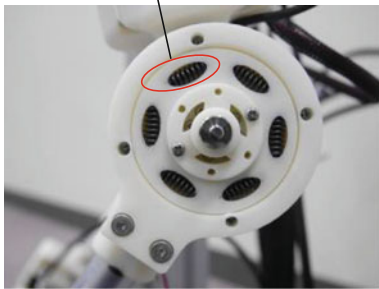
As a result, when people hit the arm of the robot, the spring will be compressed first. Through this way, the impact time would be extended while the impact force would be diminished. Comparing with the rigid joint design which directly fixed the actuator's output flange with the joint's link, one of the outstanding advantages was that the passive compliant unit gave the joint inherent compliance. Utilizing the torque sensor and control methods also could realize compliance in rigid joint design. However, the method relied on the high-performance controller and might lead the response of impact to be delayed.

In order to make the weight be light, the proposed passive compliant unit was designed to be able to be made by plastic, such as ABS. We optimized the part which was fixed with the actuator and the part which was fixed the joint. This design was different with the former design of the passive compliant units [14].

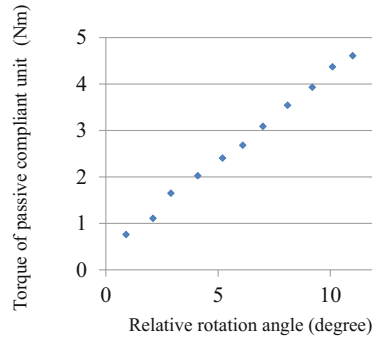
The performance of the passive compliant units was shown in Fig. 3c. The relationship between the relative rotation angle and the torque was linear. Therefore, the torque can be easily measured by angle sensor.



(a) Principle of passive compliant unit



(b) Passive compliant unit manufacture by 3D printer



(c) Relationship between relative rotation angle and output torque

Fig. 3 Passive compliant unit

4 Test of Basic Performance

Two experiments were carried out to evaluate the basic performance of the robot arm, including the payload and the inherent compliance which was related to the capacity of buffering the impact.

4.1 Experiment of Evaluating the Robot Arm's Payload

The experiment was carried out to test the payload of the robot. The robot's wrist was fixed with 500 g weight as shown in Fig. 4. Then the robot was programmed to raise the arm within 3 s. The shoulder joint rotated 90° with the speed of 30 %/s, while the elbow joint rotate 120° with the speed of 40 %/s.

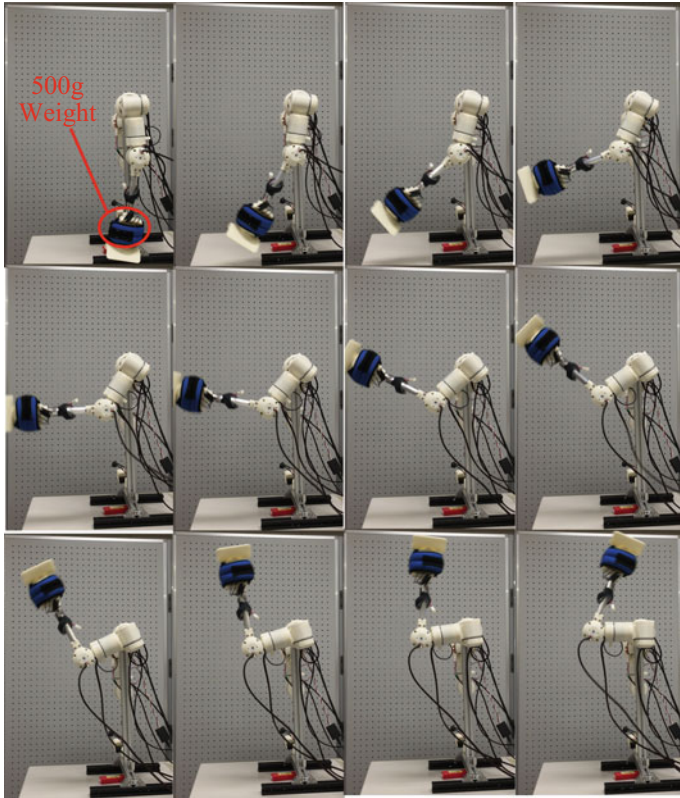


Fig. 4 Robot raised the arm with weight of 500 g

4.2 Experiment of Evaluating the Inherent Compliance

In order to evaluate the inherent compliance of the proposed robot arm, impact was applied on the robot arm. The experiment set up was shown in Fig. 5a, b. The robot's elbow joint was programmed to be 90° . All joints' RC servo was working in brake mode to keep the robot arm's configuration. The upper arm kept vertical, while the forearm kept horizontal. In addition, in order to simulate the situation that the robot was holding the object, there was 250 g loaded on the robot's wrist. A 250 g sandbag was connected to a frame which was next to the robot arm. The sandbag was raised to horizontal level and then be released to swing to hit the robot's wrist by gravity. Through the potentiometer type angle sensor which installed in the passive compliant units, the relative rotation angle of the unit was measured. The angle indicated the inherent compliance of the robot arm. Their relationship was considered as positive correlation.

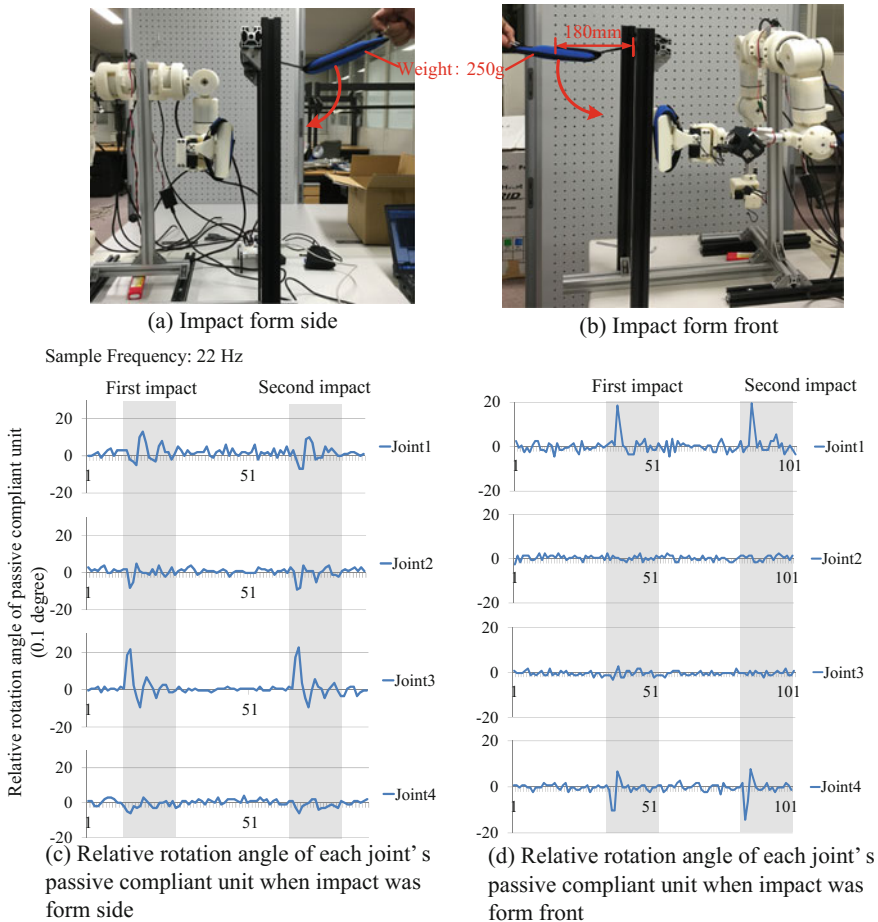


Fig. 5 Results of the experiment of evaluating inherent compliance of the robot arm

4.3 Result and Discussion

The result of the experiment revealed that the robot arm's payload achieved 500 g which met the specification defined in Sect. 2.

The result of the experiment which applied the impact on the robot arm proved the inherent of the proposed robot arm. Although all the actuator of the robot arm was keeping braked, the joint rotation was observed. In the situation of the impact form side, Joint 1 and 3 rotated 1.5° and 2° respectively. This passive rotation angle buffered the impact. The similar results were observed in the situation of the impact from front. The inherent compliance was contributed by the proposed passive compliant unit. The result revealed that the proposed passive compliant unit was effective.

The difference of the joints' performance between two orientations of impact should be noted. When the side impact occurred, the main contribution of buffering was from the Joint 1 and 3. When the front impact occurred, the main contribution of buffering was from Joint 1 and Joint 4. The result indicated that, robot's compliance was not only related to the passive compliant unit and the configuration itself, but also affected by the impact orientation. In order to obtain a good compliance of the whole robot arm, the configuration of the robot should be optimal according to the impact orientation. It might be a challenging problem in our future works.

5 Conclusion

In this research, we proposed a portable dual arm robot to assist human being in their housework, especially for the elder or disabled people. The robot was only 4.9 kg and the payload achieved 500 g per arm. We designed the passive compliant unit to make the robot be inherently compliant. Most of the robot's part was manufactured by 3D printer with ABS material. The experiment results proved the robot's performance met the basic specifications for processing housework. In future work, the robot's function will be expanded for applying in the real daily life.

Acknowledgements This work was supported by the Natural Science Foundation of Guangdong Province, China (Grant No. 2016A030310350) and Grants-in-Aid for Scientific Research (KAKENHI) from the Japan Society for the Promotion of Science (JSPS) (No. 26120006 & No. 15H02761).

References

1. Garage, W.: PR2 user manual. http://www.clearpathrobotics.com/wp-content/uploads/2014/08/pr2_manual_r-321.pdf
2. Guizzo, E., Ackerman, E.: The rise of the robot worker. *IEEE Spectrum* **49**(10), 34–41 (2012)
3. Kaneko, K., Kanehiro, F., Kanta, S., Hirukawa, H., Kawasaki, T., Hirata, M., Isozumi, T.: (2004). Humanoid robot HRP-2. In: *IEEE International Conference on Robotics and Automation*, pp. 1083–1090
4. Hirose, M., Ogawa, K.: Honda humanoid robots development. *Philos. Trans. R. Soc. Lond. A: Math. Phys. Eng. Sci.* **365**(1850), 11–19 (2007)
5. Bersch, C., Pitzer, B., Kammel, S.: Bimanual robotic cloth manipulation for laundry folding. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1413–1419. IEEE (2011)
6. Miller, S., Van Den Berg, J., Fritz, M., Darrell, T., Goldberg, K., Abbeel, P.: A geometric approach to robotic laundry folding. *Int. J. Robot. Res.* **31**(2), 249–267 (2012)
7. Gao, Y., Chang, H. J., & Demiris, Y.: User modelling for personalised dressing assistance by humanoid robots. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1840–1845. IEEE (2015)
8. Koganti, N., Nge0, J. G., Tomoya, T., Ikeda, K., Shibata, T.: Cloth dynamics modeling in latent spaces and its application to robotic clothing assistance. In: *2015 IEEE/RSJ*

- International Conference on Intelligent Robots and Systems (IROS), pp. 3464–3469. IEEE (2015)
9. Yamazaki, K., Ueda, R., Nozawa, S., Kojima, M., Okada, K., Matsumoto, K., Inaba, M.: Home-assistant robot for an aging society. *Proc. IEEE* **100**(8), 2429–2441 (2012)
 10. Watanabe, Y., Nagahama, K., Yamazaki, K., Okada, K., Inaba, M.: Cooking Behavior with Handling General Cooking Tools based on a System Integration for a Life-sized Humanoid Robot. *Paladyn, J. Behav. Robot.* **4**(2), 63–72 (2013)
 11. KAWADA, NEXTAGE datasheet, <http://nextage.kawada.jp/en/specification/#specHontaiTable>
 12. NASA, Human Performance Capabilities. http://msis.jsc.nasa.gov/sections/section04.htm#_4.6_KINESTHESIA
 13. ABB, Yumi robot datasheet, <http://new.abb.com/products/robotics/yumi>
 14. Tsagarakis, N.G., Laffranchi, M., Vanderborght, B., Caldwell, D.G.: A compact soft actuator unit for small scale human friendly robots. In: IEEE International Conference on Robotics and Automation, 2009. ICRA'09, pp. 4356–4362. IEEE (2009)
 15. Fuchs, M., Borst, C., Giordano, P.R., Baumann, A., Kraemer, E., Langwald, J., Burger, R.: Rollin'Justin-design considerations and realization of a mobile platform for a humanoid upper body. In: IEEE International Conference on Robotics and Automation, 2009. ICRA'09, pp. 4131–4137. IEEE (2009)
 16. Diftler, M.A., Mehling, J.S., Abdallah, M.E., Radford, N.A., Bridgwater, L.B., Sanders, A. M., Hargrave, B.K.: Robonaut 2-the first humanoid robot in space. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 2178–2183. IEEE (2011)
 17. Tzvetkova, G.V.: Robonaut 2: mission, technologies, perspectives. *J. Theoret. Appl. Mech.* **44** (1), 97–102 (2014)
 18. Zucker, M., Joo, S., Grey, M.X., Rasmussen, C., Huang, E., Stilman, M., Bobick, A.: A general-purpose system for teleoperation of the DRC-HUBO humanoid robot. *J. Field Robot.* **32**(3), 336–351 (2015)
 19. YASAKAWA, Motorman SDA5F datasheet, <http://www.motoman.com/datasheets/SDA5D.pdf>

Design, Analysis and Simulation of a Device for Measuring the Inertia Parameters of Rigid Bodies

Yu Liu, Song Huang, Li Jiang and Hong Liu

Abstract A device for measuring the inertia parameters of rigid bodies has been presented in this paper. It's actually a 3-URU pure rotation parallel mechanism. To improve the measuring accuracy, an adjusting mechanism composed of dovetail guides, bevel gears and a motor in the measuring device, is adopted to facilitate the adjustments of the center of gravity of the rigid body. Only three 16-bits encoders and three load cells are needed. The direct kinematic model of the parallel mechanism is built and the kinematic analysis is accomplished. The dynamic behaviors are investigated by Adams modeling. Simulation results are presented and show that the design of the device can meet the technical demand on inertia parameter measurement.

Keywords Measuring device · Inertia parameters · Adjusting mechanism · Kinematic analysis · Dynamic simulation

1 Introduction

The inertia properties of a rigid body include a set of parameters: mass, center of mass location and inertia tensor (moments and products of inertia). They play a very important role in engineering process such as the dynamic simulation of manipulator components and space crafts [1] and the study on the comfort of vehicles [2, 3], etc. In a general way, the inertia parameters can be estimated by solid or finite element modeling. However, if the model is hard to obtain, it requires some other method to measure the inertia parameters.

Researchers have developed kinds of methods [4–12] for measuring the inertia parameters of rigid bodies. C. Schedlinski had classified the method in detail [13], including dynamic balancing method [5], run-down method [6], measurement

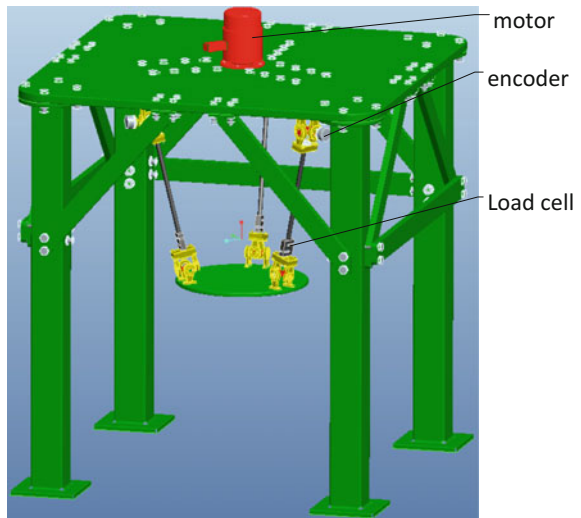
Y. Liu · S. Huang · L. Jiang (✉) · H. Liu
State Key Laboratory of Robotics and System, Harbin Institute of Technology,
150080 Harbin, People's Republic of China
e-mail: jiangli01@hit.edu.cn

robot method [7], gravitational pendulum method [8], torsional pendulum method [5], multi-filar pendulum method [9, 11] and modal method [12]. For now, the relative error of dynamic balancing method can reach 8 % and the relative error of modal method can even higher. The pendulum method need several times of measurement to identify all the ten inertia parameters, this would make the measuring time very long and more times of measurement may introducing random error. It's of great significance in engineering to design a simple and efficient device for estimating the inertia tensors. In the paper, a new measuring device is presented. An adjusting mechanism can help to optimize the measuring accuracy. Kinematic analysis and simulation of the device are accomplished.

2 Measuring Device

The device, shown in Fig. 1, is actually a 3-URU pure rotation parallel mechanism of 3 degree of freedom (3-DOF), consist of a mobile platform used for carrying the measured rigid body, three connecting rods and a fixed platform. The mobile platform is a plate made of aluminum alloy and the fixed platform is a steel plate. The connecting rods are connected to the mobile platform and the fixed platform through six universal joints. It's necessary to keep the joints low-friction to guarantee the high accuracy during the measuring. As it's shown in Fig. 1, an encoder is fixed on each universal joint near the fixed platform, and a load cell is positioned near the universal joint near the mobile platform to measure the force acting on the mobile platform along the connecting rod. All the three connecting rods have the same set of sensors, thus there are totally three encoders and three load cells.

Fig. 1 The parallel mechanism



Theory [9] suggests that the closer the rotation axis is to the center of gravity of the measured rigid body and the mobile platform, the higher accuracy may get during the measuring. An adjusting mechanism, composed of dovetail guides, bevel gears and a motor, can help to adjust the position of the rotation axis conveniently so as to optimize the measurement of the inertia tensor of the measured rigid bodies. In Fig. 2, the adjusting mechanism is shown. Each universal joint near the fixed platform is fixed on the slider moving along the dovetail guide. The sliders, which are designed specifically, are driven by the screws. On one end of the screws there are bevel gears through which the screws are driven by the motor. Through this drive system, the rotation axis of the mobile platform can be adjusted easily. For example, when the big bevel gear is driven by the motor, the three small bevel gears along with the screws would be driven to rotate at the same time. And then the sliders would all move along the screws. One may easily notice that if the sliders all move away from the big bevel gear, then the rotation axis of the mobile platform would rise in the vertical direction. And if the sliders are driven to move close to the big bevel gear, the rotation axis would descend in the vertical direction. Thus the rotation axis can be changed easily to get closer to the center of gravity of the measured rigid bodies.

The measured rigid body is fixed on the mobile platform. Together with the mobile platform, the rigid body is driven deviate from the original equilibrium position by external driving or man-made offset. Without the external driving or man-made offset, the system would go into free motion. Based on the dynamic analysis of the free motion of the system, method for identifying the inertia tensor of the rigid body can be derived, refer to [9, 13]. The influence of friction during the free motion is low enough that can be ignored. The three load cells are used to

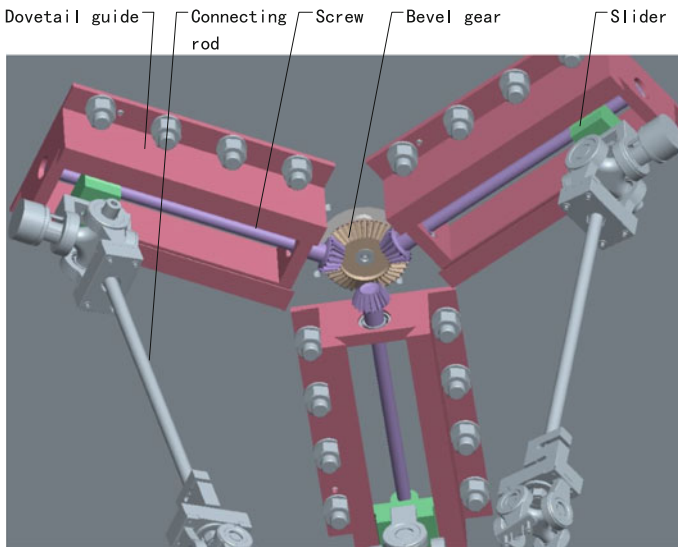


Fig. 2 The adjusting mechanism

measure the load along the connecting rods. And the three encoders are used to measure the rotation angles during the free motion. The angular velocities can be computed by deriving the rotation angles and the way to get the angular accelerations is the time derivation of the angular velocities. And the linear acceleration can also be computed using the kinematic analysis of the 3-URU parallel mechanism. Thus the kinematic and dynamic parameters can be computed through the three load cells and three encoders. Then the inertia parameters of the measured rigid bodies can be computed based on the dynamic analysis of its the free motion.

3 Kinematic Analysis

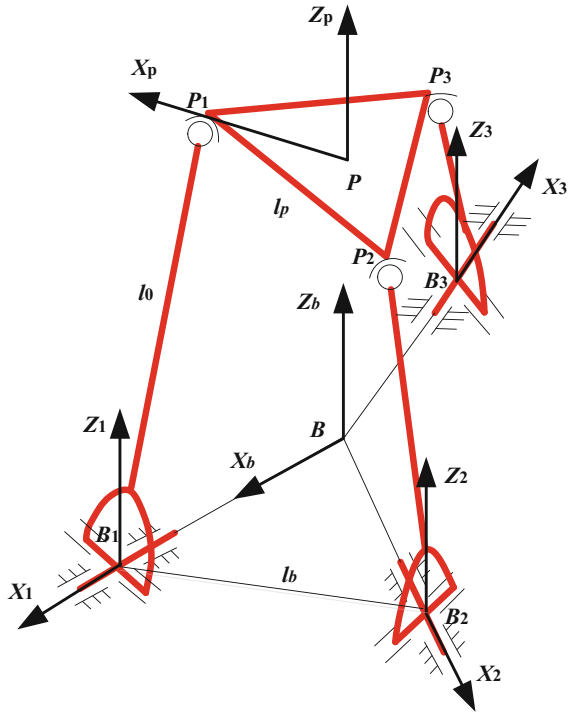
Let's consider the kinematics of the parallel mechanism. A kinematic diagram of the 3-URU parallel mechanism is shown in Fig. 3. The centers of the Hooke joints near the mobile platform are denoted as $P_i(i=1,2,3)$, which form an equilateral triangle. As to Hooke joints near the fixed platform, their centers are denoted as $B_i(i=1,2,3)$, which form an equilateral triangle. The mobile platform coordinate system $\sum_p: O_p - X_p Y_p Z_p$ is located on point P , the center of the three points $P_i(i=1,2,3)$. A base coordinate system $\sum_b: O_b - X_b Y_b Z_b$ is fixed on the center of the three points $B_i(i=1,2,3)$. Their X-axis directions point to P_1 or B_1 . In addition, located on the points $B_i(i=1,2,3)$, reference coordinate systems $\sum_i: O_i - X_i Y_i Z_i(i=1,2,3)$ are fixed on the Hooke joints. Each of the X-axis directions points to $B_i(i=1,2,3)$ from the origin of the coordinate system $\sum_b: O_b - X_b Y_b Z_b$, and their initial directions of Z-axis are perpendicular to the fixed platform, as it's shown in Fig. 3.

Some constants of the mechanism are: link lengths l_0, l_p and l_b (shown in Fig. 3). The initial coordinates of $P_i(i=1,2,3)$ in coordinate systems $\sum_i: O_i - X_i Y_i Z_i(i=1,2,3)$ are denoted as $p_i^i = \left(-\frac{\sqrt{3}}{3}(l_b - l_p), 0, \sqrt{l_0^2 - \frac{1}{3}(l_b - l_p)^2} \right)$ ($i=1,2,3$), the transformation matrices from coordinate system $\sum_b: O_b - X_b Y_b Z_b$ to $\sum_i: O_i - X_i Y_i Z_i(i=1,2,3)$ can be denoted separately as:

$$T_1^b = \begin{bmatrix} 1 & 0 & 0 & \frac{\sqrt{3}}{3}l_b \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_2^b = \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{6}l_b \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 & \frac{l_b}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^b = \begin{bmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{6}l_b \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 & -\frac{l_b}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 3 Kinematic diagram of 3-URU parallel mechanism



Consider such a motion: Hooke joint $B_i (i = 1, 2, 3)$ rotates around its X -axis, and then rotates around its Y -axis. The rotation angles are denoted as $\alpha_i (i = 1, 2, 3)$ and $\beta_i (i = 1, 2, 3)$ separately. After the motion, the initial reference coordinate systems $\sum_i : O_i - X_i Y_i Z_i (i = 1, 2, 3)$ become new ones $\sum'_i : O'_i - X'_i Y'_i Z'_i (i = 1, 2, 3)$, and the coordinates of $P_i (i = 1, 2, 3)$ with respect to the initial reference coordinate systems $\sum_i : O_i - X_i Y_i Z_i (i = 1, 2, 3)$ can be written as:

$$P_i^j = A_i^j P_i^i \tag{1}$$

where A_i^j is a rotation matrix written as:

$$A_i^j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix} \begin{bmatrix} \cos \beta_i & 0 & \sin \beta_i \\ 0 & 1 & 0 \\ -\sin \beta_i & 0 & \cos \beta_i \end{bmatrix} \tag{2}$$

Hence, the coordinates of $P_i (i = 1, 2, 3)$ with respect to $\sum_b : O_b - X_b Y_b Z_b$ can be written as:

$$P_i^b = T_i^b P_i^i \tag{3}$$

Based on the constants of the mechanism and the rotation angles $\alpha_i (i = 1, 2, 3)$ measured by the encoders, the angles $\beta_i (i = 1, 2, 3)$ can be calculated. Thus the location of the three Hooke joints near the mobile platform with respect to the base coordinate system $\Sigma_b : O_b - X_b Y_b Z_b$ can be calculated. For each of the three Hooke joint, the following relation can be written as:

$$P_1^b = P^b + A_p^b P_1^p \tag{4}$$

$$P_2^b = P^b + A_p^b P_2^p \tag{5}$$

$$P_3^b = P^b + A_p^b P_3^p \tag{6}$$

where P^b is the location of the point P in the base coordinate system Σ_b and $P_i^p (i = 1, 2, 3)$ the coordinate of P_i in Σ_p . A_p^b is the rotation matrix. By subtracting Eqs. (5) and (6) from Eq. (4), it can be simplified:

$$P_1^b - P_2^b = A_p^b (P_1^p - P_2^p) \tag{7}$$

$$P_1^b - P_3^b = A_p^b (P_1^p - P_3^p) \tag{8}$$

From Eqs. (7) to (8), the three Euler angles defining the rotation matrix A_p^b can be derived.

4 Simulation

4.1 Direct Kinematic Simulation

When the three Hooke joints on the fixed platform are driven to rotate around X_1 -axis, X_2 -axis and X_3 -axis respectively, the position of the mobile platform can be calculated, according to the direct kinematic analysis of the 3-URU parallel mechanism. In the direct kinematic simulation, the rotation angle around X_1 -axis is set to change linearly from 0° to 8° , and the other two are changed linearly from 0° to 12° and -6° respectively. The continuously-changing coordinates of the reference point P in $\Sigma_b : O_b - X_b Y_b Z_b$ are calculated, see Fig. 4. And the Euler angles defining the rotation matrix A_p^b (defining the angular displacement of the mobile platform) are shown in Fig. 5.

Fig. 4 Coordinates of point P in $\sum_b : O_b - X_b Y_b Z_b$

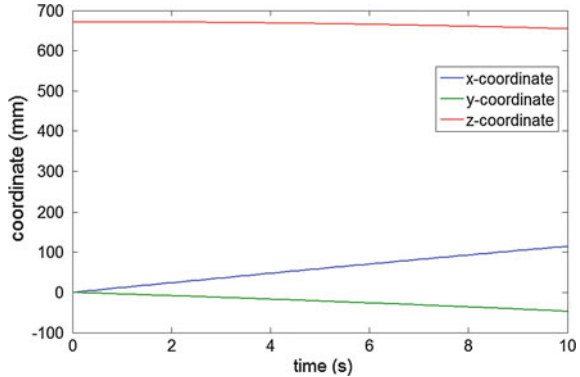
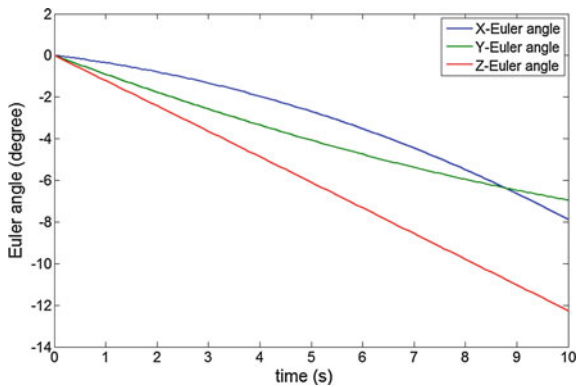


Fig. 5 The Euler angles



4.2 Dynamic Simulation

The kinematic and dynamic modeling of the parallel mechanism is accomplished in Adams, see Fig. 6. The dynamic model can be excited by a known initial condition. For example, the $\alpha_i (i = 1, 2, 3)$, defined above, are set as 4° , -6° and -7° . The rotation angles (Fig. 7) and angular velocities (Fig. 8) are measured during the free motion of the mobile platform. It's easy to notice that the quasi-frequency of rotating is near to 1 Hz. The forces acting on the system composed of the mobile platform and a rigid body (54.9 kg in total), are measured, see Fig. 9. The curves in Figs. 7 and 9 are smooth and the quasi-frequencies are low. This will make it easy to measure the rotation angles by encoders and the forces by load cells. The curves can also provide important reference for choosing the performance (range, resolution, etc.) of sensors. Three 16-bits encoders and load cells with ranges of 1000 N are used in the device. The accelerations and angular accelerations of the reference point P with respect to $\sum_b : O_b - X_b Y_b Z_b$ are shown in Figs. 10 and 11.

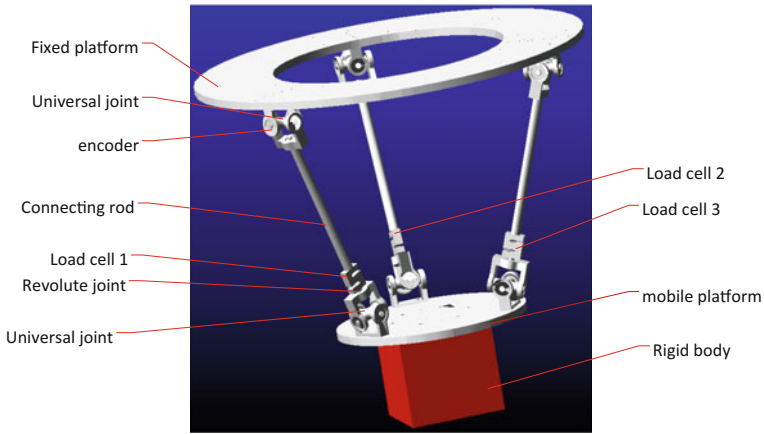


Fig. 6 The device model in Adams

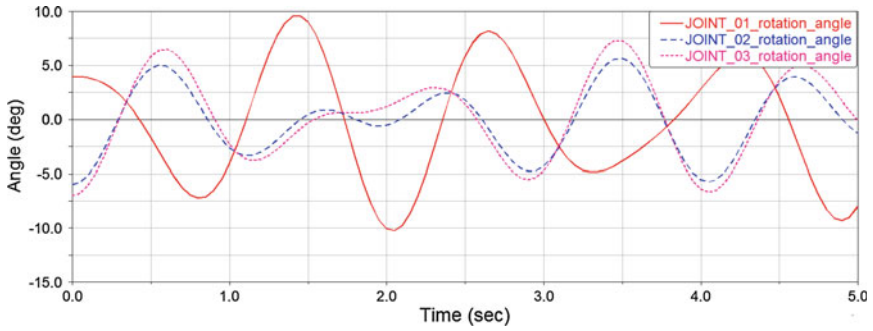


Fig. 7 Rotation angles

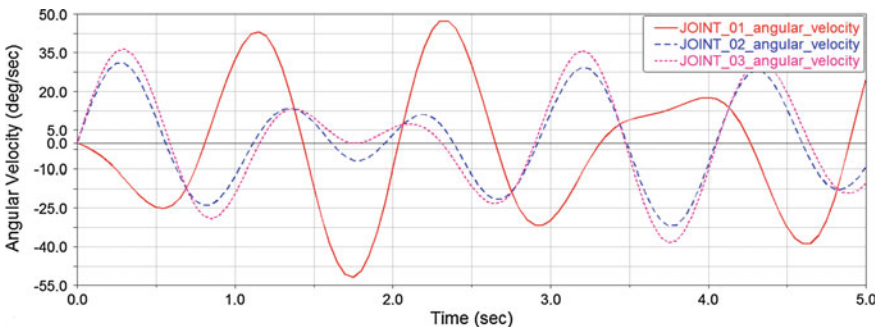


Fig. 8 Angular velocities

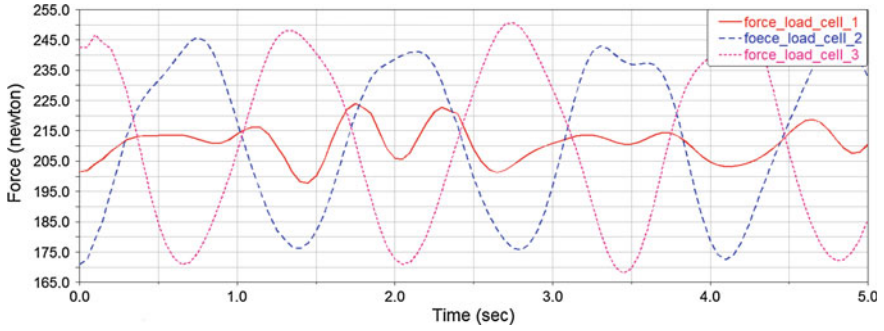


Fig. 9 The forces acting on the mobile platform

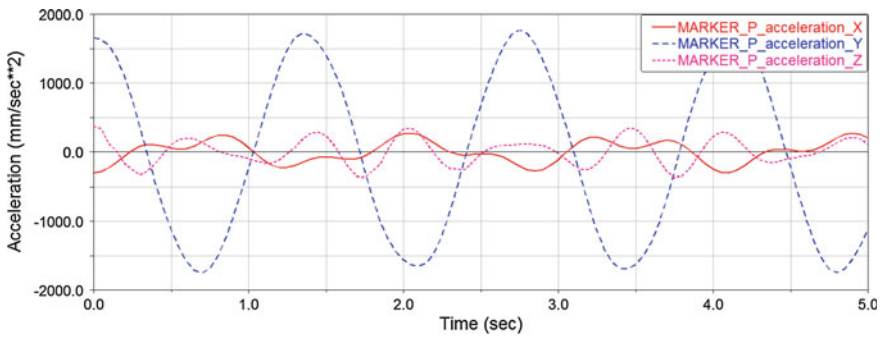


Fig. 10 Accelerations of point P in coordinate system B

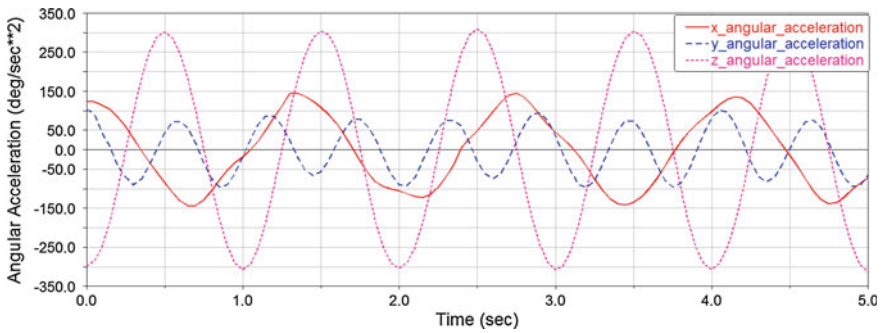


Fig. 11 Angular accelerations of point P in coordinate system B

5 Conclusion

In this paper, a 3-URU parallel mechanism, used for measuring the inertia properties of rigid bodies, has been presented. Based on the non-linear free motion of a multi-cable pendulum, the inertia tensors can be identified. Only three encoders and three load cells are needed. In the measuring device, an adjusting mechanism, composed of dovetail guides, bevel gears and a motor, is adopted, which can help to improve the measurement accuracy. Three 16-bits encoder are used for measuring the rotation angles which are extremely important for measuring the inertia parameters that must be measured accurately. And three load cells with a range of 1000 N can be used in the case that the weights of the rigid bodies are under 60 kg. Kinematic and dynamic analyses of the parallel mechanism are given through simulation and simulation results show that the system has low kinetic quasi-frequency. The system hardware is very simple and few sensors are needed, so the cost could be low. More heavy bodies will be measured and the higher accuracy will be acquired, if the sensor hardware performance is increased. An experimental device is under development.

References

1. Blokin-Mechtalin, Y.K., Bogdanov, V.V., Panchenko, I.N., et al.: Test rig for measuring the center of gravity coordinates and inertia tensor of spacecrafts: data acquisition and control system[J]. *Autom. Remote Control* **74**(74), 679–683 (2013)
2. Mastinu, G., Gobbi, M., Previati, G., et al.: Influence of vehicle inertia tensor and center of gravity location on road accident reconstruction. In: *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2011, pp 877–886
3. Gobbi, M., Mastinu, G., Previati, G.: The effect of mass properties on road accident reconstruction. *Int. J. Crashworthiness* **19**(1), 71–88 (2014)
4. Bacaro, M., Cianetti, F., Alvino, A.: Device for measuring the inertia properties of space payloads. *Mech. Mach. Theory* **74**(6), 134–153 (2014)
5. Hou, Z.C., Lu, Y.N., Lao, Y.X., et al.: A new trifilar pendulum approach to identify all inertia parameters of a rigid body or assembly. *Mech. Mach. Theory* **44**(6), 1270–1280 (2009)
6. Ou, Ma, Dang, H., Pham, K.: Simulation study of a robotics-based method for on-orbit identification of spacecraft inertia properties. *Sensors Syst. Space Appl.*, 655519–655519-17
7. Melnikov, V.G.: A new method for inertia tensor and center of gravity identification. *Nonlinear Anal.* **63**(s 5–7):e1377–e1382
8. Ringegni, P.L., Actis, M.D., Patanella, A.J.: Experimental technique for determining mass inertial properties of irregular shape bodies and mechanical assemblies. *Measurement* **29**(1), 63–75 (2001)
9. Gobbi, M., Mastinu, et al.: A method for measuring the inertia properties of rigid bodies. *Mech. Syst. Signal Process.* **25**(1), 305–318 (2011)
10. Bogdanov, V.V., Veselov, N.V., Panchenko, I.N., et al.: Test rig for measuring the object's mass, center of gravity coordinates and inertia tensor. *Autom. Remote Control* **72**(2), 425–434 (2011)

11. Tang, L., Shangguan, W.B.: An improved pendulum method for the determination of the center of gravity and inertia tensor for irregular-shaped bodies[J]. *Measurement* **44**(10), 1849–1859 (2011)
12. Almeida, R.A.B., Urgueira, A.P.V., Maia, N.M.M.: Further developments on the estimation of rigid body properties from experimental data. *Mech. Syst. Signal Process.* **24**(5), 1391–1408 (2010)
13. Schedlinski, C., Link, M.: Survey of current inertia parameter identification methods. *Mech. Syst. Signal Process.* **15**(1), 189–211 (2001)

Development of a Myoelectric Hand Incorporating a Residual Thumb for Transmetacarpal Amputees

**Yuta Murai, Suguru Hoshikawa, Shintaro Sakoda, Yoshiko Yabuki,
Masahiro Ishihara, Tatsuhiro Nakamura, Takehiko Takagi,
Shinichiro Takayama, Yinlai Jiang and Hiroshi Yokoi**

Abstract Restoring the hand functionality of partial amputees requires a myoelectric prosthetic hand, which is a robotic hand controlled by myoelectric signals from the skin surface and has the potential to restore human hand functionality. Most myoelectric hands have been developed for forearm amputees, while those for partial-hand amputees are few in number despite the higher numbers of the latter. Partial-hand amputees have limited hand functionality and cannot grasp and

Y. Murai (✉) · S. Hoshikawa · S. Sakoda · Y. Yabuki · T. Nakamura · Y. Jiang (✉) ·
H. Yokoi (✉)

The University of Electro-Communications, Chofu, Japan
e-mail: murai@hi.mce.uec.ac.jp

Y. Jiang
e-mail: jiang@hi.mce.uec.ac.jp

H. Yokoi
e-mail: yokoi@hi.mce.uec.ac.jp

S. Hoshikawa
e-mail: hoshikawa@hi.mce.uec.ac.jp

S. Sakoda
e-mail: sakoda@hi.mce.uec.ac.jp

Y. Yabuki
e-mail: yabuki@hi.mce.uec.ac.jp

T. Nakamura
e-mail: nakamura@hi.mce.uec.ac.jp

M. Ishihara
Brace on R Nagoya, Higashi-Nagoya, Japan
e-mail: ishihara@braceon.com

T. Takagi
Department of Medicine, Tokai University, Isehara, Japan
e-mail: ttkg7@me.com

S. Takayama
The National Center for Child Health and Development, Setagaya, Japan
e-mail: shintkym10@mac.com

manipulate various objects in a manner comparable to individuals with a healthy hand; thus, they require a myoelectric hand. In this study, design issues were identified, and three-dimensional computer-aided design was used to propose an integrated skeleton and housing with a supporting socket. A passive thumb mechanism with motion in the remaining part of the hand was developed where the motion control system is based on the amputee's muscles. An amputation system is proposed comprising mixed metacarpal and center-part cuts. A prototype was constructed, and its gripping functionality was evaluated. The results demonstrated an enhanced gripping performance compared to the non-use of prosthetics, which attests to the viability and effectiveness of the system.

Keywords Myoelectric prosthetic hand • EMG signal • Partial amputation • Pattern classification

1 Introduction

Recently, myoelectric hands designed for forearm amputees have seen increased use. These devices analyze weak electrical signals from muscles near the limb stump by using electromyography (EMG), and a motorized hand fitted to the fingers controls and attempts to replicate the grasping and manipulation capabilities of the human hand. It is not possible to perform multiple grasping and manipulation movements when the EMG system analyzes muscle signals individually.

Traditional myoelectric devices are primarily intended for use by amputees with amputations above the wrist joint, which represent the largest proportion of amputations in Japan [1]. There is also a demand for myoelectric prosthetic hands to be used by partial amputees. These must be able to manipulate various types of objects. Such prosthetic hands can compensate for the reduced hand functionality of partial hand amputees.

Previous researchers developed prosthetic hands for partial hand amputees that drive only the finger MP joint or drive each finger by using actuators. While the former provides a strong torque and low DOF, the latter provides a high DOF and low torque. Conventional products were developed by OttoBock [2] in Germany and Weir et al. [3] at Colorado University in the USA. Later, the Vincent system was developed by Schulz [4] in Germany, and Touch Bionics was developed in the USA [5]. These products use a normal controller with either a simple EMG threshold-type switch or mechanical switch to control the MP joint. With these conventional controllers, because the motion of the remaining thumb disturbs the MP joint motor control, manipulating objects precisely/stably by using detailed movement of the remaining thumb is difficult.

This paper proposes a myoelectric hand system for partial hand amputees and presents the experimental results on the grasping and manipulating functionality of the design when fitted to a partial hand amputee.

2 Technical and Design Challenges

2.1 Aspects of Partial Amputation

The muscle activities are divided into two kinds; the muscle disarticulates in the middle of hand which cannot move finger joints, and the muscle remains connected between hand joints and muscles. The former muscle cannot move finger joints and should use actuator to compensate for hand functions. The latter muscle should be used to move finger joints for grasping objects. EMGs from the two kinds of muscles are defined as 'disarticulating EMG' and 'residual EMG' respectively.

This paper focuses on the patterns of the remaining thumb carpometacarpal (CM) and disarticulation above the fingers' metacarpophalangeal (MP) joint.

2.2 Hand Mechanism Design Challenges for Remaining Parts

The hand's function is to hold and release objects. In order for the hand mechanism to hold and manipulate various objects for activities of daily living (ADL), many actuators and link mechanisms in many forms are needed to achieve the same hold and grip force as a human hand. A tradeoff must be made between increasing the number of actuators improves the control over the degrees of freedom (DOF) and torque but also increases the weight and affects the appearance of the device.

The prosthesis must be of a practical size and weight. The number and torque of the actuators includes another tradeoff; increasing the number of actuators requires downscaling the actuator size and torque, while increasing the torque requires decreasing the number of actuators.

The required torque is around 30 Nm as calculated from the average gripping force [6]. Therefore, determining the suitable hand size and weight is important, and the holding torque needs to be enough to hold objects of around 1–2 kg for ADL.

The prosthesis needs to match the size of the unaffected side of the subject's hand. The weight of the prosthesis (including the cosmetic globe) should be around 570 g [6]; the hand weight is considered to be around 1 % of the average body weight.

2.3 Socket Design Challenges for Remaining Parts

The socket design must incorporate the remaining thumb and not interfere with its range of motion (ROM). Likewise, the socket must not interfere with the ROM of the wrist. The stump shapes of partial amputees are more complex than those of forearm amputees. Therefore, the socket must be adapted to the stump surface and securely connected.

2.4 *Electromyography Design Challenges*

In order to use EMG to control the prosthetic hand, the EMG signal of the remaining muscle functions of the transmetacarpal amputee should be divided into two kinds. EMG signals detected from muscles that can drive the thumb are called residual EMG, and signals detected from muscles that are not connected to other joints are called disarticulating EMG.

The thumb needs to be moved on its own. The free muscle EMG needs to be obtained to control the prosthetic hand. Actuator malfunction due to the user's residual EMG must be avoided.

The pattern classification system of the proposed myoelectric prosthetic hand for a partial amputee needs to extract features of both the residual EMG and disarticulating EMG. The disarticulating EMG signal is applied to control the actuator. The residual EMG (to move the thumb) should not influence the disarticulating EMG.

3 **Developed System**

In this study, a myoelectric hand prosthetic system was developed (Fig. 1) in which the hand component can realize three motions (grasping, pinching, and lateral pinching) by combining motorized components and a link mechanism operated by the thumb. The thumb motions of both flexion and extension can be achieved by using the amputee's remaining CM thumb joint for adduction and abduction [7].

The proposed prosthesis mechanism was designed and produced by using three-dimensional computer-aided design (3DCAD) (SolidWorks, Dassault Systems SolidWorks Corp.) and a 3D printer (uPrintSE, Stratasys Ltd.). A 3D scanner was used to convert the stump shape into digital data so that the inner shape of the socket could be designed and preserve the practical functions of the remaining part of the hand. A flat base for connection with the socket was printed. The socket part was mainly printed by using acrylonitrile butadiene styrene (ABS) resin; however, this material exhibited poor mechanical strength for the thumb joint, which can be problematic during 3D printing and processing and during assembly of the hand component and socket. Therefore, only the thumb joint was made of aluminum.

The proposed system implements individual adaptive control to reduce the error rate from residual EMG of the hand. Furthermore, the thermoplastic elastomeric glove shown in Fig. 1 was needed to reduce the movement obstruction of both the prosthetic hand and the remaining part of the original hand. The use of this glove improved the friction and elasticity of the prosthesis during contact with objects (Fig. 2).

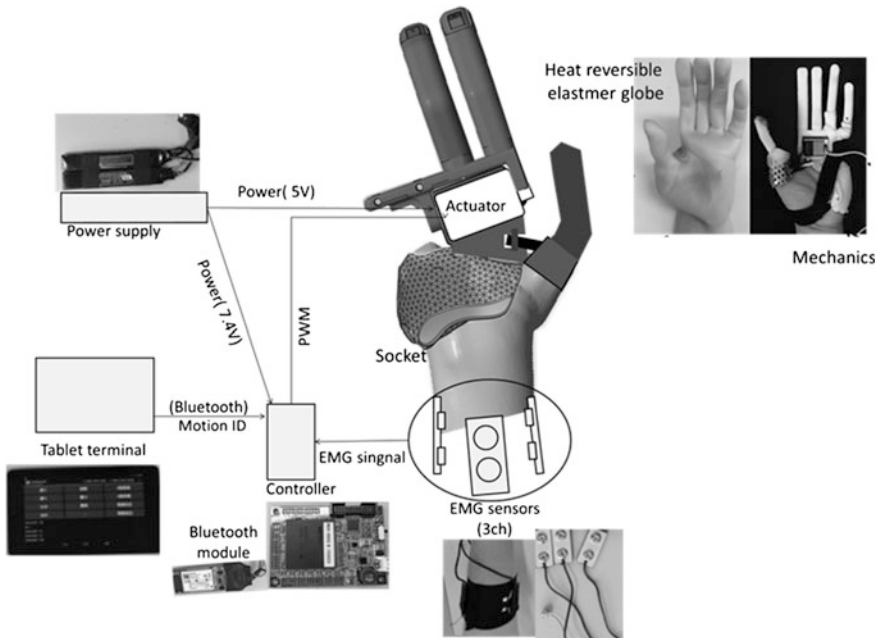


Fig. 1 Overview of developed system

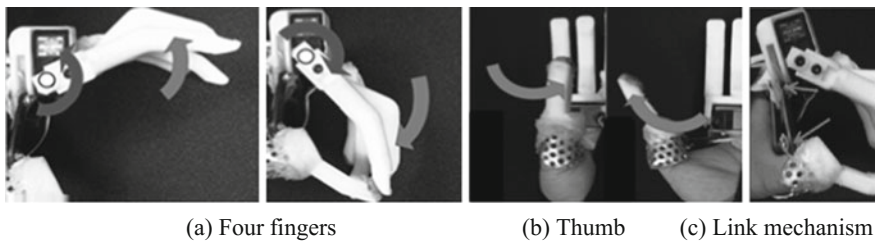


Fig. 2 Finger movement: (left) four fingers, (right) thumb

3.1 Hand Grasping and Manipulation Mechanics

In order for the system to be of a practical size and weight, a single actuator was selected to move four fingers simultaneously. Hoshikawa et al. [7] designed a 2-DOF myoelectric prosthetic hand intended to yield the minimum necessary grip manipulation performance with the lowest possible number of electric components.

This was achieved by using MP joints for the first, middle, third, and little fingers and a CM joint for the thumb (Fig. 3a, b). The latter uses the greatest range of grasping motions (grasping, pinching, lateral pinching), which are involved in 85 % of ADL (Fig. 3) [7].

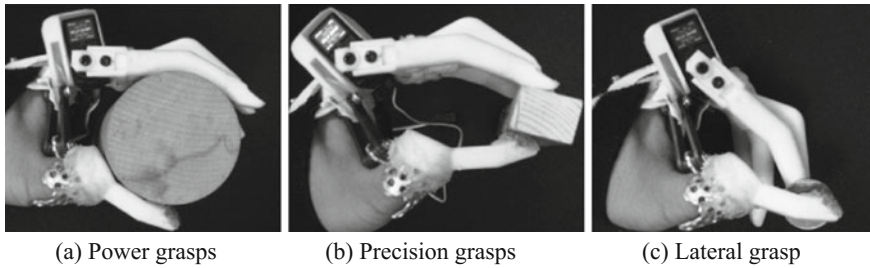


Fig. 3 Three types of grasping motions

If the subject's hand does not retain the metacarpal bone or thumb MP joint, the remaining muscles cannot move fully.

Thus, this prosthesis compensates for the thumb by extending from the stump. However, precise pinching with this prosthesis is difficult in actuality because of the unstable movement of the thumb.

The thumb DOF in the adduction and abduction directions is limited. The thumb DOF is limited to one direction for the opposition and collateral, which makes it easier to realize precise pinching by contacting the tips of the thumb, index finger, and middle finger than when the DOF of the thumb is not limited.

To limit the DOF of the thumb's movement, thumb prosthesis was developed with a link mechanism that is limited to the directions of the thumb opposition and collateral (Fig. 2b). The two circles in Fig. 2c show rotational DOF. Thus, this mechanism can perform movements in the directions of the opposition and collateral. When a strong force is added to the adduction and abduction directions, the thumb prosthesis connects to the socket, which bends and releases the force.

3.2 *Socket and Electrical Component*

The developed system satisfies the following two requirements: the socket must be designed to fit the amputee's stump, and the thumb and wrist ROMs are preserved. Therefore, edits to the socket were made by using 3DCAD and to the output socket by using a 3D printer. Editing the socket by using 3DCAD requires a plaster mold of the hand stump and scanning by a digital scanner in order to measure the thumb CM and wrist ROM.

The connection between the stump and socket is used to connect the socket back to wrist, and a surface fastener is used to connect below the little finger to the wrist.

The thumb CM and wrist ROM are determined by measuring the angle range from the relaxed position to the flexion direction and the angle range from the relaxed position to the extension direction. The ROM of the wrist was 40° for the former and 24° for the latter.

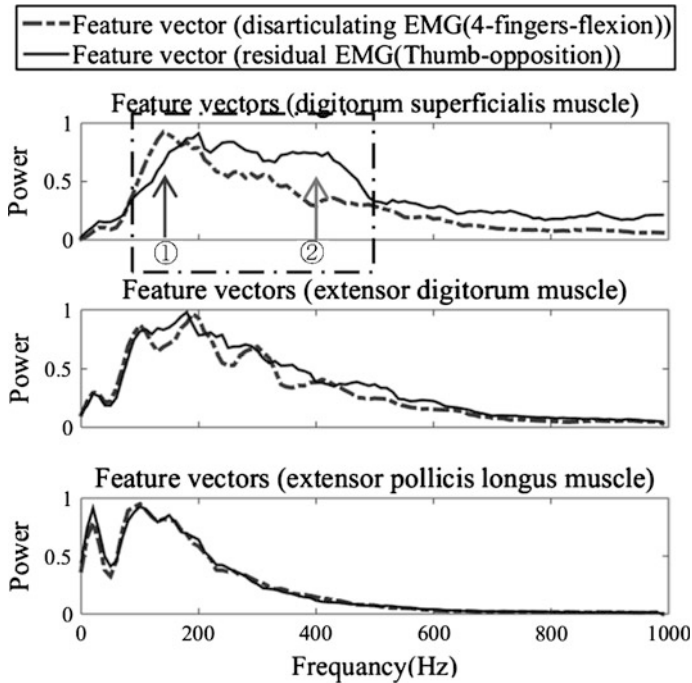


Fig. 4 Feature vectors

3.3 Control Method that Separates Both Types of EMGs

In order to distinguish residual EMG and disarticulating EMG, their features were computed and analyzed by using three-layer artificial neural networks based a discrimination system with a learning function.

The discrimination system uses feature vector data subjected to fast Fourier transform (FFT) and detected from three-channel EMG sensors of the forearm as input for the neural network [8]. The system classifies the features of both residual EMG and disarticulating EMG. This discrimination system classifies the results of the four-finger movement as flexion, extension, or relaxation.

Figure 4 shows the feature vectors when the subject tried to control the servo-motor of the four fingers and move the residual thumb. These feature vectors were derived by using 100 data points from FFT of EMG signals sampled at 2000 Hz from the three-channel EMG sensors of the forearm.

Figures 4a–c show the feature vectors detected from the flexor digitorum superficialis muscle, extensor digitorum muscle, and extensor pollicis longus muscle, respectively.

Lines 1 and 2 in Fig. 4a show that the characteristics of the distribution power spectrum differed for the detected residual EMG and disarticulating EMG. Thus, both types of EMG can be separated and classified.

Thus, to control the servomotor for the four fingers, only the disarticulating EMG is measured. With this method, this prosthesis can make three types of grasping postures by using the EMG signal detected from the forearm, as described in Sect. 3.1. This method has one problem; when the subject intends to move the four fingers and thumb at the same time, the three-layer artificial neural network cannot distinguish between both EMG signals.

3.4 Cosmetic Glove for Hand Appearance

The cosmetic glove is necessary for the aesthetic appearance of the prosthetic hand. However, the cosmetic glove usually causes resistance in the mechanical system. To realize low energy consumption for the mechanical system, the joint motions for extension and flexion of the fingers need to have low friction. A hyper elastic cosmetic glove with low energy consumption was developed from a styrene thermoplastic elastomer composite [9]. This glove realizes 1300 % elasticity and a realistic appearance with wrinkles and fingerprints through the use of the injection mold technique. The prosthetic hand uses a glove produced by flip-molding the subject's hand and realizes both low resistance to motor movement and low conflict with the thumb motion.

3.5 Development and Construction of Proposed System

Figures 1, 2, and 3 show the appearance of the prosthetic hand. Figures 2 and 3 show it with no globe, and Fig. 1 shows it with the covering globe. The actuator of the developed system employs the Hyperion Atlas High Voltage servomotor DH16-FTD, which has a dynamic torque of 12.0 kg cm and holding torque of 6.0 kg cm. The developed system was measured to have a 12 N pinching force, 22.5 N lateral pinching forces, and 15.6 N grasping force.

These specifications suggest that this prosthesis can hold and keep an object that is 1–1.5 kg. The socket and finger skeleton are made of ABS. The specifications of hand the length and hand height met the required size. Therefore, the hand thickness did not meet the required size (15 mm thicker than required) and needed to be optimized to place actuator. The signal processing part of this system consists of a prosthesis controller with individual adaptive control [8]. This control system uses a three-channel EMG sensor attached to the forearm. Table 1 presents the other specifications.

Table 1 Specifications of developed system

Item	Specifications
Weight (g)	193 (only hand and globe)
Control DOF	3 (four fingers)
Mechanical DOF	1 (four fingers), 2 (thumb CM joint)
Battery	Lipo-redline 1400 mAh, 2 S, 7.4 V
Controller	Renesas SuperH SH72544r
EMG sensor	Custom-made
Electrodes	GE Health Care, Bioload 45,352 V

4 Experiment and Evaluation

4.1 Evaluation of Prosthetic Hand Functionality: Picking up and Placing Objects Test

The experimental method was employed to evaluate the functionality of the prosthetic hand with regard to its performance at picking up and placing objects from two areas, as shown in Fig. 5. Two areas with dimensions of 15 × 15 cm² were defined and labeled as A and B. The closest borders of A and B were 50 cm apart.

The experiment was designed to evaluate the performance of a subject (a partial-hand amputee) at carrying an object beyond a urethane obstacle block of approximately 10 cm in height between A and B.

The subject stood at equal distances from A and B. The first stage of the experiment examined the subject’s ability to grasp and lift objects placed in A. The subject was asked to move objects from A by using a continuous grasping motion and place them in B. In order for the subject to hold the grasping object in A or B, the prosthetic hand or stump needed to be used to push a switch near the area. The user pushed the switch with the same hand grasping the object, and the object was

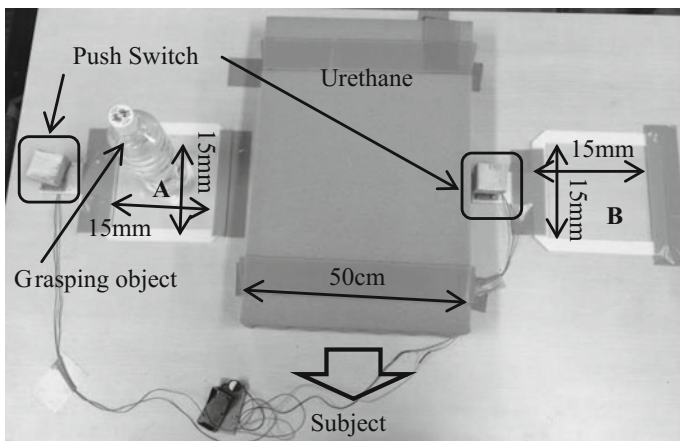


Fig. 5 Experimental system for picking up and placing objects

Table 2 Test objects for the grasping experiment

Object	Grip position	Material	Size (mm)	Wt. (g)
Plastic bottle	Surface (grasping)/Bottle cap (pinching)/Unspecified	Plastic	$\Phi 65.0 \times 277.0$ (cap: $\Phi 30$)	580
Paper packet	Surface (grasping)/Top (pinching)/Unspecified	Paper	$70.0 \times 70.0 \times 174.$	541
Pen	Surface (pinching)	Plastic	15.0 (bottom) $\times 150 \times 20.5$ (top)	24
Card	Unspecified	Plastic	$54.0 \times 85.5 \times 0.75$	5
Rice bowl	Surface (grasping)/Edge (pinching)/Unspecified	Ceramic	120.0×56.0 (bottom 45)	133
Ball	Surface (grasping/pinching)	Urethane	$\Phi 70.0$	35
Glass ball	Surface (pinching)	Glass	$\Phi 17.0$	5
Wooden chopsticks	Surface (pinching)	Wood	7.8 (bottom) $\times 203.5 \times 13.0$ (top)	5

lifted and grasped before being moved and put down in the other area. The subject was asked to grasp, lift, and place the object as many times as possible in 30 s. This task was performed three times, and the averaged results were evaluated.

Successful attempts (i.e., the object was grasped and placed in the intended position) were counted along with the failed attempts. Three types of errors were defined and recorded: (1) error while lifting the object, (2) error during movement to another area, and (3) error during placement in the square area.

4.2 Test Object and Hand Selection

Test objects with various shapes, sizes, and weights were selected from items that are widely used in ADL. Table 2 presents the objects selected for the grasping test. The experiments examined the subject's performance both with and without the proposed myoelectric hand (i.e., using the partially amputated hand in the latter case).

4.3 Evaluation Method

The user's ability to pick up and move an object from one area to another by using a continuous grasping motion was examined to evaluate the grasping and manipulation performance. The number of successful completions of the test was compared against the number of trials to determine the grasping success rate (1). The number of trials included all successes and failures (2).

$$\text{Success rate} = \frac{(\text{Number of successful attempts})}{(\text{Number of trials})} * 100 \quad (1)$$

$$\text{Number of trials} = \text{Number of successful attempts} + \text{Number of failures} \quad (2)$$

The subject was asked to pick up and position objects while using the proposed prosthesis. The subject performed the experiment on days 1–3. The effect of experience on the grasping success rate was then considered. The number of successful attempts clearly reflected the subject's ability to grasp the grasping object, and the grasping success rate indicated the gripping stability and ease.

4.4 Evaluation Results

Figure 6 shows the grasping success rate and number of successful attempts with the partially amputated hand and proposed system on days 1–3.

Minimum functionality close to that of the human hand was confirmed to be obtained; using the developed prosthesis system allowed four additional object types to be grasped.

On day 1, the partially amputated hand tends to achieve a higher success rate and more successes than developed prosthesis. The success rate and number of successes tended to increase as the subject performed the pick and place task each day.

When using the developed prosthesis continuously, the subject developed new grasping functions with no deterioration in the success rate and number of successes relative to the remaining grasping functions with the partially amputated hand.

Hence, the results confirmed that the proposed system increases the grasping and manipulation performance of a partial-hand amputee.

5 Discussion

The subject was unsuccessful in his attempts to pick up the plastic card and paper packet. This seems to have been due to the prosthetic finger shape, which did not allow thin items to be picked up because the fingers lacked the appropriate design and grip force. This can be overcome by adding nails to the fingertips, changing the hand-component design, and improving the motor force.

The number of successful grasping attempts and therefore the success rate was lower with the proposed system than the partially amputated hand in the case of the rice bowl (which required pinching the edge), the plastic bottle caps, the wooden chopsticks, and the glass ball. This seems to have been due to the subject having hand grasping functionality that made it easier for him to grasp such items with the partially amputated hand than with the myoelectric hand. This problem can be

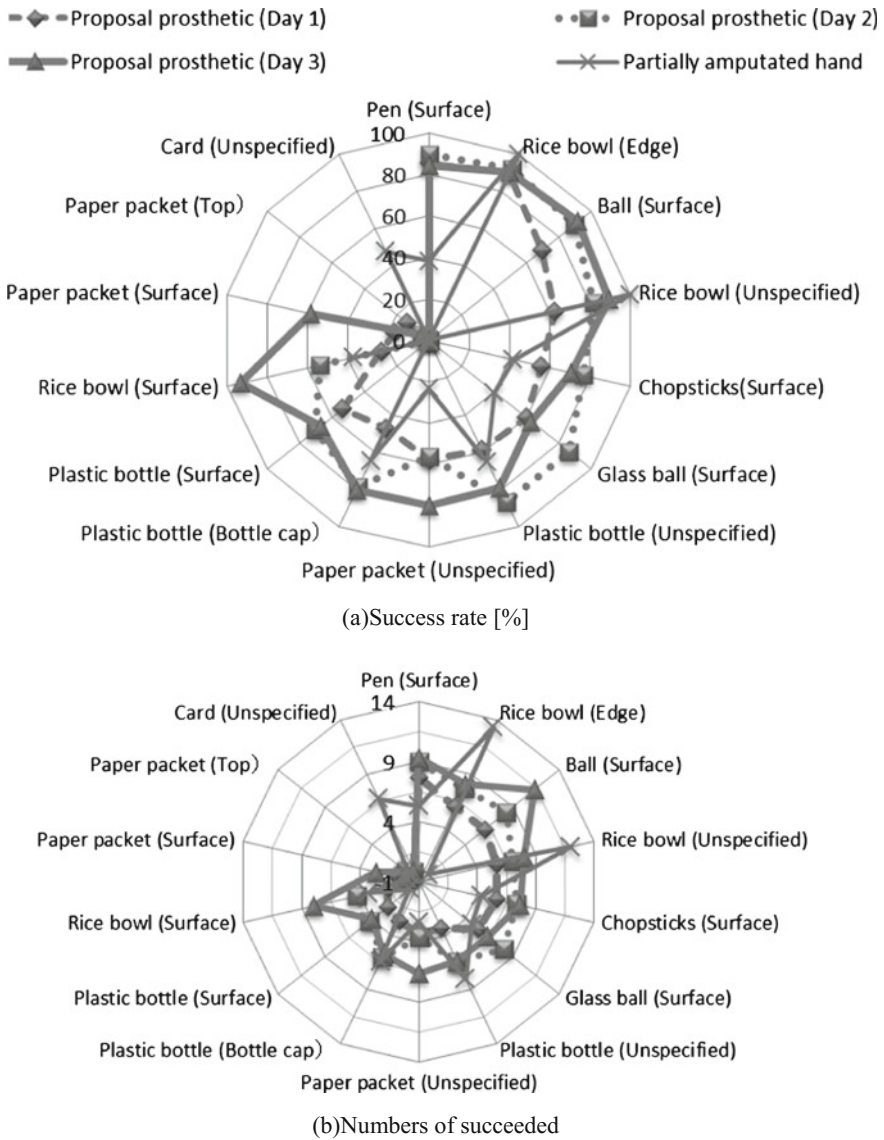


Fig. 6 Grasping success rates and number of successful attempts with the partially amputated hand and proposed system: days 1–3

overcome by installing a passive mechanism that uses the movement from the remaining part of the thumb and other parts. The partially amputated hand had a higher success rate and number of successes than the developed prosthesis for the rice bowl (edge), wooden chopsticks, plastic bottle cap, and glass ball. Moreover, the changes in the success rate and number of successes were not always related

every day. These movements to hold the object involved using the opposition thumb and little finger. With this movement as the holding mechanism, the developed prosthesis could gain an equal or better success rate and number of successes compared with the partially amputated hand.

The prosthetic hand of the proposed system was confirmed to meet the appearance requirements described in Sect. 2. The expanding servomotor is problematic and creates an unnatural feeling. One improvement is to make the servomotor smaller because this system design was found to be ineffective. The design of this system was then changed to output a pinching force. The servomotor adjusts the position and angle; it can reduce the expanding volume of the globe.

6 Conclusion

A myoelectric prosthetic hand system was developed for partial-hand amputees that can restore grasping and manipulation functionality. The proposed system distinguishes between the residual EMG of the thumb and disarticulating EMG by using adaptive pattern classification. The disarticulating EMG signal is applied to control the prosthetic hand, and the residual EMG from the thumb movement is prevented from influencing the disarticulating EMG. The overall performance of the developed device was then evaluated. Based on the experimental results for a task of picking up and positioning objects, the prosthesis was confirmed to strongly enhance the grasping and manipulation functionality compared to the case with no prosthesis.

Acknowledgements This research was partially supported by “A-step” from the Japan Agency for Medical Research and Development (AMED) and the Takahashi Industrial and Economic Research Foundation.

References

1. Takagi, T.: Treatment guideline making of a myoelectric artificial arm based on the individuality adaptation type information processing to an inborn transverse formation obstacle (forearm loss and the upper arm loss), welfare labor scientific research fund subsidy obstinacy, 2011 (in Japanese)
2. OttoBock, Transcarpal Hand, 2014. http://professionals.ottobockus.com/cps/rde/xchg/ob_us_en/hs.xsl/6909.html
3. Weir, R.F., Grahn, E.C., Duff, S.J.: A new externally powered, myoelectrically controlled prosthesis for persons with partial-hand amputations at the metacarpals. *JPO Online Library* **13–2**, 26–31 (2001)
4. Vincent Systems, Vincent Partial, 2013. <http://handprothese.de/vincent-finger>
5. Touch Bionics. i-digit pro, 2013. <http://www.touchbionics.com/products/active-prostheses/i-limb-digits>

6. Human Characteristics Database, National Institute of Technology and Evaluation, <http://www.nite.go.jp/>, 2003 (in Japanese)
7. Hoshikawa, S., Jiang, Y., Kato, R., Morishita, S., Nakamura, T., Yabuki, Y., Yokoi, H.: Structure design for a two-DoF myoelectric prosthetic hand to realize basic hand functions in ADLs. presented at the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC15), Milano, Italy, August 2015
8. Yokoi, H., Yabuki, Y., Seki, T., Nakamura, T., Morishita, S., Kato, R., Takagi, T., Takayama, S.: Prosthesis controlled by EMG signals. *J. Soc. Biomech.* **38**, 39–46 (2014). (in Japanese)
9. Yabuki, Y., Hoshikawa, S., Young, X., Jing, X., Nakamura, T., Tanahashi, K., Ishihara, M., Seki, T., Kato, R., Yokoi, H.: Development cosmetic globe for myoelectric prosthetic arm made of elastomer gel. In: 30th The Japanese Society of Prosthetics and Orthotics Conference Collection, p. 151, 2014 (in Japanese)

Part XVII
Robotics for ITER Remote Handling

Structural Design of Multi-joint Foldable Robot Manipulator for Remote Inspection in Experimental Advanced Superconducting Tokamak (EAST)

Baoyuan Wu, Weibin Guo, Yi Liu, Qiang Zhang, Ling Zhou, Qingquan Yan, Zhong Zheng and Zengfu Wang

Abstract The tele-operated in-vessel inspection system is required for maintenance in nuclear fusion environment. A Multi-joint Foldable Robot (MFR) is designed for Experimental Advanced Superconducting Tokamak (EAST). The whole MFR system has been designed and implemented. The motion control experiments show that the design is reasonable and effective. The MFR system developed can be used to perform inspection task in a nuclear environment.

Keywords Experimental advanced superconducting tokamak (EAST) · In-vessel inspection · Multi-joint foldable robot · Kinematic analysis

1 Introduction

Controlled nuclear fusion can be one of the most promising solutions to the global energy crisis. To achieve sustained nuclear fusion reaction, the conditions are very harsh, such as high temperature, high vacuum, and strong magnetic field. Further-more, Plasma Facing Components (PFCs) in Experimental Advanced Superconducting Tokamak (EAST) [1] consist of beryllium. Thus Lots of neutrons and gamma rays would be generated during plasma operation, and meanwhile, the in-vessel components can be activated and polluted. Moreover, the fuel tritium will permeate into PFCs or attach to materials surface. And the dust can spread into air

B. Wu · W. Guo (✉) · Y. Liu · Q. Zhang · L. Zhou · Q. Yan · Z. Zheng · Z. Wang (✉)
Institute of Intelligent Machines (IIM), Hefei Institutes of Physical Science (HIPS),
Chinese Academy of Sciences (CAS), Hefei 230031, Anhui, China
e-mail: wbguo@iim.ac.cn

Z. Wang
e-mail: zfwang@iim.ac.cn

Q. Zhang · Z. Wang
Department of Automation, University of Science and Technology of China,
Hefei 230026, Anhui, China

easily. Therefore, to prevent the pollution and radiation from harming operators, Remote Handling (RH) technology [2] is adopted for in-vessel inspecting and handling.

In-vessel inspection using robot manipulator in cantilever form is useful for operators understanding the running state and making maintenance plan. In particular, it is worth mentioning that this solution can be contactless, and pollution-free with the VV inner surroundings. And also, operators' safety can be ensured by remote monitoring and control. The maximum radius of the Vacuum Vessel (VV) outer border is about 2800 mm. The maximum distance between the inner borders of the VV is about 1500 mm. Therefore, the required robot manipulator length for inspecting a whole circle of the VV is about 13000 mm.

Currently, the research on the Long cantilever robot in the nuclear fusion surroundings is little. A remote handling system including two robot arms to complete maintenance and installation work is developed by JET [3]. Gargiulo et al. [4] developed an inspection robot named articulated inspection arm (AIA) and completed inspection of the vessel under UHV conditions. Chen Weidong et al. [5, 6] also developed a type of the flexible in-vessel inspection system for the plasma facing components (PFCs) in Shanghai Jiao Tong University. Our groups in the Hefei Institutes of Physical Science (HIPS), Chinese Academy of Sciences (CAS) have already did many pre-research [7]. Comparing with those research, based on the previous prototype we developed, we developed a new generation of multi-joint flexible in-vessel inspection robot system with Foldable joint feature. This structure has a obvious advantage on self lower volume cost, but larger efficient workspace.

2 Mechanism Scheme for Multi-joint Foldable Robot (MFR)

2.1 Task Requirement and System Configuration

In current stage, the Multi-joint Foldable Robot (MFR) is designed for the VV inspection and visual reconstruction. The workspace of the MFR should be almost cover the whole inner space of the VV. The MFR is required to carry multiple end effectors, as vision unit, or tools. As in Fig. 1, the whole system mainly consists of the Robot sys., the Vision sys., and the Monitor sys.. And the Robot sys. roughly consists of the Multi-joint Foldable Robot (MFR), the Mobile Platform (MP), and the Handling Trolley (HT). Also, the Vision sys. primarily consists of the Fiber Optic Camera (FOC), and the Fiber Optic Cable Reeling system (FOCR) etc. Whereas, the Monitor sys. is mainly composed of the Robot Control Platform (RCP), and the 3-D Display Control Station (DCS). The Handling Trolley is used for transporting the whole Robot system conveniently. The FOCR and the MFR is equated on the MP for displacement. The FOC is fixed on the MFR for guiding the MFR movement, and visual reconstruction. The FOCR is used for the fiber cable

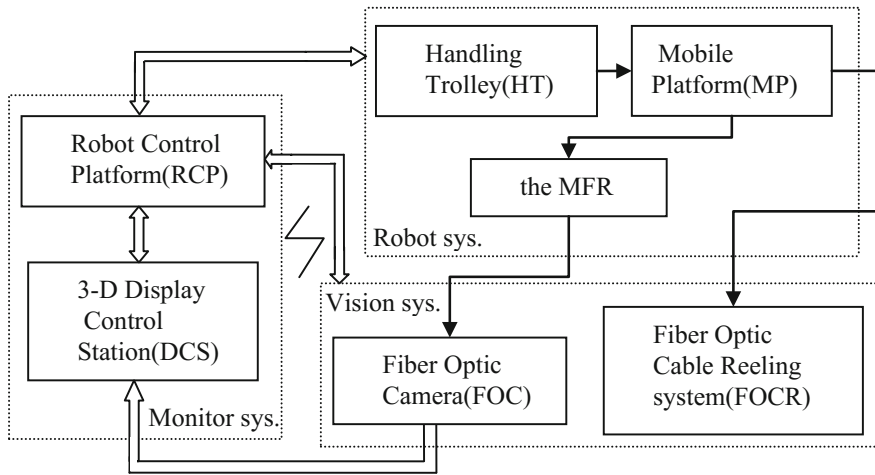


Fig. 1 System Configuration for MFR

reeling and conservation. The Robot control and sensor data is transmitted between the Robot sys. and the RCP. The Vision sys. control, image and sensor data is transmitted between the Vision sys. and the RCP. The FOC image data is feedback to the DCS for visual understanding and reconstruction. The image, command and sensor data are shared and transmitted between the RCP and the DCS. The data is transferred between the Robot sys. the Vision sys. and the Monitor sys. via wireless communication.

The required task motion is listed as follows. The MFR is preserved in the pipe of the Vacuum Vessel Port, when the fusion reaction is in progress. When inspection and maintenance task is scheduled, the MFR unfolds for the Planning trajectory. If necessary, the mobile platform will carry the whole MFR for a further position. The MFR mainly consists of the foldable robot and the mobile platform.

2.2 Mechanism Design for Multi-joint Foldable Robot (MFR)

Based on the task motion, the MFR and the mobile Platform are designed. The chosen architecture of the whole system is a modular design formed by a mobile platform and a multi-joint Foldable robot manipulator with inspection tool installed on it. The linear propulsion device with 1 DOF, which is formed by the car and the guide rail, is a prismatic mechanism. It can provide large and high precision linear displacement and deploy the multi-joint manipulator further.

Multi-joint mechanical arm involves two generation of experimental prototype,

- (1) telecommunication and power Supply line bunches of motor, encoder of each joint can pass through the inner of steel corrugated pipe and mechanical arm. The mechanical arm is concise to avoid the joint position of each arm, which does not interfere motion of each joint. And thus the design can also protect and insulate against heat from the inner of TOKAMAK.
- (2) Each arm is connected to the joint by adopting thread and thimble. It has improved the whole rigidity, integrated level of systematic assembles and it obviously reduces the ptosic amount of the mechanical arm.
- (3) lighter titanium alloy material has been adopted as the head attachment of corrugated pipe, which reduces the weight of each arm effectively and it also contributes to reducing the mechanical arm whole flaggingly too.

Due to the large operational range required and the limited size of the penetration hole, with respect to the kinematics, the robot system using limited numbers of actuated joints and links with small outer tube diameter was selected. In order to minimize the size of actuators and the link deflection, the size and weight of the robot should be strongly reduced. When considering the parameters of EAST VV and the operational range of the robot, the minimum length of the robot in full extension is decided. Considering the easier solution of the kinematics and easy control, the number of the links and the axis of the joints are identified, as in Table 1.

As in Table 2, it is composed by 9 links and 8 revolute joints and can operate the process tool to the desired position. The robot should penetrate the Vacuum Vessel through the limited size penetration Port in the equatorial plane. When plasma discharge, the robot is stored in a special cask. When needed, the robot would inspect the whole EAST vacuum vessel.

Table 1 Main mechanical parameters of the robot

Overall height (mm)	2190
Overall length (mm)	5020
Quantity of mechanical arm	9
Single arm length (mm)	583
Single arm weight (kg)	3.3
Total DOF	9
Maximum speed (mm/s)	50

Table 2 DOFof mechanical arm

DOF distribution	DOF type	Quantity	Range
Horizontal direction	Rotation	6	0°–220°
	Movement	1	0–1.5 m
Vertical direction	Rotation	2	0°–90°

2.2.1 Foldable Joint Design

The mechanism of every joint is the same and mainly includes driving wheel, motor, joint wheel, fixed pulley and steel wire. Due to the limited space, the axis of motor is parallel to that of each arm. The groove shape heart of driving wheel is in alignment to that of joint wheel. Axle of motor and the corresponding joint axle make up the orthogonal axis department, and the motor driving power of driving wheel can be transmitted to joint wheel through steel wire. In order to guarantee steady operation and friction decreasing of steel wire, the shape heart of groove of joint wheel and that of fixed pulley are flat and in the same level. The shape heart of angle encoder is coaxial with the joint axle in the same arm, and it obtains the angle of joint in real time. This joint structure fully utilizes the longitudinal space of the arm to realize the structural compactness design, which improves the rigidity of joint and it is convenient for measurement and assembly (Fig. 2).

2.2.2 Simulation Analysis

Buckling Stability Emulation.

The buckling factor of safety (BFS) is the factor of safety against buckling or the ratio of the buckling loads to the applied loads. As can be seen from the simulation result shown in Fig. 3, the applied loads are less than the estimated critical loads. BFS is positive & negative, namely $BFS < -1$ or $BFS > +1$, the structure can meet the requirement of stabilizes.

Joint Driving Moment Emulation.

Under the circumstances that the mechanical arm totally extend, the fixed end joint needs to overcome the frictional and starting moment of corresponding pivot. Figure 4 illustrates that specified torsion 5 Nm of motor can meet the power requirement of the fixed end joint.

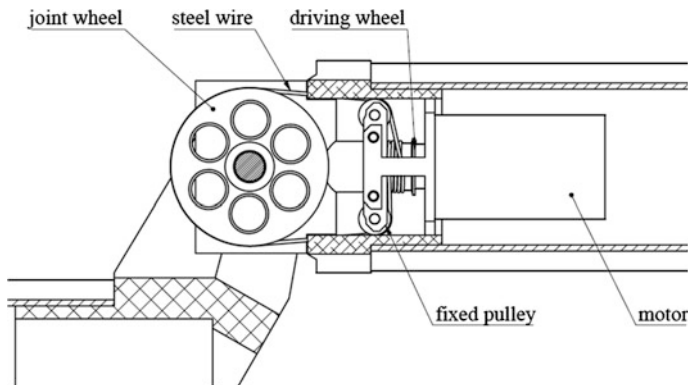


Fig. 2 Joint Mechanism

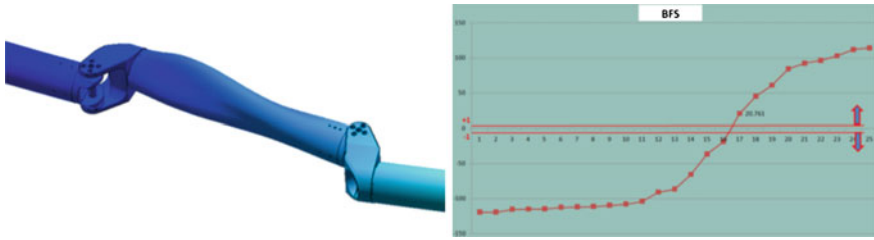


Fig. 3 Buckling simulation

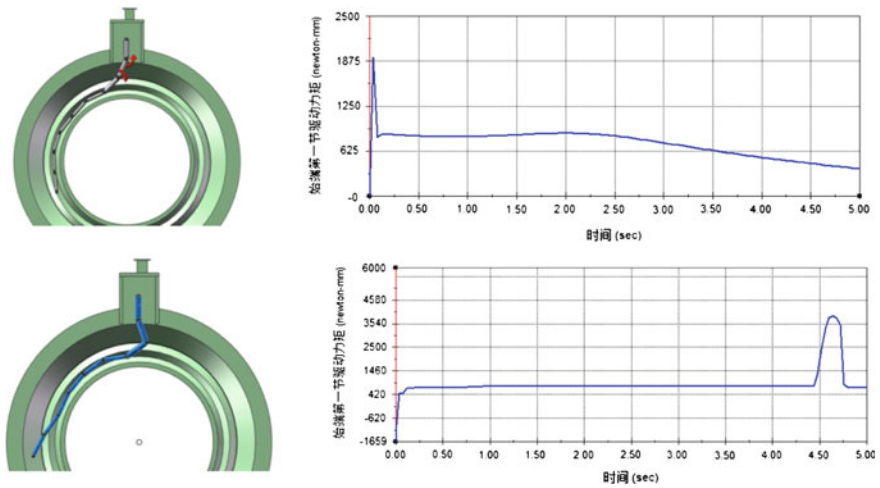


Fig. 4 driving moment analysis

3 Motion Analysis & Experiment Validation

Considering the inspection task, the inspection process can be separated into 3 stages: move into/out the vacuum vessel, arrive at the position of the horizontal cross-section of the Vacuum Vessel, reach the desired inspection location of the vertical cross-section.

3.1 Analytical Kinematics

3.1.1 Coordinate system

In Fig. 5, The MFR consists of 9 links and 9 joints, The joint 1 is prismatic and joints 2–9, are revolute joints, respectively. In order to solve the kinematic

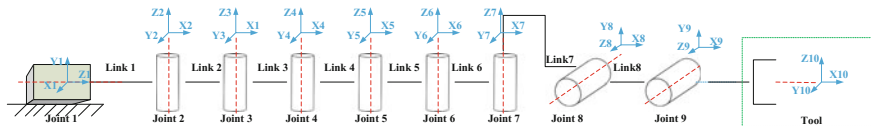


Fig. 5 The D-H coordinate system

Table 3 The MFRD-H parameters

i	$\alpha_n(^{\circ})$	a_n (m)	d_n (m)	$\theta_n(^{\circ})$	Type	Range
1	0	0	d_1	0	P	Linear 1.8 m
2	90	0	0.088	θ_2	R	Yaw $\pm 90^{\circ}$
3	0	0.6	-0.088	θ_3	R	Yaw $\pm 110^{\circ}$
4	0	0.6	0.088	θ_4	R	Yaw $\pm 110^{\circ}$
5	0	0.6	-0.088	θ_5	R	Yaw $\pm 110^{\circ}$
6	0	0.6	0.088	θ_6	R	Yaw $\pm 110^{\circ}$
7	0	0.6	-0.088	θ_7	R	yaw $\pm 110^{\circ}$
8	90	0.6	0.088	θ_8	R	Pitch $\pm 110^{\circ}$
9	0	0	-0.088	θ_9	R	Pitch $\pm 110^{\circ}$

problems, the frame is attached as follows: frame from 1 to n is attached to the joint n ($n = 9$), the origin of the frame 0 is coincident with the center of the EAST VV, the frame $n + 1$ is attached to the tool frame. Based on the attached frames, the kinematic model can be established. The D-H parameters are shown in Table 3.

3.1.2 Kinematics Model

Due to the inspection procedure, the position of the horizontal cross-section of the Vacuum Vessel is determined by joints 1–7. While, the position of the vertical cross-section is implemented by joints 8 and 9 respectively. Therefore, the overall forward kinematics T_{10}^0 can be divided into two kinematic equations T_7^0 and T_{10}^7 .

$$\begin{cases} T_7^0 = T_1^0 T_2^1 \dots T_5^4 T_7^6 \dots \dots \dots (a) \\ T_{10}^7 = T_8^7 T_9^8 T_{10}^9 \dots \dots \dots (b) \\ T_{10}^0 = T_7^0 T_{10}^7 \dots \dots \dots (c) \end{cases} \quad (1)$$

The inverse kinematics is referred to find the values of the joints angle and the parameters of the geometric link when the orientation and position of the end effector relative to the base frame is given, which is important for the robot control. For the serial-link mechanism such as the long reach robot, the inverse solution could be gotten from Eq. (3) when the pose of the end-effector is known.

$${}^0_{10}T = {}^0_1T_1 {}^1_2T_2 \dots {}^8_{10}T_9 = \prod_{n=1}^{10} {}^n_{n-1}T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ {}^{10}n_x & {}^{10}o_x & {}^{10}a_x & {}^{10}p_x \\ 0 & 0 & 0 & 0 \\ {}^{10}n_y & {}^{10}o_y & {}^{10}a_y & {}^{10}p_y \\ 0 & 0 & 0 & 0 \\ {}^{10}n_z & {}^{10}o_z & {}^{10}a_z & {}^{10}p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$${}^0T^{-1} {}^{10}T_10 = {}^1T_n \quad (3)$$

Equation (3) is redundant. Redundancy is important for the robot moving around obstacles and finding the desired location under geometry constrains with the disadvantages of increasing the difficulty of solving the inverse kinematics formula. Considering the geometry constrains of the VV and the redundancy mechanism, it is difficult to acquire the appropriate inverse kinematic solution. In order to facility the solution of the inverse kinematics and reduce the position control difficulty, the inspection procedure and the geometric parameters are taken into account.

Since the inspection stage can be separated, the Eq. (1) can be separated into two independent equations: Eqs. (1a, b). The complicated spatial problem have been simplified into two separate planar problems.

The MFR Jacobian matrices, which is very important to describe the relationships not only between the joint velocities and the corresponding angular and linear velocities in the Cartesian space but also between the joint torque and the force, in the base frame can be expressed as (4).

$$J = \begin{bmatrix} J_v \\ J_w \end{bmatrix} = \begin{bmatrix} {}^0Z_1 & {}^0Z_2 \times P_{2-10} & {}^0Z_3 \times P_{3-10} & {}^0Z_4 \times P_{4-10} & {}^0Z_5 \times P_{5-10} & {}^0Z_6 \times P_{6-10} & {}^0Z_7 \times P_{7-10} & {}^0Z_8 \times P_{8-10} & {}^0Z_9 \times P_{9-10} & 0 \\ 0 & {}^0Z_2 & {}^0Z_3 & {}^0Z_4 & {}^0Z_5 & {}^0Z_6 & {}^0Z_7 & {}^0Z_8 & {}^0Z_9 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} {}^0R \cdot Z & {}^0R \cdot Z \times P_{2-10} & {}^0R \cdot Z \times P_{3-13} & {}^0R \cdot Z \times P_{4-10} & {}^0R \cdot Z \times P_{5-10} & {}^0R \cdot Z \times P_{6-10} & {}^0R \cdot Z \times P_{7-10} & {}^0R \cdot Z \times P_{8-10} & {}^0R \cdot Z \times P_{9-10} & 0 \\ 0 & {}^0R \cdot Z & {}^0R \cdot Z & {}^0R \cdot Z & {}^0R \cdot Z & {}^0R \cdot Z & {}^0R \cdot Z & {}^0R \cdot Z & {}^0R \cdot Z & 0 \end{bmatrix} \quad (4)$$

where, J_v is related to the linear velocities, J_w is related to the angular velocities, 0Z_i is the Z vector of frame i with respect to the base frame, P_{m-n} is the origin of frame m with respect to the origin of frame n, 0R is the homogeneous transformation matrix with respect to the base frame.

When the Jacobian matrix is known, the instantaneous kinematics problem can be solved with the Eq. (5)

$$\dot{q} = J^{-1}(q)v_n \quad (5)$$

where v_n is the total velocity of end-effector, $J(q)$ and is $J^{-1}(q)$ the Jacobian matrix and the inverse Jacobian matrix, respectively, \dot{q} is traditional velocity and angular velocity about joint 1 and joints 2–9, respectively.

3.1.3 Workspace

For the MFR, the workspace is determined by the geometric structure of the robot (see Table 1) and the motion limitation of the joints, which can be calculated based

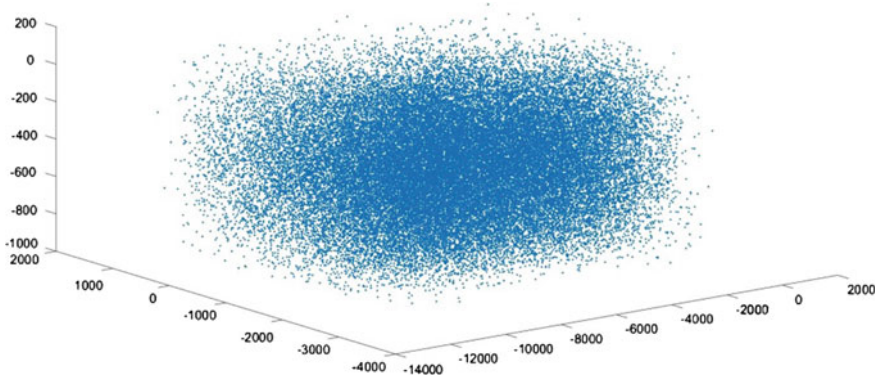


Fig. 6 Workspace of end-effectors by Monte Carlo method

on the forward kinematic model, as expressed as (6). Benefit from the long cantilever structure and the sufficient numbers of joints, the device has large workspace, which is important for performing the inspection work. The MFR cloud workspace can be seen as Fig. 6.

$$W = \{W(q): q \in Q\} \subset R^3 \quad (6)$$

where, W is the workspace, $W(q)$ is joint variable function, Q for joint space, R^3 is three dimensional space. The generalized range of joint motion can be expressed in inequality, $q_i, \min \leq q_i \leq q_i, \max$.

3.2 Experiments Validation

As shown in Fig. 7, we perform several motion control experiments in our laboratory. Experiments shows that, the designed mechanism is fulfilling the requirement. And the movement capability is validated.



Fig. 7 Control experiments

4 Conclusion

We designed a type of Multi-joint Foldable Robot aiming at the nuclear fusion Vacuum Vessel surroundings. The assembly test shows that the design mechanism is feasible and available. And the motion control experiments validate the whole system. At present, the prototype we manufactured can accomplish the inspection task. Furthermore, we will focus on improving the robot hanging down and developing the modular job tools for maintenance.

Acknowledgements This research work was supported by the China National Magnetic-Confinement Fusion Energy R&D Program (ITER) and funded by the National Science and Technology Major Project of the Ministry of Science and Technology of China (2012GB102002, No. 2012GB102007) and the National Natural Science Foundation of China (NSFC) under Grant No. 61503361. The natural science foundation of Anhui Province 1308085MF86. The authors wish to thank the research group for their great contributions to the presented research. The views and opinions expressed in this paper are the sole responsibility of the authors.

References

1. <http://west-east.ipp.ac.cn/index.php>
2. Ribeiro, I., Damiani, C., Tesini, A., Kakudate, S., Siuko, M., Neri, C.: The remote handling systems for ITER. *Fusion Eng. Des.* **86**(6–8), 471–477 (2011)
3. Loving, A., Allana, P., et al.: Development and application of high volume remote handling systems in support of JET and ITER. *Fusion Eng. Des.* **87**(5–6), 880–884 (2012)
4. Perrot, Y., Gargiulo, L., et al.: Long-reach articulated robots for inspection and mini-invasive interventions in hazardous environments: recent robotics research, qualification testing, and tool developments. *J. Field Robot.* **29**, 175–185 (2012)
5. Peng, X.B., Yuan, J.J., et al.: Kinematic and dynamic analysis of a serial-link robot for inspection process in EAST vacuum vessel. *Fusion Eng. Des.* **87**, 905–909 (2012)
6. Wang, H., Lai, Y., et al.: Design and implementation of motion planning of inspection and maintenance robot for ITER-like vessel. *Fusion Eng. Des.* **101**, 111–123 (2015)
7. Wu, Z., Wu, B., Zhang, Q., Wang, Z.: Development and analysis of a long reach robot for EAST vacuum vessel inspection. *J. Fusion Energy* (2015). doi:[10.1007/s10894-10015-19907-10896](https://doi.org/10.1007/s10894-10015-19907-10896)

The Design of Pipe Cutting Tools for Remote Handling in Maintenance Manipulator for Tokamak

Xizhe Zang, Zhenkun Lin, Yixiang Liu, Yanhe Zhu and Jie Zhao

Abstract The article describes the remote cutting tools designed for pipe cutting at Tokamak. Because of the special working environment, there are some design requirements for the cutting tools: the cutting swarf should be removed effectively, the quality of cut necessary for re-welding, and the cutting tools should be compact to fit into the maintenance manipulator for remote handling. The cutting tool consists of two parts: the cutting part and the sealing part. The two parts of cutting tool are designed to cut pipes and retain all cutting debris, respectively. A remote computer is used to input the commands and display data. During the cutting operation, the programmable controller has been used for controlling the cutting tool. The commands through the controller to motor drivers, implement the desired action.

Keywords Cutting tool · Tokamak · Remote · Compact

1 Introduction

As a new energy, nuclear fusion energy is rich in raw materials and pollution-free after the reaction, so it is generally considered to be an important way to solve today's energy crisis [1]. To promote the commercialization of fusion reactor engineering technology, many countries around the world have implemented a plan named the International Thermonuclear Experimental Reactor (ITER) [2]. With the implementation of the plan, the maintenance manipulator for Tokamak gradually becomes a hot topic. In this paper, a remote cutting tool installed on the manipulator designed for cutting pipes at Tokamak was studied.

As the ITER equipment will be working with the fusion reactor, so the high requirement for working environment is necessary. And the assembly of many of

X. Zang (✉) · Z. Lin · Y. Liu · Y. Zhu · J. Zhao
State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin
150080, Heilongjiang Province, China
e-mail: zangxizhe@hit.edu.cn

the ITER components requires the high vacuum integrity of welded joints. If a component failure will affect the operation of the ITER equipment. So the facility to replace it by remote handling techniques must be provided [3]. This includes the cutting of pipe joints using tools placed by a maintenance manipulator. A remote cutting tool which has been designed for cutting pipe joints at ITER will be reported below.

In order to be compatible with remote maintenance manipulator and the high requirement for working environment of ITER the weight of the cutting tool should be light, the structure of the cutting tool should be compact and the cutting debris should be retained by the cutting tool.

2 Selection of Cutting Technique

Replacing the failure component includes the cutting of pipe joints and the re-welding of the replacement one, so it is required that the quality of cut is adequate for reassembly without additional machining work. Because of the severe access restrictions of the ITER equipment the cutting tools should be compact to fit into the maintenance manipulator. So from the above, several cutting processes were considered: thermal, erosion and mechanical.

2.1 *Thermal*

The thermal technique includes gas cutting, plasma cutting and laser cutting. In the gas cutting process heat is provided by hydrogen burning with oxygen in a series of small flames emitted from holes in the nozzle periphery. This heats the steel locally to a temperature where an exothermic oxidation reaction will readily occur. A jet of oxygen is emitted from the center of the nozzle when the appropriate temperature has been reached. This supports the oxidation reaction and displaces oxidised metal, forming a cut. However, the oxidised surface of the cut is not suitable for re-welding.

The plasma system uses an electric arc between a tungsten electrode and the workpiece to be cut. The tip of the electrode is surrounded by a nozzle through which air flows at a high velocity. This constricts the arc into a narrow column of arc plasma, which heats and melts the workpiece. The air from the nozzle blows molten metal away from the liquid pool, producing a cut [4]. The process is started by striking an arc between the electrode and the constricting nozzle, before transferring the arc to the workpiece. Whereas the structure of the plasma cutting equipment is difficult to fit into the maintenance manipulator, it was not investigated in this article.

Laser cutting is a technology, which enables a laser to cut of various materials using the high-point of the cutting jet by the introduction of energy and technical

gas of high purity [4]. In the laser cutting process, we should site the source close to the component or guide the laser beam from a remote source. Both of the two methods above are difficult to implement, and laser cutting equipment is too large to be compact to fit into the maintenance manipulator. So use of a laser was impractical.

2.2 Erosion

Electric discharge machining (EDM) and Water Jet cutting are the widely used two kinds of erosion cutting methods. EDM is a thermo-electrical material removal process, in which the tool electrode shape is reproduced mirror wise into a work material, with the shape of the electrode defining the area in which the spark erosion will occur [5, 6]. EDM is accomplished with a system comprising two major components: a machine tool and power supply. Water jet is the method consisting of cutting the material by the use of thin water jets under high pressure with added abrasive slurry used to cut the target material by means of erosion. And the cutting process can be described briefly as follows. The water fed by a pump under pressure, after passing through the water causes the suction nozzle to the abrasive mixing chamber. Then a mixture of water and abrasive is directed to the mixing nozzle in order to form and stabilize. The result is a stream of hydro-abrasive, which has enough power to cut through even the toughest materials. However both methods involve liquids and produce very fine debris particles which it is impractical to retain it.

2.3 Mechanical

Mechanical cutting, which includes tooling and machining, is a process that uses power-driven equipment to shape and form material into a predetermined design. The principal processes for mechanical cutting are: grinding, turning, and milling. And also the wheel cutters have been applied in some cutting fields.

Grinding is an abrasive machining process that uses a grinding wheel as the cutting tool. It can produce very fine finishes and very accurate dimensions, yet in production contexts it can also produce cutting debris [7]. Cutting debris includes metal particles and fragments of the cutting wheel. And because of the high velocity of the grinding cutting wheel the cutting debris will have high velocity, so it is difficult to contain all of cutting debris. The reduction of the cutting wheel diameter during operation, and the high power input required also made this process unattractive for remote use.

In turning, a rotating cutting tool orbit around the pipe to implement the cutting operation. An incremental radial feed generates a narrow cut which has a flat face

which is suitable for re-welding. However, this process produces a continuous ribbon of swarf which could clog a tool mechanism.

Milling process are operations in which the cutting tool rotates to bring cutting edges to bear against the workpiece. In milling, a rotating tool with multiple cutting edges is moved slowly relative to the material to generate a plane or straight surface. The direction of the feed motion is perpendicular to the tool's axis of rotation. The speed motion is provided by the rotating milling cutter. In the milling process milling produces the cutting debris then made milling unattractive for our design.

Wheel cutters use sharp rimmed wheels to form a groove in the pipe section by plastic deformation. An incremental radial feed deepens the groove until brittle fracture occurs, producing a cut without any debris. The absence of debris makes this a most attractive solution for the remote cutting use at ITER. So we in this article we select wheel cutters for the ITER cutting tools and are discussed in detail below.

3 Remote Cutting Tool

The application of the remote cutting tool in maintenance manipulator for ITER yields a number of requirements for the cutting tool:

- To be extremely compact to fit into the maintenance manipulator.
- In order to be compatible with remote handling techniques he tools must be light.
- To be easily mounted at the end of the maintenance manipulator and simple to operate.
- In the cutting process the remote cutting tool must retain all cutting debris.
- The quality of cut should be adequate for reassembly without additional machining work.

With the requirement talked above a mechanical structure model for the remote cutting tool is designed in this paper.

3.1 3D Mechanical Model of the Cutting Tool

The tool, shown in Fig. 1, consists of two parts: cutting part and connection part. And as we can see from the figures the cutting tool is covered with an enclosure. This component and the remote auto-cover (as shown in the Fig. 1b) could be a form of protection of the cutting part also it can avoid contaminating the inside environment of ITER device with the cutting swarf. Motor drivers are placed in the connection part of the cutting tool. This type of installation saves the space of

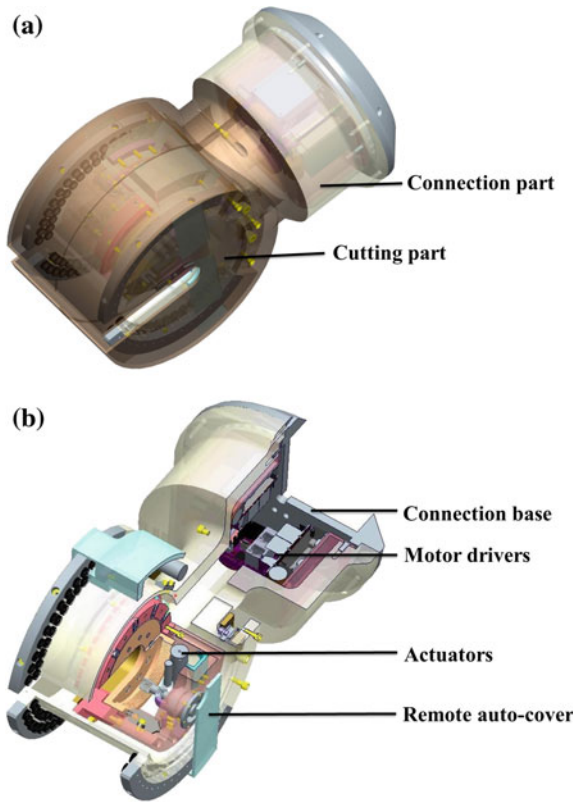


Fig. 1 a Remote cutting tool—assembly drawing and b remote cutting tool—sectional drawing

the cutting part effectively and makes the operation of the cutting part more convenient. The design of the connection base makes the cutting tool installed on maintenance manipulator more easily.

3.2 Structure of the Cutting Tool

Figure 2 illustrates the structure of remote cutting tool. The cutting part consists of the cutter, drive rotary module and the drive motor of remote auto-cover. In this article we select wheel cutters for the ITER cutting tools and the cutter components are showed in detail in Figs. 3 and 4. The drive rotary module (as shown in detail in the Fig. 5) of cutting tool drives the cutter rotating around the axis. The covers of cutting tool are driven by two motors (as shown in detail in the Fig. 6). By controlling the motor rotation to be able to implement open and close of the remote auto-cover. We will discuss it in detail in section E.

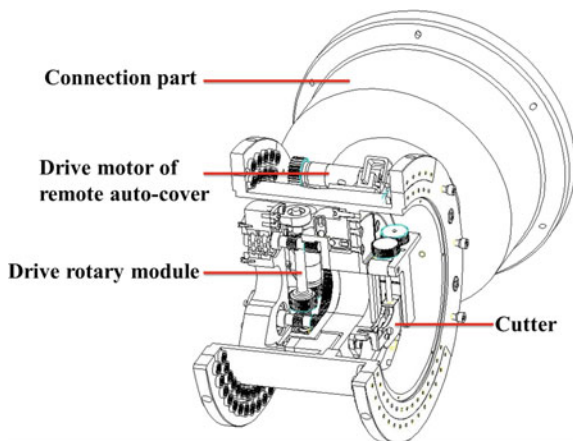


Fig. 2 Remote cutting tool—structural drawing

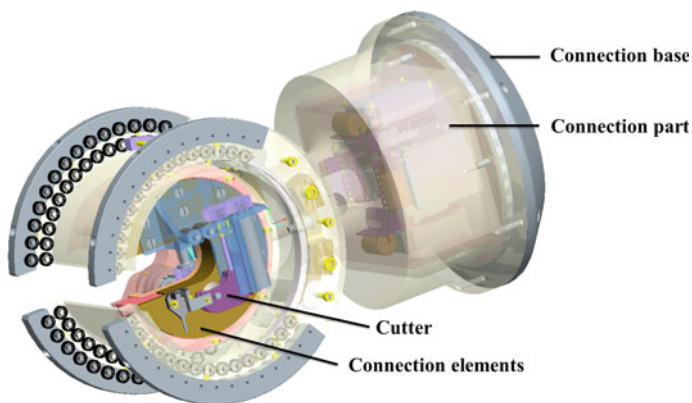


Fig. 3 Installation drawing of the cutter

3.3 Structure and Works of the Cutter

Figures 3 and 4 show the installation and structure of cutter of remote cutting tool. The cutter is placed on the central of the cutting part connected with the drive rotary module through the connection elements. Through the drive rotary module's driving the cutter can rotate around the axis to implement the ring cutting of pipes. The remote cutting tool is installed on maintenance manipulator through the connection base. Through controlling maintenance manipulator action the cutting tool can get the workpiece which one should be cut quickly.

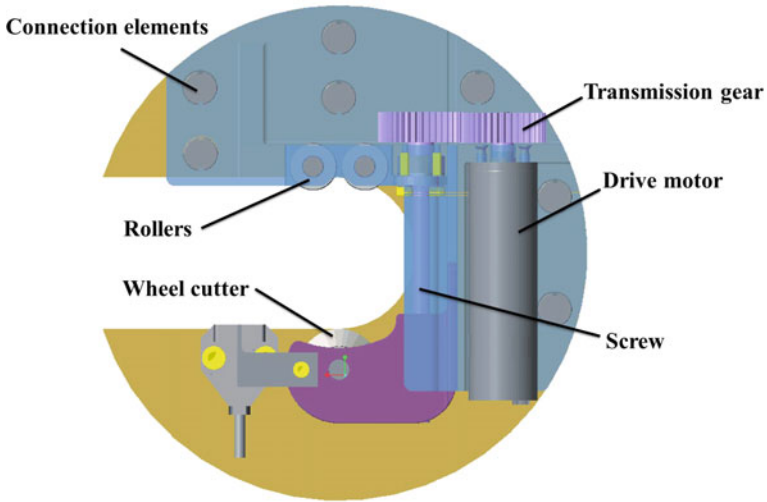


Fig. 4 Structural drawing of the cutter

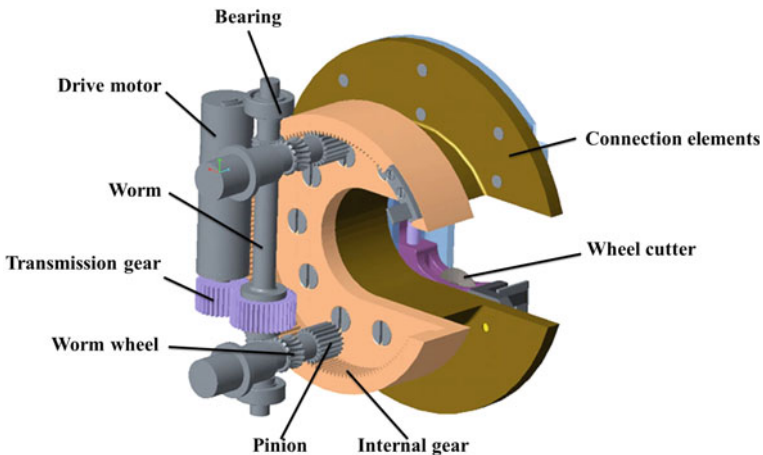


Fig. 5 Structural drawing of the drive rotary module

As we can see from the Fig. 4 the cutter includes feed motor, transmission gear, screw, cutter support, wheel cutter, rollers and connection elements. The cutter support is driven by feed motor through transmission gear and screw to provide an incremental feed vertical to the axis. Meanwhile, through the driving of rotary module the cutter can rotate around the axis. The depth of groove on pipes increase with the incremental radial feed until brittle fracture occurs. In this way there is no debris produced in cutting process. And the absence of debris makes it most suitable for the remote cutting use at ITER.

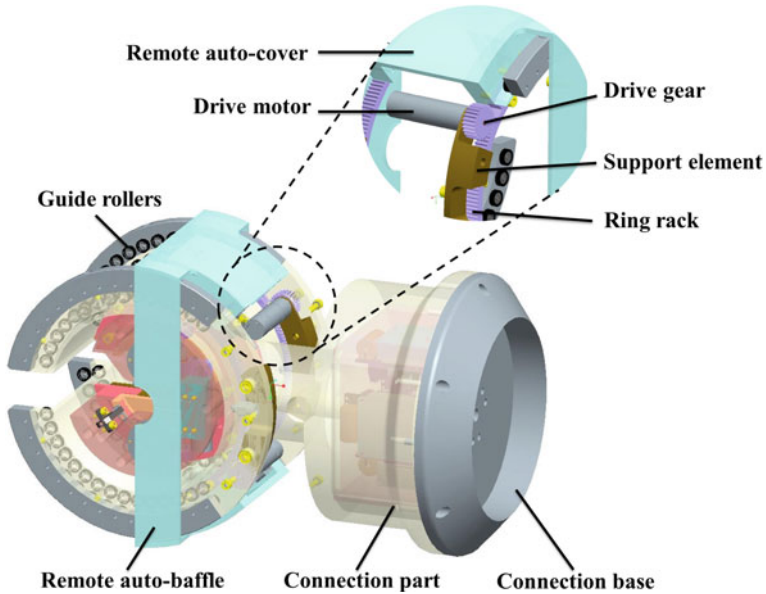


Fig. 6 Structural drawing of remote auto-baffle

3.4 Structure and Works of the Drive Rotary Module

Figure 5 illustrates the structure of the drive rotary module. As we can see from the figure this module includes rotary motor, transmission gears, worm, worm wheel, pinions, internal gear, bearings and connection elements. The internal gear is driven by rotary motor through worm and gear to provide the overall module and cutter a rotation around the axis. Continuous rotational motion can't be generated because of the gap of the internal gear. To solve this problem we set two pinions meshing with internal gear. The two pinions in this module are distributed in different positions meshing with the internal gear. In this manner, when one transmission fails, the other one can work normally.

3.5 Structure and Works of the Drive Rotary Module

Figure 6 demonstrates the structure of remote auto-cover of the cutting tool. The remote auto-cover consists of two drive motors, transmission gear, ring rack, guide rollers and the cover. The cover is driven by motors through transmission gear and ring rack to realize the opening and closing. When the cover opened cutting tool catch the pipe which one should be cut. Then with the cover closing the inside of cutting tool formed a closed space. In this way can avoid contaminating the

environment of ITER device during cutting process. Guide rollers installed on the cutting tool have roles in guiding and reducing friction during the movement of remote auto-cover.

4 System Overview

Figure 7 shows system of the remote cutting tool. As we can see from the figure the cutting tool system includes human-computer interactive system, central control unit, controller, hub, actuator, motor and encoder. During the cutting process we can control the cutting operation through human-computer interactive system.

In the movement, the controller gives instruction to the actuators. Then the actuators drive the motors. We choose Elmo for controller and actuator, and Maxon for motors and encoders.

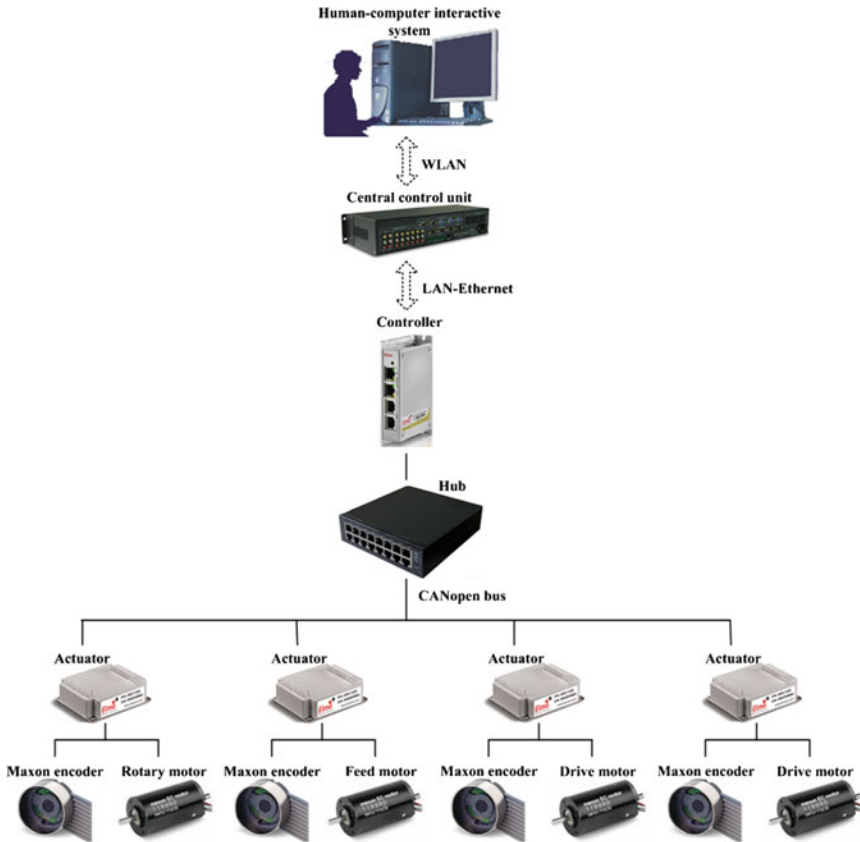


Fig. 7 System of the remote cutting tool

5 Conclusion

Because of the special working environment, there are some restrictions for the cutting operation. In this paper, we designed a remote cutting tool able to work in this environment well. We adopt wheel cutting as the tool's cutting method. In this way can effectively reduce the producing of cutting debris during cutting process. Different from the traditional cutting method the remote cutting tool cutting workpieces through the rotation of cutting tool around workpieces. The remote cutting tool is more compatible with the maintenance service at Tokamak. The remote auto-cover installed on cutting tool can avoid contaminating the inside environment of ITER device with the cutting swarf. Through the reasonable design and layout the remote cutting tool is compact to fit into the maintenance manipulator.

Acknowledgements This work was supported by National Magnetic Confinement Fusion Science Program "Multi-purpose Remote Handling System with Large-scale Heavy Load Arm" (2012GB102004).

References

1. Feng, K.: The International Thermonuclear Experimental Reactor (ITER) plan. *China Nuclear Power* **3**, 212–219 (2009)
2. Tesini, A., Palmer, J.: The ITER remote maintenance system. *Fusion Eng. Des. ISFNT-8 SI* **83**, 810–816 (2008)
3. Shuff, R., Mills, S.: A study of pipe jointing technology with reference to ITER requirements. *Fusion Eng. Des.* **84**, 1767–1769 (2009)
4. Krajcarz, D.: Comparison metal water jet cutting with laser and plasma cutting. *Proc. Eng.* **69**, 838–843 (2014)
5. Chena, S.L., Lina, M.H., Hsiehb, S.F., Chiou, S.Y.: The characteristics of cutting pipe mechanism with multi-electrodes in EDM. *J. Mater. Process. Technol.* **203**, 461–464 (2008)
6. Singh, S., Maheshwari, S., Pandey, P.C.: Some investigations into the electric discharge machining of hardened tool steel using different electrode materials. *J. Mater. Process. Technol.* **149**, 272–277 (2004)
7. Krzysztof, N.: The method of assessment of the grinding wheel cutting ability in the plunge grinding. *Central Eur. J. Eng.* **2**, 399–409 (2012)

Optimal Trajectory Planning for Manipulators with Flexible Curved Links

Liang Zhao, Hesheng Wang and Weidong Chen

Abstract Trajectory planning for manipulators with flexible links is a complicated task that plays an important role in design and application of manipulators. This paper is concerned with optimal trajectory planning for a two-link manipulator consisting of a macro flexible curved link and a micro rigid link for a point-to-point motion task. Absolute nodal coordinate formulation (ANCF) is used to derive the dynamic equations of the flexible curved link, an optimal trajectory method is adopted to generate the trajectory that minimizes the vibration of the flexible curved link. The Hamiltonian function is formed and the necessary conditions for optimality are derived from the Pontryagin's minimum principle (PMP). The obtained equations form a two-point boundary value problem (TPBVP) which can be solved by numerical techniques. Finally, simulations for the two-link manipulator are carried out to demonstrate the efficiency of the presented method. The results illustrate the validity of the method to overcome the high nonlinearity nature of the whole system.

Keywords Optimal trajectory planning · Flexible curved link · Absolute nodal coordinate formulation · Pontryagin's minimum principle

1 Introduction

Traditional manipulators are shown to be more and more inefficient as a result of their high power consumption and low speed caused by heavy links. Flexible links have been increasingly used in design and application of manipulators for their advantages over rigid ones. High speed operation and high ratio of payload to link weight can be obtained by developing lightweight mechanisms. However, lower link stiffness generates flexibility and causes vibration and deformation issues. In

L. Zhao · H. Wang · W. Chen (✉)

Key Laboratory of System Control and Information Processing, Department of Automation, Ministry of Education of China, Shanghai Jiao Tong University, Shanghai 200240, China
e-mail: wdchen@sjtu.edu.cn

these systems, the dynamic equations are highly nonlinear due to the flexibility, the traditional dynamic modeling method is no longer suitable. The vibration and deformation issues must be considered in trajectory planning of a manipulator with flexibility. Large efforts have been made to solve these problems and the problems can be divided to two main parts including dynamic modeling for a manipulator with flexible links and optimal trajectory planning method. However, there are few works about trajectory planning for a manipulator with flexible curved links.

Modeling for a manipulator with flexible links is a hard work, there have been a large number of related works during the last 30 years. The main strategy is turning the continuous nonlinear dynamic systems which possessing an infinite number of elastic degrees of freedom into finite dimensional systems. Assumed Mode Method (AMM) and Finite Element Method (FEM) are commonly used to describe these systems [1]. In [2], the lumped masses and massless springs method was used to model a long-reach articulated inspection manipulator. Kineto-Elasto-Dynamic Method given in [3] was applied to the dynamic analysis of elastic link mechanisms, without considering the coupling between rigid link motions and elastic deformations. However, numerical and experimental studies [4] have demonstrated that an accurate dynamic modeling of a manipulator with flexible link must consider the coupling between rigid link motions and elastic deformations. And FEM is considered to be more accurate than the AMM in describing flexible multi-link systems as reported in [5]. A highly accurate nonlinear dynamic model of a manipulator with flexible link is built, based on finite element discretization and Equivalent Rigid-link System formulation [6]. The ANCF [7] developed for modeling large rotation and deformation problems of flexible multi-body systems, has been successfully applied to large deformation problems such as catenary systems [8], solar panels [9].

Modeling of a flexible curved link is more complicated than a straight one. In [17], a curved beam element is developed for the analysis of large deformation of flexible multi-body systems using the absolute nodal coordinate formulation on the basis of the straight beam element. A new spatial curved slender beam finite element and a new cylindrical shell finite element are proposed in the frame of gradient-deficient absolute nodal coordinate formulation [18]. In [10], a spatial curved beam element based on Euler-Bernoulli beam assumption is derived, and numerical results demonstrate the validity of the method.

In point-to-point trajectory planning problem, only the initial and final states are given, and the manipulator is free to move between them. The trajectory planning problem can be transformed to an optimal problem minimizing a cost function. There have been a large number of studies about trajectory planning for rigid manipulators. In [11], an objective function composed of total execution time and the integral of the squared jerk (defined as the derivative of the acceleration) is minimized, so as to ensure the resulting trajectory is smooth enough. As for the trajectory planning of flexible-link systems, the feedforward techniques are used in this paper to minimize the vibration without using any sensors.

The approaches to solving such optimal trajectory planning problems can be classified into two main varieties [6]: direct and indirect methods. In direct method,

the optimal problem is converted into a parameter optimization one [12]. The problem of determining the maximum payload of a rigid manipulator is solved by direct method using the B-Spline functions [13]. Energy minimization approach for a two-flexible-link manipulator is evaluated using genetic algorithms and polynomial functions [14]. In indirect method, calculus of variation and PMP [15] are used to solve the optimal problems. And the resulting TPBVP can be solved by MATLAB. Indirect method is very accurate, especially when a large number of elastic degrees of freedom or composite objective is involved [15]. In [6, 16], the finite element method is employed to model and derive the dynamic equations of a manipulator and the indirect method is used to solve the optimal problem.

To the best of the authors' knowledge, there is no paper applying ANCF to derive the dynamic equations in solving optimal trajectory planning problem and there are very few papers about trajectory planning for manipulators with flexible curved links. In this paper, an accurate nonlinear dynamic model of a manipulator with a flexible curved link is generated based on the spatial curved beam element derived in [10] using ANCF, and the minimization of vibration is considered in solving the trajectory planning problem by indirect method.

The paper is organized as follows: In Sect. 2, the nonlinear dynamic model of a manipulator with a flexible curved link is generated based on the spatial curved beam element [10]. The optimal trajectory planning method is derived in Sect. 3 to minimize the vibration of the flexible curved link. In Sect. 4, simulations are given to illustrate the efficiency of the method. Finally, conclusions are given in Sect. 5.

2 Dynamic Modeling

In this section a brief explanation of the dynamic model used for the definition of the trajectory planning problem is given. The ANCF [7] is used to derive dynamic equations of the flexible curved link. This formulation uses global displacements and its slopes as the nodal generalized coordinates. Such generalized coordinates lead to a constant mass matrix and a simple expression of the generalized inertia forces. The resulting model, whose high accuracy has been proved in several papers [8, 9], accounts for the inertial nonlinearities of the manipulator and gives a coupled description of both the rigid link motions and elastic deformations.

In this paper, a modified spatial curved beam element shown in Fig. 1 is used to model the flexible curved link based on the element derived in [10]. The generalized nodal coordinates of the beam element are defined as:

$$e^{ij} = \left[(r^i)^T \quad \left(\frac{\partial r^i}{\partial X} \right)^T \quad (r^j)^T \quad \left(\frac{\partial r^j}{\partial X} \right)^T \right]^T \quad (1)$$

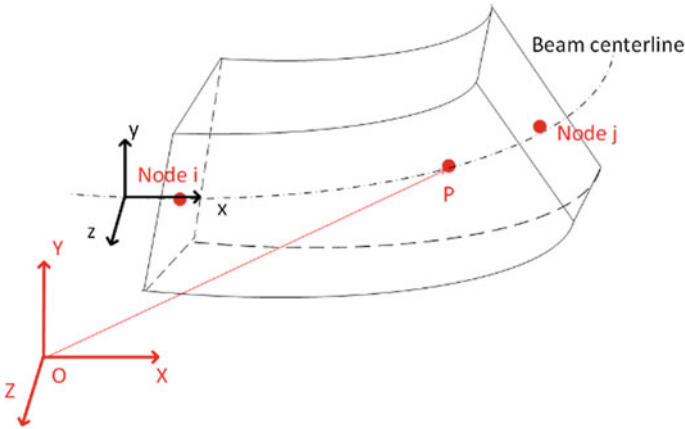


Fig. 1 Spatial curved beam element

where i and j are two nodes of the spatial curved beam element, r^i is the global position vector at node i , $\partial r^i / \partial X$ is gradient coordinate at node i , e^{ij} is a vector containing 12 elements.

The element shape function is defined as:

$$S^{ij} = [s_1(x)I_3 \quad s_2(x)I_3 \quad s_3(x)I_3 \quad s_4(x)I_3] \tag{2}$$

where

$$\begin{aligned} s_1 &= 1 - 3\xi^2 + 2\xi^3, s_2 = l(\xi - 2\xi^2 + \xi^3), \\ s_3 &= 3\xi^2 - 2\xi^3, s_4 = l(-\xi^2 + \xi^3), \end{aligned} \tag{3}$$

$0 \leq \xi = x/l \leq 1$, l denotes the initial arc length of the curved beam element, x is the local position defined in element coordinate system o -xyz.

Then the position vector of an arbitrary point on the centerline of the curved beam element can be defined in the global coordinate system O -XYZ as:

$$r^{ij} = S^{ij}(x)e^{ij} \tag{4}$$

According to the Euler-Bernoulli beam assumption, the cross section of the curved beam element is always perpendicular to the centerline of the curved beam element. Then the Frenet frame [20] of an arbitrary point p on the curved beam element is described as:

$$\begin{cases} \alpha(s) = \frac{dr}{ds} \\ \beta(s) = \frac{1}{\kappa(s)} \frac{d\alpha(s)}{ds} \\ \gamma(s) = \alpha(s) \times \beta(s) \end{cases} \tag{5}$$

where $\kappa(s)$ is the curvature of the centerline [20], $s = x$ represents the arc coordinate on the centerline of curved beam element.

The position vector r of the point p can be defined in the global coordinate system O-XYZ as:

$$r_p^{ij} = r^{ij} + y\beta(s) + z\gamma(s) \tag{6}$$

where y and z denote the local position coordinate in element coordinate system o-xyz. Using Eqs. (1) and (6), the kinetic energy T^{ij} can be represented as:

$$T^{ij} = \frac{1}{2} \int_{V^{ij}} \rho^{ij} (\dot{r}_p^{ij})^T \dot{r}_p^{ij} dV^{ij} = \frac{1}{2} (\dot{e}^{ij})^T M^{ij} \dot{e}^{ij}, \tag{7}$$

where $M^{ij} = \int_{V^{ij}} \rho^{ij} (S^{ij})^T S^{ij} dV^{ij}$ is the element constant mass matrix, leading to zero centrifugal and Coriolis inertia force.

The nonlinear elastic force can be defined from strain energy [10]. The strain energy can be derived using the continuum mechanics approach, as follows:

$$U = U^l + U^\kappa = \frac{1}{2} \int_0^l \int_A \left(E(\varepsilon_l)^2 + Ey^2(\kappa - \kappa_0)^2 \right) dAdx, \tag{8}$$

where U^l and U^κ are respectively longitudinal and bending strain energy, E is modulus of elasticity, ε_l and κ are longitudinal strain and curvature. And the nonlinear elastic force can be written as:

$$Q = \frac{\partial U}{\partial e} = EA \int_0^l \left(\frac{\partial \varepsilon_l}{\partial e} \right)^T \varepsilon_l dx + EI \int_0^l \left(\frac{\partial \kappa}{\partial e} \right)^T (\kappa - \kappa_0) dx, \tag{9}$$

where $I = \int_z \int_y y^2 dz dy$ is the cross sectional moment of inertia.

Therefore, the dynamic equations of a spatial curved beam element can be written in the following matrix form, Q_f^{ij} is the generalized external force of the element.

$$M^{ij} \ddot{e}^{ij} + Q^{ij} = Q_f^{ij}, \tag{10}$$

A general multibody system consists of collection of rigid and flexible links subject to kinematic constraints and forces as mentioned in [13]. The formulation of the nonlinear algebraic constraint equations of some commonly used joints is discussed in [19], such as revolute joints, fixed joints and sliding joints. The kinematic constraint equations that describe joints between links as well as time dependent specified motion trajectories are defined in terms of the system generalized coordinate vector q and time t as

$$C(q, t) = 0 \quad (11)$$

Using the preceding representation of the absolute node coordinate formulation and constraints between rigid and flexible links, we can derive the dynamic equations of a multi-body system. The generalized coordinate of the system is defined as $q = [r^T, e^T]$. Finally, the dynamic equations of the system can be defined as

$$\begin{bmatrix} M_r & 0 & (C_r)^T \\ 0 & M_e & (C_e)^T \\ C_r & C_e & 0 \end{bmatrix} \begin{bmatrix} \ddot{r} \\ \ddot{e} \\ \lambda \end{bmatrix} = \begin{bmatrix} Q_r \\ Q_e \\ Q_d \end{bmatrix} \quad (12)$$

where M_r is the rigid link's mass matrix, M_e is the constant mass matrix of the flexible curved link, Q_r is the generalized external force of the rigid link, Q_e is the generalized force vector of the flexible curved link including elastic force and external forces, λ is the Lagrange multiplier associated with the kinematic constraints. The system constraint equations defined in terms of the system generalized coordinate q and time t is $C(r, e, t) = 0$, $C_e = \partial C / \partial e$ is the derivation matrix of constraint equations with respect to the generalized coordinates e , $C_r = \partial C / \partial r$ is the Jacobian matrix for the rigid link coordinate, and Q_d is the right-hand side of acceleration for the constraint equations.

Equation (12) can be written in compact form as:

$$M\ddot{q} = Q \quad (13)$$

By defining the state vector as:

$$X = [X_1 \quad X_2]^T = [q \quad \dot{q}]^T \quad (14)$$

Equation (13) can be rewritten in state space form as:

$$\dot{X} = [\dot{X}_1 \quad \dot{X}_2]^T = [X_2 \quad M^{-1}Q]^T \quad (15)$$

Then optimal trajectory planning method is employed to determine the position and velocity variable $X_1(t)$ and $X_2(t)$ which optimize a well-defined performance measure when the model is given in Eq. (15).

3 Optimal Trajectory Planning Method

The target of this study is to find a way to compute a trajectory that brings the manipulator from a given initial state $X(t_0) = b_0$ to the final state $X(t_f) = b_f$ in a given time. Among the infinite number of choices, the trajectory that we are looking

for is the solution of the optimization problem expressed by Eq. (16). In Eq. (16), $\|X\|_W^2 = X^T W X$ is the generalized squared norm, W is symmetric, positive semi-definite weighting matrix. $X_2(t)$ is the velocity variable defined in Eq. (14). And $C(q, t) = 0$ is the constraint equation mentioned before.

$$\begin{cases} \min_{\theta(t)} J_0 = \int_{t_0}^{t_f} \frac{1}{2} \|X_2\|_W^2 \\ \text{subject to} \\ C(q, t) = 0 \\ X(t_0) = b_0 \\ X(t_f) = b_f \\ \dot{\theta}^- \leq \dot{\theta} \leq \dot{\theta}^+ \end{cases} \quad (16)$$

The above problem is a Bolza problem [6], the objective function is minimized over the entire duration of the motion. One of the ways to solve this problem is to formulate a TPBVP through the use of PMP and Hamilton-Jacobi-Bellman (HJB) equation [15]. According to the PMP, the following conditions must be satisfied to solve the optimal problem:

$$\begin{aligned} \dot{X} &= \partial H / \partial P \\ \dot{P} &= -\partial H / \partial X \\ 0 &= \partial H / \partial \theta \end{aligned} \quad (17)$$

where by defining the nonzero costate vector $P = [P_1^T \ P_2^T]^T$, the Hamilton function can be obtained as:

$$H(X, \theta, P) = 0.5 \|X_2\|_W^2 + P_1^T X_2 + P_2^T (M^{-1} Q) \quad (18)$$

If constraints are present on the joint acceleration, then the following limitation must be taken into account:

$$\ddot{\theta} = \begin{cases} \ddot{\theta}^+ & \ddot{\theta} \geq \ddot{\theta}^+ \\ \ddot{\theta}^* & \ddot{\theta}^- \leq \ddot{\theta} \leq \ddot{\theta}^+ \\ \ddot{\theta}^- & \ddot{\theta} \leq \ddot{\theta}^- \end{cases} \quad (19)$$

θ being the revolute joint angle and θ^* being the optimal joint angle is then obtained.

The boundary conditions will be given as: $X(t_0) = b_0, X(t_f) = b_f$. Differential equations given in Eq. (17) combining with $4n$ boundary conditions are used to determine the $4n$ state and costate variables, n is the number of generalized variables. The set of differentials and boundary conditions can construct a TPBVP, which can be efficiently solved using the collocation method. MATLAB routine “bvp4c” has been proved to be quite efficient for the task, therefore it was used to compute the problems presented in this work.

4 Simulation

In this Section, simulations are carried out for the two-link manipulator as shown in Fig. 2. This manipulator is composed of a macro flexible curved link and a micro rigid link. One end of the macro link is fixed at joint A, the other end is connected with the micro link through joint B. And the macro link is parallel to the horizontal plane initially, micro link can rotate around joint B in the vertical plane. The kinematic and dynamic parameters of the manipulator are reported in Table 1. The flexible curved link has been represented by a single finite element using absolute nodal coordinate formulation.

Fig. 2 Two-link manipulator

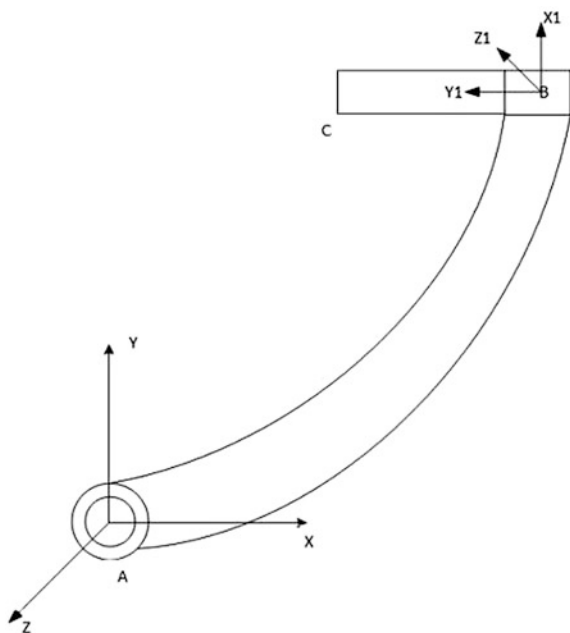


Table 1 kinematic and dynamic parameters of the manipulator

Name	Value	Unit
Radius of macro link	$R = 2$	m
Length of links	$L1 = 3.14, L2 = 1$	m
Width of links	$b1 = 0.2, b2 = 0.1$	m
Height of links	$h1 = 0.2, h2 = 0.1$	m
Density of links	$d1 = 2700, d2 = 7800$	kg/m^3
Module of elasticity of links	$E1 = 6.9e10, E2 = 1.2e11$	Pa

The generalized coordinates of the macro link can be defined as:

$$e = [e_1 \ e_2 \ e_3 \ e_4 \ e_5 \ e_6 \ e_7 \ e_8 \ e_9 \ e_{10} \ e_{11} \ e_{12}]^T \tag{20}$$

The generalized coordinates of the micro link can be defined as:

$$r = [r_1 \ r_2 \ \theta]^T \tag{21}$$

where r_1, r_2 are the location of the center of mass. θ is the joint angle between the micro link and the macro link. Then the state vectors can be defined as follows:

$$\begin{aligned} X_1 &= [x_1 \ x_3 \ x_5 \ x_7 \ x_9 \ \dots \ x_{29}]^T \\ X_2 &= [x_2 \ x_4 \ x_6 \ x_8 \ x_{10} \ \dots \ x_{30}]^T \end{aligned} \tag{22}$$

It can be written as:

$$\dot{x}_{2i-1} = x_{2i}, \quad \dot{x}_{2i} = M^{-1}Q \quad i = 1 \dots 15 \tag{23}$$

And the boundary condition is given as:

$$\theta_0 = 0, \quad \theta_f = \pi/4 \text{ rad}, \quad t_f = 1 \text{ s} \tag{24}$$

The penalty matrix in Eq. (16) can be selected as:

$$W = \text{diag}(0000002222000000) \tag{25}$$

Figure 3 shows two kinds of trajectory of the micro link including the optimized trajectory by solving the problem described in Eq. (16) and a cubic polynomial trajectory. Both the initial joint angle is 0 and the final joint angle is $\frac{\pi}{4}$ rad at 1 s, the

Fig. 3 Trajectory of the micro rigid link

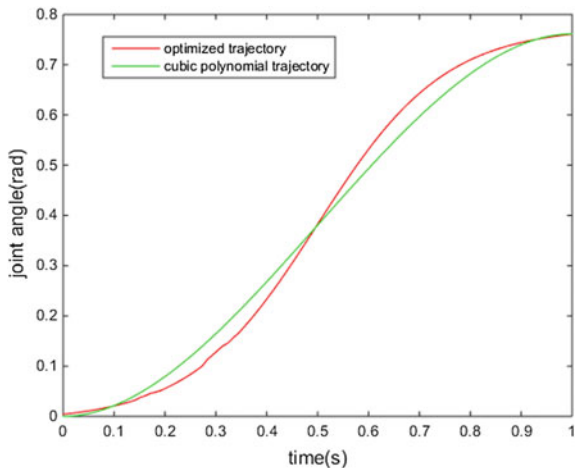


Fig. 4 Deformation at the end of the flexible curved link

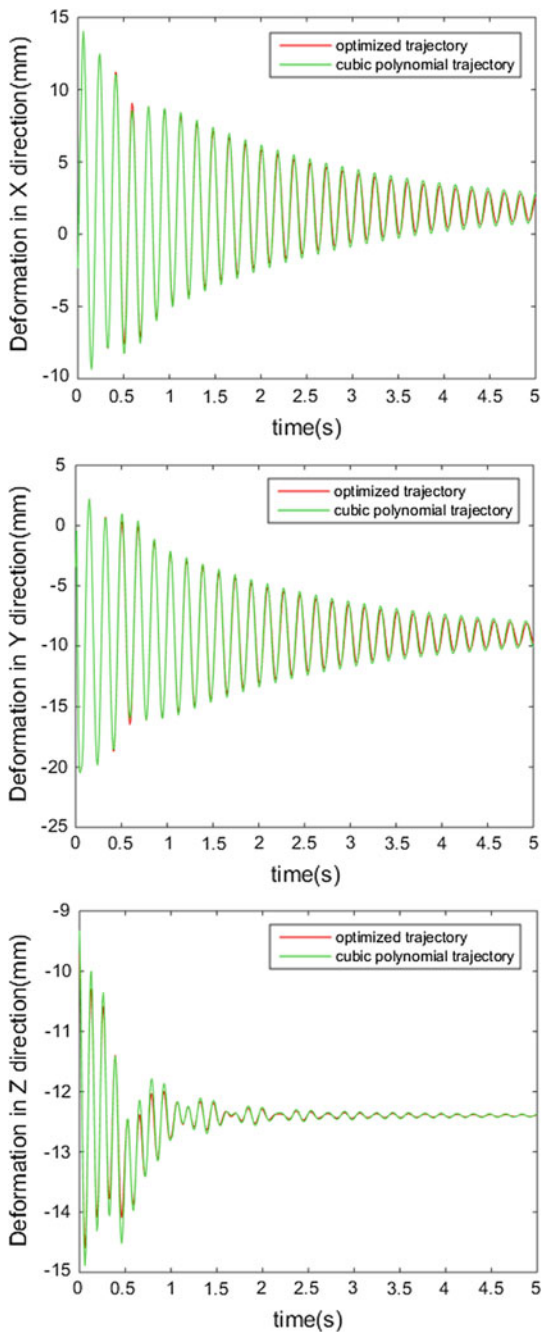
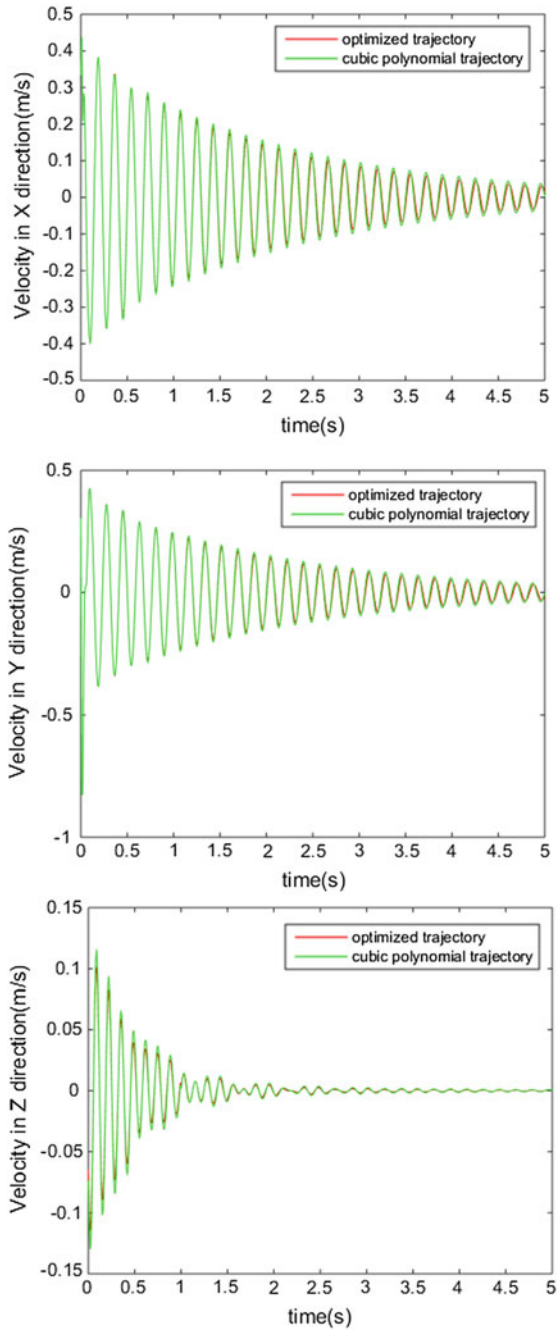


Fig. 5 Velocity at the end of the flexible curved link



micro link keeps still after 1 s. Vibration phenomenon of the flexible curved link is generated as a result of the micro link's motion.

Figures 4 and 5 show the deformation deviating from the initial position and the velocity of the motion at the end of the flexible curved link in three directions respectively. The velocity of the motion at the end of the flexible curved link can be used to describe the phenomenon of vibration. The simulation curves show that the maximum amplitude and velocity arise at the beginning of the motion for both two circumstances. The vibration phenomenon still exists after one second motion of the micro link and decreases gradually later.

Figures 4 and 5 show that the deformation and vibration caused by two kinds of motion have little difference in X and Y direction, but there is obvious difference in Z direction between these two kinds of motion. The decrease of both the deformation and vibration phenomenon illustrates the validity of the optimization method.

5 Conclusions

In this paper, formulation for trajectory planning of a two-link manipulator consisting of a macro flexible curved link and a micro rigid link in point-to-point motion based on the indirect optimal approach is presented. A spatial curved beam element is adopted to model the manipulator using absolute nodal coordinate formulation and an efficient solution on the basis of TPBVP is used to minimize the vibration of the manipulator. Simulations are given to investigate the validity of the algorithm. And numerical results illustrate the capability of the method to overcome the high nonlinearity nature of the whole system.

Acknowledgements This work was supported in part by China Domestic Research Project for the International Thermonuclear Experimental Reactor (ITER) under Grant 2012GB102001.

References

1. Dwivedy, S.K., Eberhard, P.: Dynamic analysis of flexible manipulators, a literature review. *Mech. Mach. Theory* **41**(7), 749–777 (2006)
2. Dubus, G., David, O., Measson, Y.: Vibration control of a flexible arm for the ITER maintenance using unknown visual features from inside the vessel. In: *IEEE/RSJ International Conference. Intelligent Robots and Systems. IROS 2009*, pp. 5697–5704. IEEE (2009)
3. Winfrey, R.C.: Dynamic analysis of elastic link mechanisms by reduction of coordinates. *J. Eng. Ind.* **94**(2), 577–581 (1972)
4. Turcic, D.A., Midha, A., Bosnik, J.R.: Dynamic analysis of elastic mechanism systems. Part II: Experimental results. *J. Dyn. Syst. Meas. Control* **106**(4), 255–260 (1984)
5. Theodore, R.J., Ghosal, A.: Comparison of the assumed modes and finite element models for flexible multilink manipulators. *Int. J. Robot. Res.* **14**(2), 91–111 (1995)

6. Boscariol, P., Gasparetto, A.: Model-based trajectory planning for flexible-link mechanisms with bounded jerk. *Robot. Comput.-Integr. Manuf.* **29**(4), 90–99 (2013)
7. Shabana, A.A.: *Dynamics of multibody systems*. Cambridge university press. (2013)
8. Seo, J.H., Sugiyama, H., Shabana, A.A.: Three-dimensional large deformation analysis of the multibody pantograph/catenary systems. *Nonlinear Dyn.* **42**(2), 199–215 (2005)
9. Zhao, J., Tian, Q., Hu, H.Y.: Deployment dynamics of a simplified spinning IKAROS solar sail via absolute coordinate based method. *Acta. Mech. Sin.* **29**(1), 132–142 (2013)
10. Lin, L., Lu, J., Chen, W.: Modeling and simulation of EAST flexible in-vessel inspection robot based on absolute nodal coordinate formulation. In: 2014 IEEE International Conference Robotics and Biomimetics (ROBIO), pp. 161–166. IEEE (2014)
11. Gasparetto, A., Zanotto, V.: A technique for time-jerk optimal planning of robot trajectories. *Robot. Comput.-Integr. Manuf.* **24**(3), 415–426 (2008)
12. Hull, D.G.: Conversion of optimal control problems into parameter optimization problems. *J. Guid. Control Dyn.* **20**(1), 57–60 (1997)
13. Wang, C.Y.E., Timoszyk, W.K., Bobrow, J.E.: Payload maximization for open chained manipulators: finding weightlifting motions for a Puma 762 robot. *IEEE Trans. Robot. Autom.* **17**(2), 218–224 (2001)
14. Faris, W.F., Ata, A., Sa'adeh, M.Y.: Energy minimization approach for a two-link flexible manipulator. *J. Vibr. Control* (2009)
15. Kirk, D.E.: *Optimal Control Theory: An Introduction*. Courier Corporation (2012)
16. Korayem, M.H., Haghpanahi, M., Rahimi, H.N.: Finite element method and optimal control theory for path planning of elastic manipulators. In: *New Advances in Intelligent Decision Technologies*, pp. 117–126. Springer, Berlin, Heidelberg (2009)
17. Sugiyama, H., Suda, Y.: A curved beam element in the analysis of flexible multi-body systems using the absolute nodal coordinates. *Proc. Inst. Mech. Eng. Part K: J. Multi-body Dyn.* **221**(2), 219–231 (2007)
18. Liu, C., Tian, Q., Hu, H.: New spatial curved beam and cylindrical shell elements of gradient-deficient absolute nodal coordinate formulation. *Nonlinear Dyn.* **70**(3), 1903–1918 (2012)
19. Sugiyama, H., Escalona, J.L., Shabana, A.A.: Formulation of three-dimensional joint constraints using the absolute nodal coordinates. *Nonlinear Dyn.* **31**(2), 167–195 (2003)
20. Gerstmayr, J., Shabana, A.A.: Analysis of thin beams and cables using the absolute nodal co-ordinate formulation. *Nonlinear Dyn.* **45**(1–2), 109–130 (2006)

Predictive Display for Telerobot Under Unstructured Environment

Qing Wei, NaiLong Liu and Long Cui

Abstract In this paper, the issue that traditional virtual reality (VR) system can't be used in unstructured environment is addressed, and a novel predictive display method based on 3D scene reconstruction online is proposed. In this method, the virtual environment is modeled and reconstructed online by the structured light, so the consistency with the real world can be ensured; An adaptive random sample consensus (ARANSAC) algorithm is proposed to denoise the point cloud when modeling the environment, which adjusts the parameters of the standard RANSAC algorithm adaptively; A rigid contact model based haptic rendering algorithm is adopted to generate the force feedback directly from the manipulator's dynamics, which eliminates the time delay and security problem introduced by the feedback force and increases the frequency of force feedback; Moreover, a low frequency filter is designed to keep the virtual force smooth. Finally, the effectiveness of the proposed method under unstructured environment is demonstrated by experiments.

Keywords Telerobot · VR · Unstructured environment · Scene reconstruction · Force feedback

Q. Wei · N. Liu (✉) · L. Cui
State Key Laboratory of Robotics, Shenyang Institute of Automation,
Chinese Academy of Sciences, Shenyang 110016, China
e-mail: liunailong@sia.cn

Q. Wei
e-mail: weiqing@sia.cn

L. Cui
e-mail: cuilong@sia.cn

Q. Wei · N. Liu
University of Chinese Academy of Sciences, Beijing 100049, China

1 Introduction

Telerobotics, perhaps one of the earliest aspects of robotics, and has been used in many areas, like space exploration, underwater exploration and nuclear environment. When teleoperating a robot, the perception is very important for the teleoperator [1]. However, affected by the random communication delay, the interactive information is delayed, deficient and scrambled. In order to overcome the time delay issue, many control strategies have been proposed [2–10], and the predictive display method has been reputed as the most promising solution to tackle big time delay. But, the predictive display technology relies heavily on the accuracy of the virtual model, so how to apply it to unstructured environment become a serious problem.

In Mars exploration, NASA built the 3D virtual environment by the received stereo images [11], but the accuracy, instantaneity and robustness of this method can't be insured. A point cloud based geometry modeling method is proposed by Fan Qiang [12], the algorithm of this method doesn't require complex calculation, so the instantaneity can be insured, and this method has been used in many virtual reality systems [13–15]. But it does not mention the environment dynamics and the point cloud processing algorithms have not been deeply discussed, and the lose of contact force feedback is dangerous in contact tasks. In [3], a virtual force computing method is proposed, and the dynamic parameters of the remote environment is identified by the sliding-average least-square algorithm. But the algorithm can not be used in variable time delay condition, and the safety can't been insured when the slave environment is hard. Moreover, the collaborative expression of predictive vision and force is also an important and urgent problem should be solved.

In order to solve the aforementioned issues, a novel predictive display method based on 3D scene reconstruction online is proposed in this paper. In this method, the geometry model of the environment is constructed by point cloud in real time, and an adaptive random sample consensus (ARANSAC) algorithm is proposed to denoise the point cloud. On the force interaction, the manipulator's dynamics which can be calculated in advance is used to calculate the virtual force. The proposed method eliminates the time delay and security problem introduced by the feedback force and improve the adaptability of the algorithm in unstructured environment. In order to realise the collaborative expression of predictive vision and force, the physical simulation and graphic simulation are calculated in different thread, so the haptical and graphical refresh rate meet the requirements of continuous perception. The teleoperation system constructed in this paper was shown in Fig. 1.

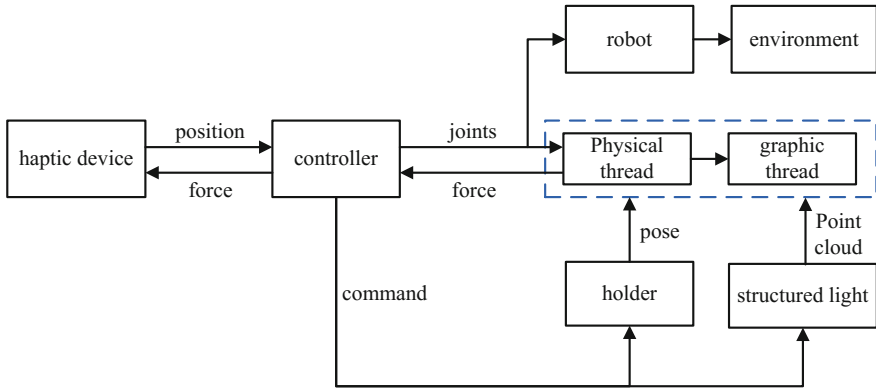


Fig. 1 The teleoperation system constructed in this paper

2 3D Scene Reconstruction

In this paper, we use structured light to get the point cloud of the fusion reactor. The structured light calculates the depth data by triangulation principle, so the processing method is very simple, and has a high real time capability. But affected by light and other factors, the original point clouds are normally quite noisy, so denoising and smooth operation are very important. In our experiment, the reconstructed model consists of three parts: fusion reactor (a part of the spherical), beryllium tiles (plane) and obstacles (No contact task, don't need denoising). The model of the point clouds need denoising are known, and an adaptive random sample consensus (ARANSAC) algorithm is proposed in our work to perform this operation.

2.1 A-RANSAC and Point Cloud Denoising

The standard random sample consensus (RANSAC) algorithm is proposed by Fischler [14], and proceeds as follows: first, samples of points are drawn uniformly and at random from the input data set. Each point has the same probability of selection. For each sample of points, a model hypothesis is created by computing model parameters from it. In the next step, the quality of the model is evaluated on the whole input data set. A cost function, which evaluates the model quality by calculating a distance threshold δ , is applied to identify the inliers. This process is repeated and the model with the largest number of inliers is stored, and terminated when the probability p that at least one of the random samples of points is free from outliers becomes high enough. Assume that w is the proportion of inliers in the data, n is the sample size and k is the number of iterations required for RANSAC.

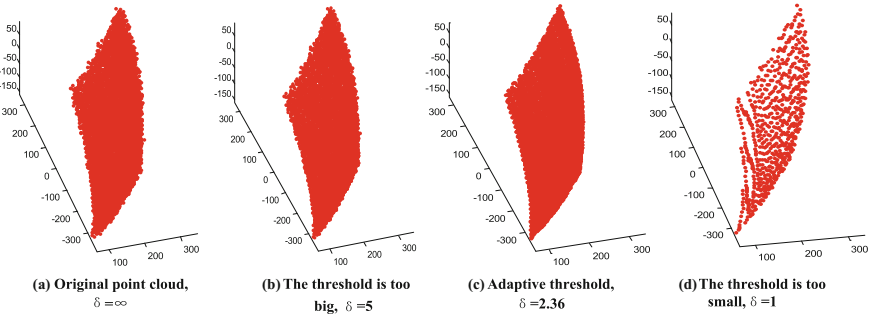


Fig. 2 The denoising results contrast of adaptive threshold and estimated threshold

$$1 - p = (1 - w^n)^k \tag{1}$$

Then, we can get:

$$k = \frac{\log(1 - p)}{\log(1 - w^n)} \tag{2}$$

In the standard RANSAC, w and δ are two important parameters. However, their values are estimated, if the w is too large, maybe we can't get the right model; In turn, if the w is too small, the instantaneity of the algorithm will be reduced. On the other hand, if the threshold δ is too large, the denoising result maybe not good; In turn, choose the threshold δ too small will cause atrophy of the target model.

In order to reduce the randomness of parameter estimation, an adaptive random sample consensus (ARANSAC) algorithm is proposed. In our algorithm, the w and δ are adjusted based on the point cloud and least square method. First of all, identify the original mode through least square method, and the δ is calculated as (3).

$$\delta = \sqrt{\frac{\sum_{i=1}^n (d_i - d_m)^T (d_i - d_m)}{n - 1}} \tag{3}$$

where d_i is the distance of point I to the original model, and d_m is the average of d_i . The number of inliers n_i can be get based on the original model and δ , and the w is calculated as $w = n_i/n$. Then the algorithm proceeds as the standard random sample consensus (RANSAC) algorithm.

In order to verify the effectiveness of the proposed algorithm, a contrast matlab simulation is conducted, and the result was shown in Fig. 2 (the data is a part of the point cloud of the fusion reactor). From the result we can see, compared with the standard random sample consensus (RANSAC) algorithm, our algorithm not only denoises the point cloud, but also keeps more surface details. What more, the randomness of the parameter estimation is eliminated.

2.2 Scene Reconstruction

The point cloud scanned by the structured light is in the camera coordinates, but the robot is in the world coordinates. So the point cloud should be converted to the world coordinates before reconstruction. In our system, the structured light camera fixed at the end of a holder, the holder has three DOFs, which ensures the structured light can scan all the workspace of the robot. The joints of the holder can be obtained through the controller, and the transformation matrix from the camera coordinates to the world coordinates is shown as (4).

$${}^c_w T = \begin{pmatrix} c_1 c_2 c_3 - s_1 s_3 & s_1 c_3 + c_1 c_2 s_3 & c_1 s_2 & l \cdot s_1 \\ -c_1 s_3 - s_1 c_2 c_3 & c_1 c_3 - s_1 c_2 s_3 & -s_1 s_2 & l \cdot c_1 \\ -s_2 c_3 & -s_2 s_3 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

where, $s_i = \sin \theta_i$, $c_i = \cos \theta_i$, $i = 1, 2, 3$, θ_i is the joints of the holder, and l is the long of the holder.

As the point cloud has been denoised, we use the ball-pivoting algorithm to convert it to a triangle mesh. The vertex normal is the average of the triangle normals that sharing this vertex. The framework of the mesh data is fixed as vertex-normal-index, which decreases the number of logic judgment when loading the model, and improves the real-time performance of the system.

3 Force Predictive Display

Force information is an effective supplement to the vision, particularly in the contact tasks, where 70 % of the perceptual information are get from force. However, influenced by the communication delay, the feedback force from the slave always be delayed, which dislocates the force feedback and the prediction graphics. The dislocation not only makes the operator fatigue but also threaten operation safety, so the real time force feedback is also very important in the predictive display system.

A great number of research on virtual force calculation have been reported, for example, mass-damp-spring model, artificial potential field model and virtual fixture method. However, those methods can't be used in our system directly. Because the environment is unstructured and the task is contact task with rigid bodies, so even very small delay or modeling errors would cause large contact forces.

Considering the deformation and penetration would not happen in rigid collision, a novel mass-damp-spring model based virtual force calculation method is proposed. The dynamic parameters of the environment is instead by manipulator's, and the calculation model as follows:

$$\begin{aligned}\tau &= M(\Theta) \ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) \\ F &= J^{-T}(\Theta) \cdot [\tau - G(\Theta)]\end{aligned}\quad (5)$$

where, τ is the joint torque, $M(\Theta)$ is the mass matrix of the manipulator, $V(\Theta, \dot{\Theta})$ is the coriolis and centrifugal, $G(\Theta)$ is the gravity terms and F is the contact force and torque. $J(\Theta)$ is manipulator's jacobian matrix when the collision occurs. As the manipulator would stop after collision, so the inverse of the jacobian matrix only calculates once in a collision course, and the frequency of force feedback increases. The virtual manipulator uses the same velocity planning algorithm as the real manipulator, $\ddot{\Theta}$ and $\dot{\Theta}$ is calculated based on the joints when the collision occurs, the target joints and the manipulator's velocity planning algorithm.

4 Experiments

Our experiment platform is shown in Fig. 3, consists of a stereoscopic projection system, a structured light camera (3D CaMega CP-300), a holder, a 6DOFs robot (Staubli TX60L), a master manipulator (Force Dimension Omega.3) and a control server. The communication channel between the master and slave is internet.

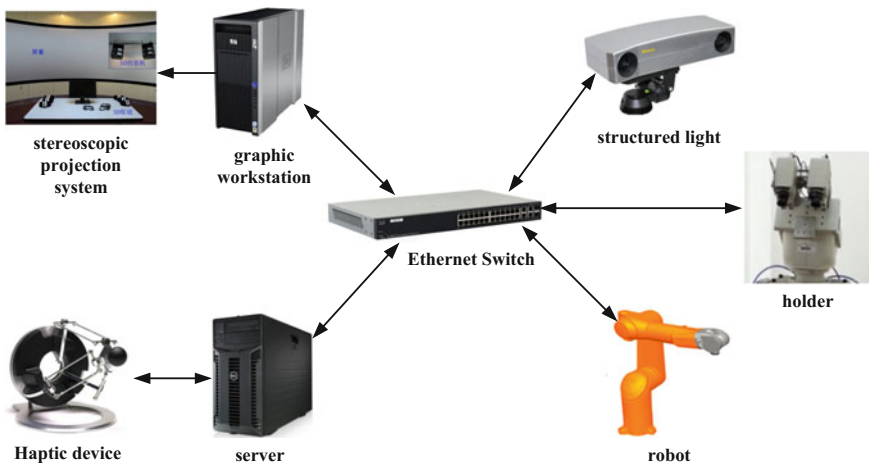


Fig. 3 Our experiment platform

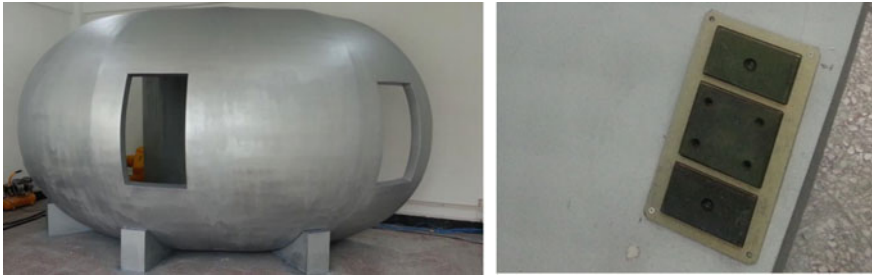


Fig. 4 The slave environment

4.1 Geometry Reconstruction

The slave environment consists of a fusion reactor and three beryllium tiles, as Fig. 4. In order to improve the instantaneity of the system, only a part of the environment (in the workspace of the robot) need to be reconstructed.

First, move the holder to a suitable pose by the master controller; then give a scanning instruction to the structured light camera; after the scanning finished, the point cloud and the joints of the holder are send to the virtual reality system to reconstruct the virtual scene. The coordinate transformation result of the point cloud and the reconstruction result is shown in Fig. 5, and Fig. 6 shows the reconstruction result when a complex shape obstacles suddenly appeared.

The reconstruction experiments indicate that the proposed method can reconstruct the virtual scene under unstructured environment on line.

4.2 Virtual Force Feedback

When calculating the virtual force, the inertia tensor of the robot links was calculated by numerical integration based on their 3D model. Due to the visual residue phenomenon, the operator will feel the video continuous when its refresh rate is

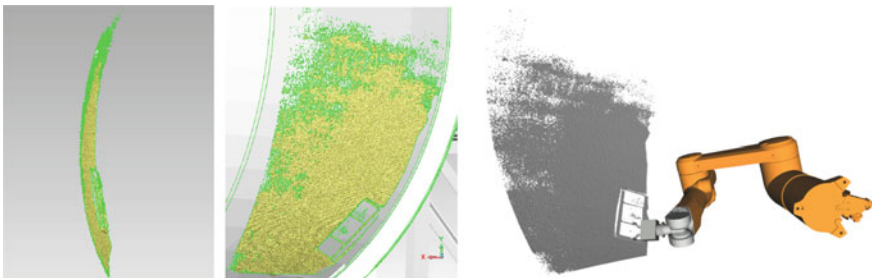


Fig. 5 Virtual scene reconstruction result

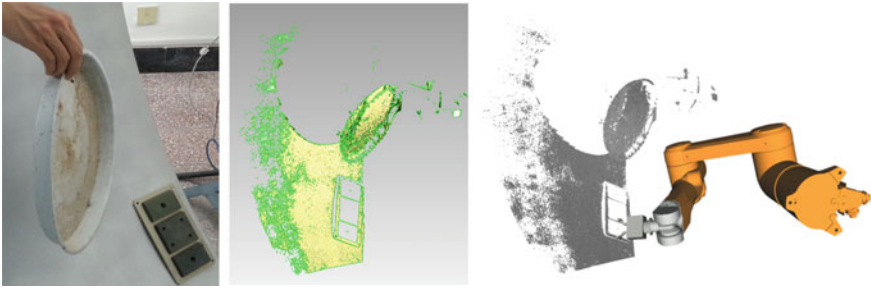


Fig. 6 Virtual scene reconstruction result when an obstacle appeared

more than 24 Hz. However, if we want the operator feel a continuous force, the refresh rate of the feedback force should more than 200 Hz. In order to improve the frequency of the force feedback, a separate thread is created for physical simulation. The GUI refresh rate and the physical simulation frequency are shown as follows:

From Fig. 7 we can see that both the GUI refresh rate and the force feedback rate meet the continuous perception demand of human.

When calculating the virtual contact force, the first and second order differential of the joints is used, so it's very sensitive to joint jitter. However, the jitter of the operator is inevitable when operating the master manipulator, so the contact force calculated from the dynamical model jittering heavily, as Fig. 8a.

In order to keep the feedback force smooth, a low pass filter is designed, and the cutoff frequency of the filter is determined by the FFT result of the virtual force, as Fig. 8b. And the virtual feedback force after filtered is shown in Fig. 8c. Restricted by the flexible and response time of the operator, the shaking value of the contact force is bigger at the moment of the collision occurs, and the value is bigger when the contact force is big, as Fig. 8d. Because of the algorithm proposed in this paper doesn't need the dynamics of the environmental, and the geometry is reconstructed on line, so it can be used in unstructured environment.

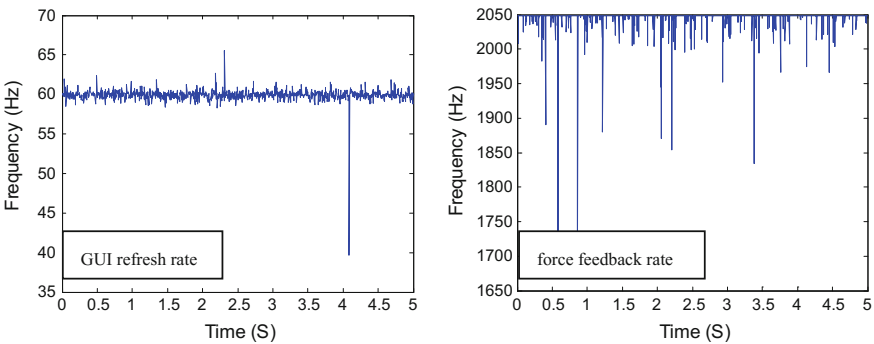


Fig. 7 Simulation rate of the predictive display system

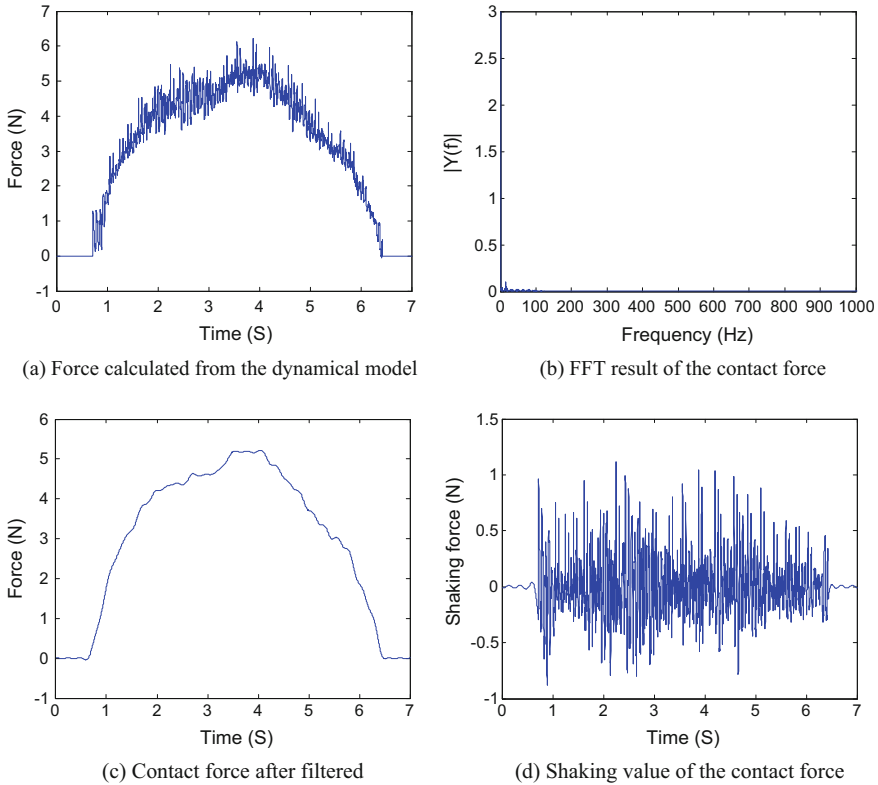


Fig. 8 **a** Force calculated from the dynamical model. **b** FFT result of the contact force. **c** Contact force after filtered. **d** Shaking value of the contact force

5 Conclusion

This article examines the predictive method under unstructured environment, and a novel predictive display method based on 3D scene reconstruction online is proposed. In this method, the slave environment is monitored by camera, and the virtual environment is modeled and reconstructed online, so the consistency with the real world can be ensured. For denoising the point cloud, an adaptive random sample consensus (ARANSAC) algorithm is proposed, which adjusts the parameters of the standard RANSAC algorithm adaptively; On the force calculation, the rigid contact model and the manipulator's dynamics are used to calculate the virtual force, which eliminates the time delay and security problem introduced by the feedback force and increases the frequency of force feedback; In our program, the physical simulation and graphic simulation are calculated in different thread, which ensures the GUI refresh rate and the force feedback rate are meet the continuous

perception demand of human, and a low frequency filter is designed to keep the virtual feedback force smooth. At last, the effectiveness of the proposed method is confirmed by experiments.

References

1. Aiguo, S.: Force telepresence telerobot (1): review of the history and development. *J. Nanjing Univ. Inf. Sci. Technol.* **5**, 1–19 (2013)
2. Woo-Keun, Y., Goshozono, T., Kawabe, H., Kinami, M., Tsumaki, Y., Uchiyama, M., et al.: Model-based space robot teleoperation of ETS-VII manipulator. *IEEE Trans. Robot. Autom.* **20**, 602–612 (2004)
3. Li, H., Song, A.: Virtual-environment modeling and correction for force-reflecting teleoperation with time delay. *IEEE Tran. Ind. Electron.* **54**, 1227–1233 (2007)
4. Polat, I., Schere, C.: Stability analysis for bilateral teleoperation an IQC formulation. *IEEE Trans. Robot.* **28**, 1294–1308 (2012)
5. Anderson, R.J., Spong, M.W.: Bilateral control of teleoperators with time delay. *IEEE Trans. Autom. Control* **34**, 494–501 (1989)
6. Mollet, N., Chellali, R., Brayda, L.: Virtual and augmented reality tools for teleoperation: improving distant immersion and perception. *J. Trans. Edutainment II*(5660), 135–159 (2009)
7. Hokayem, P.F., Spong, M.W.: Bilateral teleoperation: an historical survey. *J. Automatica* **42**, 2035–2057 (2006)
8. Polushin, I.G., Dashkovskiy, S.N., Takhmar, A., Patel, R.V.: A small gain framework for networked cooperative force-reflecting teleoperation. *J. Automatica* **49**, 338–348 (2013)
9. Mengyin, F., Xianwei, L., Tong, L., Yi, Y., Xinghe, L., Yu, L.: Real-time SLAM algorithm based on RGB-D data. *J. Robot.* **37**, 683–692 (2015)
10. Jingzhou, S., Yang, B., Hanxu, S., Qingxuan, J., Xin, G., Yifan, W.: A decoupled force feedback haptic device based on interaction force signal. *J. Robot.* **36**, 477–484 (2014)
11. Edwards, L., Sims, M., Kunz, C., Lees, D., Bowman, J.: Photo-realistic terrain modeling and visualization for mars exploration rover science operations. In: 2005 IEEE International Conference on Systems, Man and Cybernetics, pp. 1389–1395, Waikoloa, HI (2005)
12. Fan, Q., Xiaoxia, X., Libo, C.: The research about modeling technology of irregular virtual reality scene based on point cloud data. In: International Conference on Computer Application and System Modeling (ICCSM), pp. 538–541, Taiyuan, China (2010)
13. Seong-Oh, L., Hwasup, L., Hyoung-Gon, K., Sang Chul, A.: RGB-D fusion: real-time robust tracking and dense mapping with RGB-D data fusion. In: International Conference on Intelligent Robots and Systems (IROS 2014), pp. 2749–2754, Chicago, IL, USA (2014)
14. Shellshear, E., Berlin, R., Carlson, J.S.: Maximizing smart factory systems by incrementally updating point clouds. *IEEE Comput. Graph. Appl.* **35**, 62–69 (2015)
15. Turner, E., Cheng, P., Zakhor, A.: Fast, automated, scalable generation of textured 3D models of indoor environments. *IEEE J. Sel. Top. Signal Process.* **9**, 409–421 (2015)

Control System Design and Implementation of Flexible Multi-joint Snake-Like Robot for Inspecting Vessel

Yi Liu, Qingquan Yan, Qiang Zhang, Weibin Guo, Odbal,
Baoyuan Wu and Zengfu Wang

Abstract In nuclear fusion research, inspecting tokamak vessel by controlling remote-handling robot is promising. To achieve collision-free, precise robot movement in tokamak vessel, it is necessary for the control system to perform motion planning, closed-loop control, interface operation and other functions. In this paper, a control system based on Ethernet and CAN network for a developed snake-like robot for inspecting tokamak vessel is proposed. Then, the robot kinematics model is built. Motion planning procedure is completed by a stepwise iterative algorithm. And a dual-loop control method for each joint is realized. A control software is developed to perform control computation and interface operation. Finally, the effectiveness of the control system for the robot is verified by actual experiments.

Keywords Snake-Like robot · Motion planning · Dual-Loop control

1 Introduction

In nuclear fusion research, tokamak vessel is a key experimental device. To ensure the vessel working properly, it requires regular inspection in the vessel. Due to large numbers of equipments inside the vessel, complex pipelines, narrow work space, and toxic parts polluted by radioactive and toxic substances, it is improper to contact toxic components by hand during maintenance of the vessel. Remote handling robot used in nuclear fusion environment is in urgent demand. Many researchers have made much ongoing in-depth research, and acquired a series of

Y. Liu · Q. Yan · Q. Zhang · W. Guo · Odbal · B. Wu · Z. Wang (✉)
Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, China
e-mail: zfwang@ustc.edu.cn

Y. Liu
e-mail: yliu@iim.ac.cn

Z. Wang
Department of Automation, University of Science and Technology of China, Hefei, China



Fig. 1 The snake-like robot for inspecting tokamak vessel

important achievement. Harrist et al. [1] and Loving et al. [2] introduced a JET remote handling system including two robot arms to complete much maintenance and installation work. Gargiulo et al. [3, 4] developed an inspection robot named articulated inspection arm (AIA) and completed inspection of the vessel under UHV conditions. Neri et al. [5] developed a laser in vessel viewing system (IVVS) for ITER. Peng et al. [6] designed a remote inspection system called flexible in-vessel inspection system for the plasma facing components (PFCs). Wang et al. [7] introduced 12 DOFs remote handling robot used for inspecting the working state of the ITER-like vessel.

We have developed a flexible multi-joint snake-like robot for inspecting tokamak vessel, as shown in Fig. 1. Light titanium alloy material is adopted as its body material. The robot consists of nine joints, one pantilt, one mobile platform and one fiber winch. The total length of the robot is 5020 mm. Each joint is driven by a DC motor and a planetary gearhead. The max advancing speed of the robot can reach 50 mm/s.

In this paper, a control system based on Ethernet and CAN network for the developed snake-like robot is proposed. First, the control system architecture is introduced. Then, main structure of the robot is presented and the kinematics model is built. Motion planning procedure is achieved by using a stepwise iterative algorithm. For each joint, a dual-loop control method is designed and applied by using absolute position sensors. A control software is developed in windows operation system, which performs control computation, status display and interface operation. Finally, related verification experiments are made in a simulation vessel. The effectiveness of the control system for the robot is verified.

2 Control System Architecture

An Ethernet and CAN-bus based network control structure is adopted in the control system, as shown in Fig. 2. It mainly composed of one industrial computer, one multi-axis controller, motor drivers, dc motors, photoelectric switches, absolute

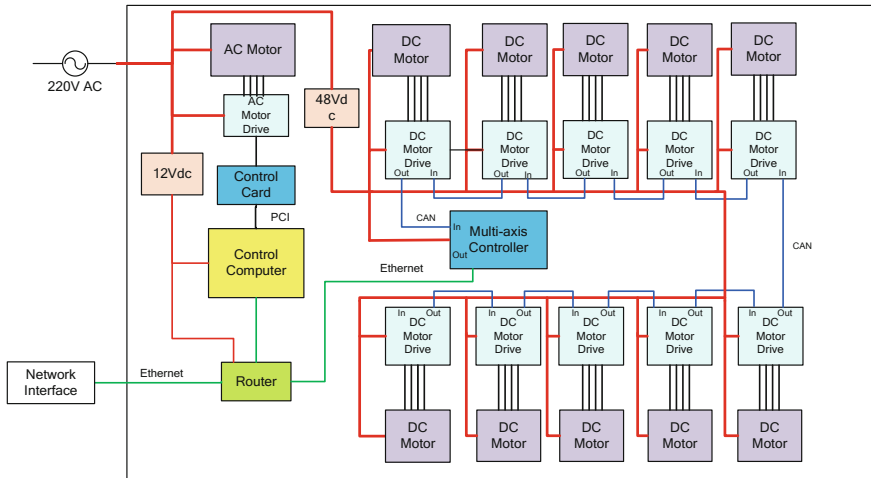


Fig. 2 Control system block diagram

position sensors, routers and other components. The controlled objects include eight robot arms, one camera pantilt, one mobile platform, and one fiber winch. The motion of mechanical components is accomplished by using motors and planetary gears. In order to improve accuracy of joint control, absolute position sensors are used to measure rotation angle of each joint.

For the robot arms, camera pantilt and fiber winch, multi-axis control structure based on CAN bus network is used, composed of control computer, multi-axis controller, motor drives. For the mobile platform, control structure based on PC plus motion control card is used, composed of control computer, motion control card and motor drives.

To simplify design process and reduce the influence of the uncertainty of the dynamic model on the control algorithm, kinematics model of the robot is established and used to control the robot's motion. Control software runs in the control computer to perform command parsing, path planning, trajectory control, and status monitoring. Through Ethernet and CAN-bus, Output instructions are translated into control instruction of motor drives by multi-axis controller and motion control card.

3 Kinematics Model

The snake-like robot with redundant degrees of freedom as shown in Fig. 3 comprises six degrees of freedom J1–J6 in the horizontal direction, used for deployment and obstacle avoidance. And it also comprises two degrees of freedom J7–J8 in the vertical direction, used to lifting the end of the arm.

Fig. 3 Kinematics model of the snake-like robot

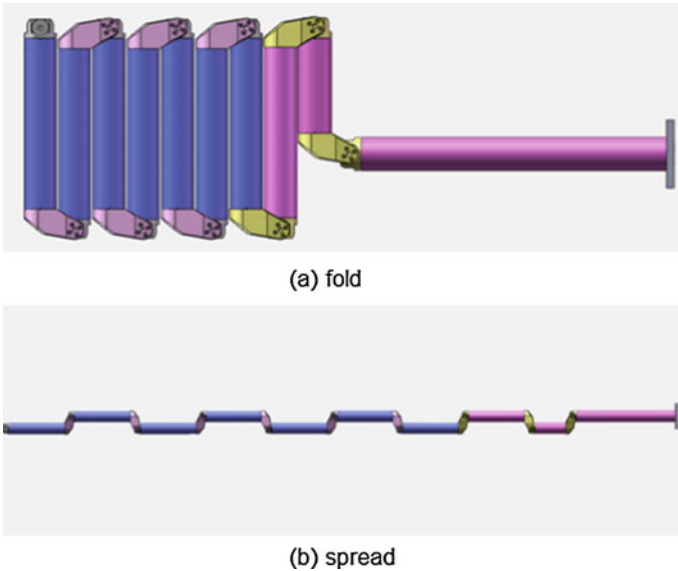
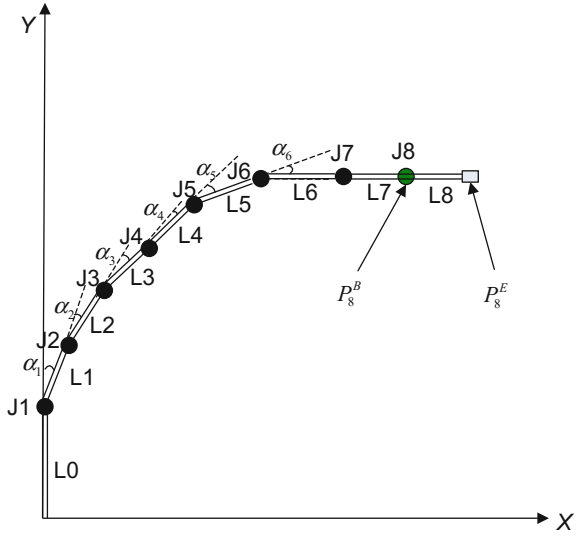
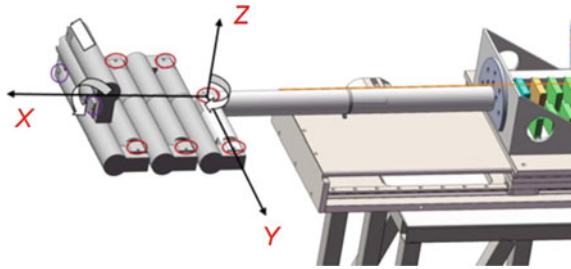


Fig. 4 Folding and spreading status of the robot

By offsetting adjacent links, the snake-like robot can fold and spread as shown in Fig. 4.

The global coordinate system is set as shown in Fig. 5.

Fig. 5 The set of global coordinate system



Because of the offset between two adjacent links, the ending coordinates of the current link does not coincide with the starting coordinates of the next link. It is necessary to calculate the starting and ending coordinates of each link respectively.

The coordinate equations of starting point and ending point of n-th link are shown as following:

$$\left\{ \begin{array}{l} x_n^B = x_1^B + L_a \sum_{i=1}^n (-1)^{i-1} \sin(\theta_i) + \sum_{i=1}^{n-1} L_i \cos(\theta_i) \\ y_n^B = y_1^B + L_a \sum_{i=1}^n (-1)^i \cos(\theta_i) + \sum_{i=1}^{n-1} L_i \sin(\theta_i) \\ z_n^B = z_1^B + \sum_{i=1}^{n-1} L_i \sin(\lambda_i) \\ \theta_i = \sum_{j=1}^i a_j \end{array} \right. \quad (1)$$

$$\left\{ \begin{array}{l} x_n^E = x_1^B + L_a \sum_{i=1}^n (-1)^{i-1} \sin(\theta_i) + \sum_{i=1}^n L_i \cos(\theta_i) \\ y_n^E = y_1^B + L_a \sum_{i=1}^n (-1)^i \cos(\theta_i) + \sum_{i=1}^n L_i \sin(\theta_i) \\ z_n^E = z_1^B + \sum_{i=1}^n L_i \sin(\lambda_i) \\ \theta_i = \sum_{j=1}^i a_j \end{array} \right. \quad (2)$$

where (x_n^B, y_n^B, z_n^B) and (x_n^E, y_n^E, z_n^E) are respectively the starting coordinate and ending coordinate of n-th link, L_a is offset distance between adjacent links, (x_1^B, y_1^B, z_1^B) is the starting coordinates of 1st link, α_i is the rotary angle of the i-th joint, θ_i is the angle between the i-th link and X axis, and λ_i is the angle between the i-th link and horizontal plane.

4 Motion Planning Algorithm

According to the starting position and the desired position, the position error, the movement distance of the robot and the risk of collision are selected as a check condition. A stepwise iterative algorithm is used to search a collision-free path along the center line of the vessel for the robot.

The vessel is vertically symmetrical annular body. The intermediate position in the vertical direction is selected as the working surface for the snake-like robot, in order to ensure maximum area. According to a known model of the vessel, two dimensional plane model of the vessel is established, namely a plane ring model.

The motion progress of the snake-like robot from initial full-folded posture to final inspecting posture in the vessel can be divided into four stages as shown in Fig. 6.

(1) Spreading stage.

Initially the snake-like robot is fully folded. The position increment of the robot is calculated step by step. First, rotate the n -th joint so that the end of the expanded link approaches the center line of the entrance. Then continue to rotate the $(n-1)$ -th joint behind, and at the same time adjust the rotating angle of the preceding joints to keep the end of links near the center line of the entrance. Successively spread each joint as so on.

(2) Straightening stage.

After the snake-like robot enters into the vessel, the link which enters into the vessel currently is straightened, nearly parallel to the X axis, by rotating related joint.

(3) Stage of moving along the center line of the vessel

To ensure smooth change in the posture of the snake-like robot forward, after the snake-like robot enters into the vessel, the end point of each link moves along the center line of the vessel. When the robot moves a step, rotate each joint to make the end of each link move near the center line through multiple iterations of calculation.

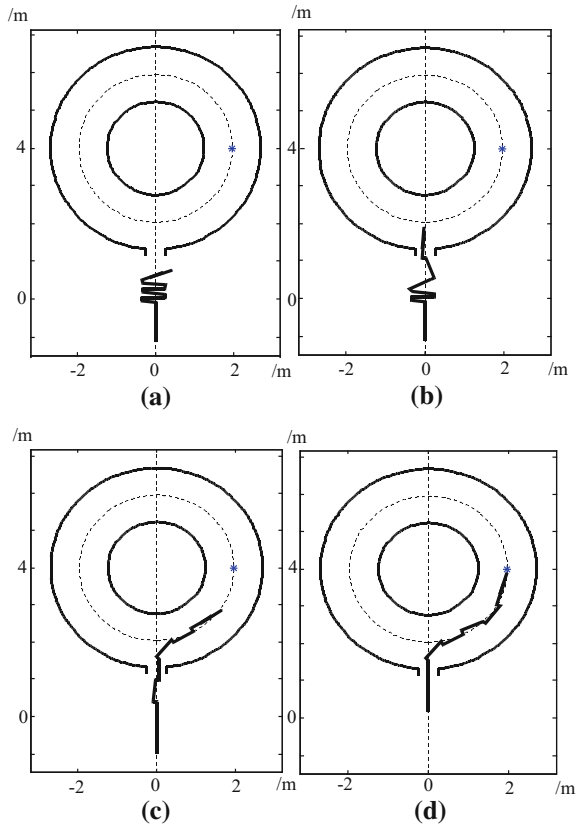
(4) Stage of translation moving

After the snake-like robot is fully deployed, if the robot needs to continue in-depth, the mobile platform starts regular motion. The snake-like robot is pushed into the vessel inside. While adjusting the posture of the robot by rotating each joint, the robot continues to move along the center line of the vessel.

After the snake-like robot moves to the designated location, the pantilt starts scanning motion, guiding the camera to observe the surface along the cross section of the center line of the vessel.

In the actual planning calculations, since the snake-like robot body lies in different areas of the vessel, calculation of four stages is simultaneously executed. To simplify operation, the joint angles which increase or decrease with a fixed amount, varies according to the direction of adjustment.

Fig. 6 Sketch map of four stages. **a** Spreading. **b** Straightening. **c** Moving along the centre line. **d** Translation moving



In each step of the planning process, calculate the distance between the end point of snake-like robot and the desired goal. When the distance between the end point of snake-like robot and the desired goal is smaller than the set minimum distance value, stop path planning.

5 Joint Dual-Loop Control

Because of the large reverse gap from the rope-driven joints, to achieve precise joint position motion, one capacitance-type absolute position sensor is installed at each joint to obtain the absolute position of the joint, as shown in Fig. 7. Therefore, a joint dual-loop control method, including joint position loop control module based on absolute position sensor and motor speed loop control module based on motor incremental position encoder, is designed and applied.

In the joint position loop control module as shown in Fig. 8, the absolute position of joint is controlled by a PI controller according to feedback position of



Fig. 7 Joint equipped with absolute position sensor

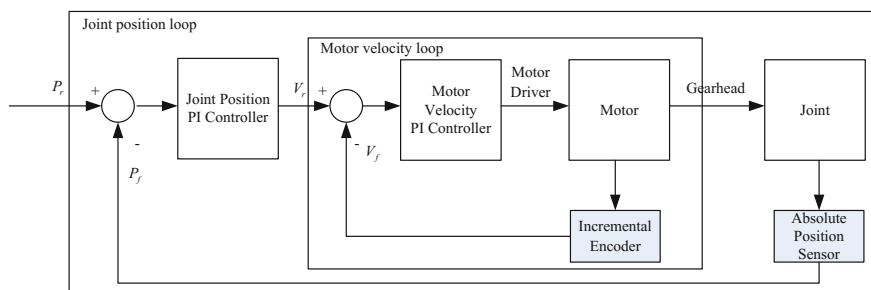


Fig. 8 Joint dual-loop control diagram

the absolute position sensor, and the output speed command is sent to the motor speed loop module. In the motor speed loop control module, based on the feedback speed information from the motor position control increment encoder and the speed command from the joint position loop module, and another PI controller is used to control the speed of the motor. After uses of joint dual-loop control method, positioning accuracy of each joint angle is improved significantly, which can reach 0.1° .

6 Control Software

In the control software, windows dialog structure is used to perform a response mechanism including timers, multi thread and buttons, as shown in Fig. 9. The main function of the control software comprises command parsing, path planning, trajectory control and status monitoring. Three control modes are adopted, including manual mode, semi-automatic mode and automatic mode. In manual

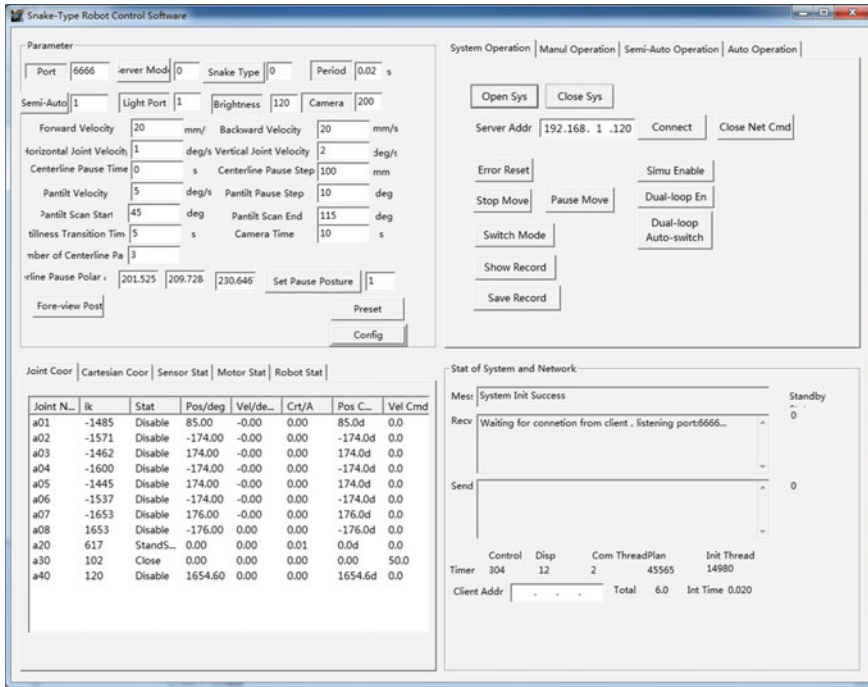


Fig. 9 Control software interface

mode, a remote operator directly controls the movement of each axis by using a joystick. In semi-automatic mode and automatic mode, according to the instructions from the primary interface software, path and trajectory of the robot are generated automatically. Then multi-axis coordinated motion of the robot is performed automatically.

7 Experiment Verification

In a simulation vessel, as shown in Fig. 10, an experiment of deployment and observation of the snake-like robot is designed and completed. It takes about 12 min to finish deployment and observation in the vessel. The position tracking curves of each joint are shown in Fig. 11, and steady positioning error of each joint is not more than 0.1°.

The repeated positioning error of the robot is measured by using a total station while the robot deploy in the simulation vessel. By measuring the relative coordinates of the end position of the robot with respect to the base, the smallest envelope spherical radius of the relative coordinates obtained by multiple measurements is not more than 7.8 mm, which showed that the repeated positioning error is not more than 7.8 mm.



Fig. 10 The robot moving in the simulation vessel

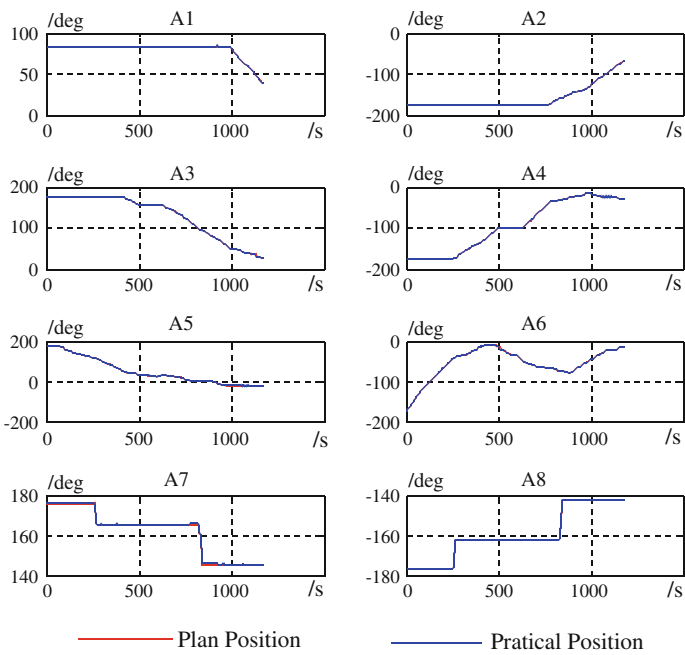


Fig. 11 The position tracking curves of each joint

8 Conclusion

In this paper, a control system based on Ethernet and CAN network for a developed snake-like robot for inspecting tokamak vessel is proposed. Then, the robot kinematics model is built and used for motion planning. Motion planning procedure is performed by a stepwise iterative algorithm. And a full closed-loop control method for each joint is implemented. A control software is developed to perform these functions. Finally, the effectiveness of the control system for the robot is verified by actual experiments. For further improvement of future nuclear remote handling robot, it has certain reference value.

Acknowledgements This research work was supported by the China National Magnetic-Confinement Fusion Energy R&D Program (ITER) and funded by the National Science and Technology Major Project of the Ministry of Science and Technology of China (No.2012GB102007) and the National Natural Science Foundation of China (NSFC) under Grant No.61503361. The authors wish to thank the research group for their great contributions to the presented research. The views and opinions expressed in this paper are the sole responsibility of the authors.

References

1. Haist, B., Mills, S., Loving, A.: Remote handling preparations for JET EP2 shutdown. *Fusion Eng. Des.* **84**, 875–879 (2009)
2. Loving, A., Allana, P., et al.: Development and application of high volume remote handling systems in support of JET and ITER. *Fusion Eng. Des.* **87**, 880–884 (2012)
3. Gargiulo, L., Bayetti, P., et al.: Operation of an ITER relevant inspection robot on Tore Supra tokamak. *Fusion Eng. Des.* **84**, 220–223 (2009)
4. Perrot, Y., Gargiulo, L., et al.: Long-reach articulated robots for inspection and mini-invasive interventions in hazardous environments: recent robotics research. *Qualification Test. Tool Dev. J. Field Robot.* **29**, 175–185 (2012)
5. Neri, C., Bartolini, L., et al.: The laser in vessel viewing system (IVVS) for Iter: test results on first wall and divertor samples and new developments. *Fusion Eng. Des.* **82**, 2021–2028 (2007)
6. Peng, X.B., Yuan, J.J., et al.: Kinematic and dynamic analysis of a Serial-link robot for inspection process in east vacuum vessel. *Fusion Eng. Des.* **87**, 905–909 (2012)
7. Wang, H., Lai, Y., et al.: Design and implementation of motion planning of inspection and maintenance robot for ITER-like vessel. *Fusion Eng. Des.* **101**, 111–123 (2015)

Research on the Tokamak Equipment CAsk (TECA) for Remote Handling in Experimental Advanced Superconducting Tokamak (EAST)

Lifu Gao, Weibin Guo, Baoyuan Wu, Daqing Wang, Yuan Liu, Yi Liu, Qiang Zhang, Zengfu Wang and Liangbin Guo

Abstract The tele-operated transfer cask is required for maintenance in nuclear fusion environment. A type of the transfer cask system, namely TECA (Tokamak Equipment CAsk) is designed for Experimental Advanced Superconducting Tokamak (EAST). The whole TECA system schemes, composed of the Air Cushion Vehicle (ACV), the Docking Pallet (DP), and the Conservation Cask (CC), are designed, and the prototype is accomplished. The motion and control of the ACV, the DP, and the DSD are detailedly analyzed. The motion control experiments validate the TECA applicable.

Keywords Experimental Advanced Superconducting Tokamak (EAST) · Transfer cask system · Mechanism design · Motion control

1 Introduction

Controlled nuclear fusion can be one of the most promising solutions to the global energy crisis. To achieve sustained nuclear fusion reaction, the conditions are very harsh, such as high temperature, high vacuum, and strong magnetic field. Furthermore, Plasma Facing Components (PFCs) in Experimental Advanced Superconducting Tokamak (EAST) [1] consist of beryllium. Thus Lots of neutrons and

L. Gao (✉) · W. Guo (✉) · B. Wu · D. Wang · Y. Liu · Y. Liu · Q. Zhang · Z. Wang
Institute of Intelligent Machines (IIM), Hefei Institutes of Physical Science (HIPS), Chinese Academy of Sciences (CAS), Hefei 230031, Anhui, China
e-mail: lifugao@iim.ac.cn

W. Guo
e-mail: wbguo@iim.ac.cn

L. Gao · D. Wang · Y. Liu · Q. Zhang · Z. Wang
Department of Automation, University of Science and Technology of China, Hefei, Anhui 230026, China

L. Guo
Jurong Smartech Display Co., Ltd, Jurong, Jiangsu 212400, China

gamma rays would be generated during plasma operation, and meanwhile, the in-vessel components can be activated and polluted. Moreover, the fuel tritium will permeate into PFCs or attach to materials surface. And the dust can spread into air easily. Therefore, to prevent the pollution and radiation from harming operators, Remote Handling (RH) technology [2–5] is adopted for in-vessel components handling. A type of Transfer Cask System we designed, namely TECA (Tokamak Equipment CAsk), is used for transferring in-vessel components need be maintained and exchanged between the Vacuum Vessel (VV) and the Hot Cell (HC) by remote control. The in-vessel components, such as port plug, are very large and heavy. Besides, during the docking and sealing stage, the TECA should be controlled precisely to ensure leak proof between the TECA and the port, when the in-cask tractor perform installing or removing tasks.

International Thermonuclear Experimental Reactor (ITER) plan, beginning from 1980s, is one of the most ambitious energy projects in the world today. Our groups in the Hefei Institutes of Physical Science (HIPS), Chinese Academy of Sciences (CAS) have already did many pre-research [6–9]. Review on the Transfer Cask System, Cheng et al. [10] studied the Double Seal Door (DSD) system and structure design according to three types of Vacuum Vessel Port (VVP) and made FEM (Finite Element Method) analysis for key components, to provide technique reference for engineering. Pan et al. [11] studied the control system of remote transfer CASK, and provided the detailed scheme including the precise collimation for large CASK. Qin et al. [12, 13] studied the DSD motion processes and kinematics by ADAMS, and analyzed the hydraulic pole and guide rail force. They obtained the stress distribution and optimized the guide rail to guide the manufacture and also other ITER components analysis. Nevertheless, most of those are focus on single module and theoretical analysis, yet the research for whole system and functional prototype is little. Currently, the design and control performances of Air Cushion Vehicle (ACV) and Docking Pallet (DP) are little discussed. The VVP independent sealing and automatic attaching functions of the DSD haven't been solved clearly. Moreover, the door body must be penetrated through so as to reduce the sealing effects. Focusing on heavy payload transferring, precise docking, and double sealing, we proposed a scheme for ACV design and control. A novel parallel lifting mechanism is adopted for precise docking control. Also, a DSD structure with independent sealing and automatic attaching function is presented, and, it can also improve the sealing effects.

2 Mechanism and Control Scheme for TECA

2.1 TECA Construction and Motion Control Sequences

As shown in Fig. 1, the TECA primarily consists of the Conservation Cask (CC), the Docking Pallet (DP), and the Air Cushion Vehicle (ACV). The Air Cushion Vehicle (ACV) is used for transporting the heavy payloads, as, the DP, the CC, and

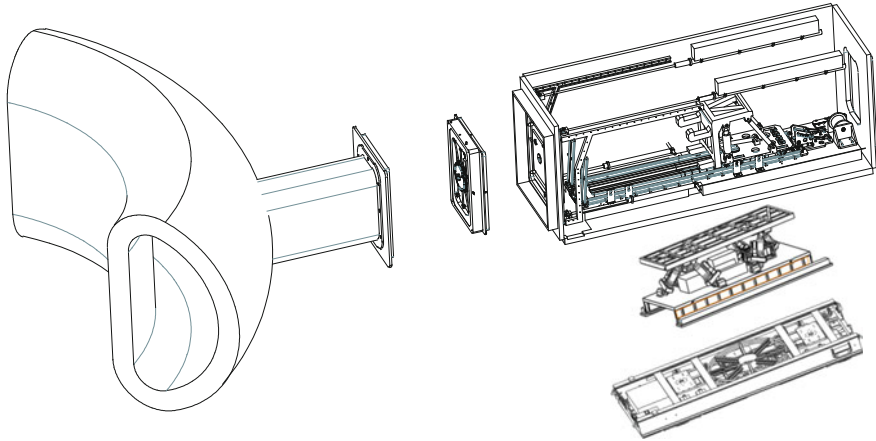


Fig. 1 System configuration for TECA

also the in-vessel components. The Conservation Cask (CC), mainly composed of the cask envelope, the Double Seal Door (DSD), the In-cask Tractor (IT), and other auxiliary, is used to keep the in-vessel components and the harmful dust from outside. The Docking Pallet (DP) is used for supporting the CC and precisely docking between the DSD and the Vacuum Vessel Port (VVP). The Double Seal Door (DSD) consists of the Maintenance Door (MD) and the Cask Door (CD). The MD is used to seal the VVP after the port plug removed; and The CD is used to seal the cask envelope, preventing toxics from spreading into air. In-cask Tractor (IT) is installed in the CC for operating.

The DSD Tilting mechanism, composed of two top and two bottom linear cylinders, is used for DSD opening and closing. Before the In-cask Tractor (IT) starts to operate the port plug, the DSD must dock with the VVP precisely and seal between them by the Docking Pallet (DP). And then the DSD is opened by the linear cylinders and tilted up to the ceiling of the cask along the flexural guide rail. When the In-cask Tractor (IT) tasks completed, the DSD will be closed again. Yet, the DSD separates into the MD and The CD. The MD is on the VVP, and The CD is in the CC, for seal respectively. Afterwards, the TECA transfers the port plug to the Hot Cell (HC).

Due to working scenarios of thermonuclear fusion task, the whole TECA motion control sequences are depicted in Fig. 2. Before the maintenance task, the plug is on the VVP, and the DSD is on the TECA. When maintenances start, the TECA is transferred to the Vacuum Vessel (VV) by controlling the ACV. When the setting position is approached, the ACV stops. In step①, by controlling the DP, the DSD makes alignment and docking with the VVP, to complete sealing between them. The sensors return the cask position and orientation to the TECA central control module. If the alignment cannot be satisfied, the DP adjustment goes on. When the connect control finishes, the DSD will be opened and tilted upward in step②. The sensors return the DSD pose to the DSD central module. If the open control finishes, the IT will perform the maintenance tasks.

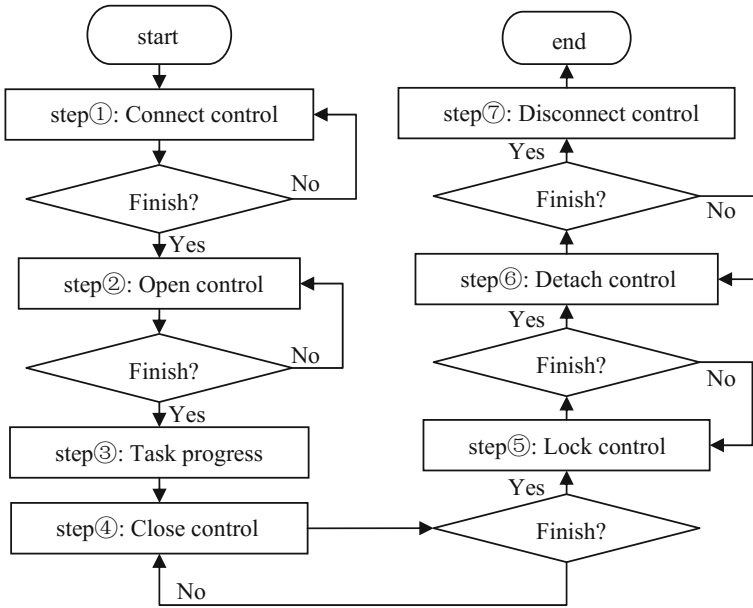


Fig. 2 TECA control flow: connect control- the DSD docking with the VVP and sealing between them, open control—the DSD is tilted upward to the CC ceiling, task progress—the IT operates the plug, close control—the DSD is tilted downward to seal both the TECA and the VVP, lock control- the MD locks and seals the VVP, Detach control—the MD is detached from the CD, disconnect control- the CC disconnects with the VVP and unseals

When the IT finishes approaching the port plug and pulling it into the cask envelope in step③, the DSD will be closed in step④. The DSD pose is feed back to the DSD central module. If the close control finishes, the VVP will be locked and sealed by the MD in step⑤. The MD connecting rod[] stretching position is gathered by the MD central module. If the desired stretching position is achieved, the lock movement is finished. When the MD completes locking and sealing the VVP, the DSD will be separated into the MD and The CD In step⑥. The CD connecting rod shortening position is gathered by the CD central module. When the desired position is obtained, the CD finishes detaching the MD. Then the CC will disconnect from the VVP in step⑦. The sealing between the VVP and the CC will be released. At the end, the TECA transfers the port plug to the HC. The motion control is accomplished.

2.2 Air Cushion Vehicle (ACV) Construction

The ACV primarily consists of five system as, air cushion drive sys., walking and steering sys., hose reeling sys., navigation sys., and automatic charging sys., as shown in Fig. 3. Based on heavy load (over 30 ton) and high navigation accuracy (−10 to 10 mm), we made detailed design and analysis [14].

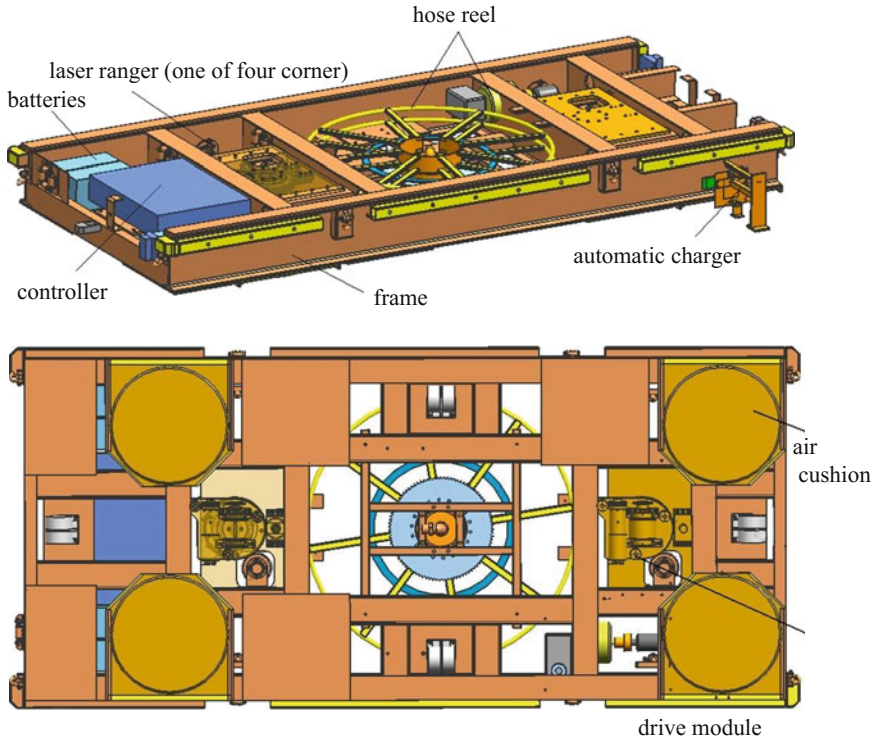


Fig. 3 The ACV construction

2.3 Docking Pallet (DP) Construction

The DP, which can realize six DOF movement, is designed for the DSD aligning and docking with the VVP. In Fig. 4, the DP primarily consists of three parts as, the Upper Mounting Plate, six Linear Cylinders, and the Lower Platform. The Upper Mounting Plate is designed to fix the CC. The Lower Platform is used as suspension system transferred by the ACV. Six Linear Cylinders are designed to realize the CC displacement and rotation in three directions.

2.4 Conservation Cask (CC) Construction

The CC is used for operating and preserving in-vessel components from leak outside, when the TECA transferring between the VV and the HC, as in Fig. 1. The cask envelope is designed for supporting the housing space and ensuring leak proof. The DSD is designed for automatic sealing when docking together or disconnecting

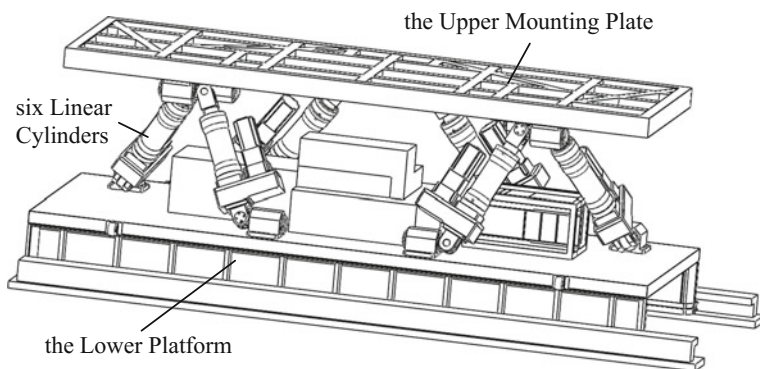


Fig. 4 Docking pallet mechanism

apart. The IT is designed for lifting the port plug, and dragging into the CC, when the CC is docking and sealing between the VVP.

2.5 *Double Seal Door (DSD) Construction*

The DSD [15] mainly consists of the MD, the CD, and the Tilting mechanism for opening and closing the DSD. As mentioned, the DSD is needed to seal the TECA and the VVP independently. Also, it's needed to dock between the TECA and the VVP automatically. Thirdly, it's needed to attach and detach between the MD and the CD automatically. Fourthly, it's needed to tilt upward when the tractor works. Besides, it's needed to seal when the TECA and the VVP are connected together.

Currently, the MD is operated via transmission by the CD when attached. While, we designed the MD which can independently lock and seal when detached from the CD. In the CD&MD design, the swing type driving and wedge Lock mechanism is adopted for sealing, and also, the probe-cone docking mechanism is used for attaching. The Tilting mechanism mainly consists of the flexural guide rail, the top and bottom linear drive. One end of the top drive is hinged with the DSD, and the other is fixed on the cask envelope. The two ends of the bottom drive are hinged with the DSD and the cask envelope respectively. The DSD moves along the flexural guide rail via four rollers.

2.6 *In-Cask Tractor (IT) Construction*

The IT mainly consists of three systems as, the lifting sys., the guide rail sys., and the dragging sys. The lifting sys. is composed of the holder frame, the linkage mechanism and the linear drive cylinder. The lifting sys. is mainly used for tilting

and holding the port plug before the dragging movement. The cable reel mechanism is adopted in the dragging sys. The port plug is pulled in or pushed out of the CC by reeling the cable. The guide rail sys., composed of the fixed part and the movable part, is used for guiding the dragging movement. The fixed part is used inside the CC. while the moveable part is used out of the CC, when the connect control finished.

3 Kinematics Analysis and Motion Control

Based on the structure and motion requirement, the whole TECA trajectory is primarily composed of three stages. The first stage is the transferring period by the ACV, between the VV and the HC. The second is the alignment and docking period by the DP, between the CC and the VVP. The third is the sealing and the task period by the DSD and the IT, in the CC.

3.1 Air Cushion Vehicle Control

The ACV control is mainly composed of the air cushion control and the navigation control. The cushion control [14] is of great concern for the precise attitudes control and stable loading during the transfer process. The air cushion chart is schematically listed in Fig. 5. Combining with laser rangers and force sensors, adaptive attitude of air bearings could be fine-tuned by PLC and servo valves. Furthermore, it will stimulate the research of employing some control methods for minimizing consumed energy.

3.2 Docking Pallet Control

The DP control is used for alignment and docking between the CC and the VVP. There are four range sensors fixed under the bottom of the CC four corners. Thus, the parameters of the CC height to the ground, the CC roll angle, and also the CC pitch angle can be estimated. Besides, there are eight range sensors fixed on the CC front panel. Two of them are mounted for sensing the upward; also, four of them are mounted for sensing the forward; besides, the left two of them are mounted for sensing the sideward. Accordingly, the three rotation angle and the three displacement of the DSD can be estimated. The DP kinematics [16] are analyzed, and the control chart is shown in Fig. 6.

The DP control experiments were performed and the results were collected for analysis. The repeated positioning accuracy is measured by x, y, z- axis respectively. Aiming at one specified pos, control the DP moving to the destination

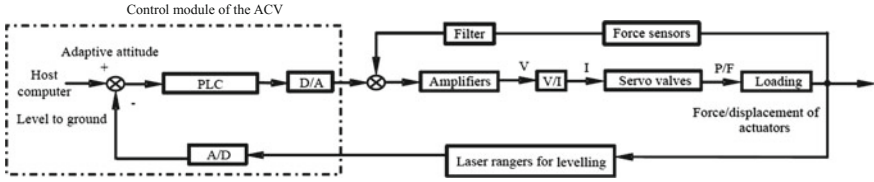


Fig. 5 The ACV air cushion control

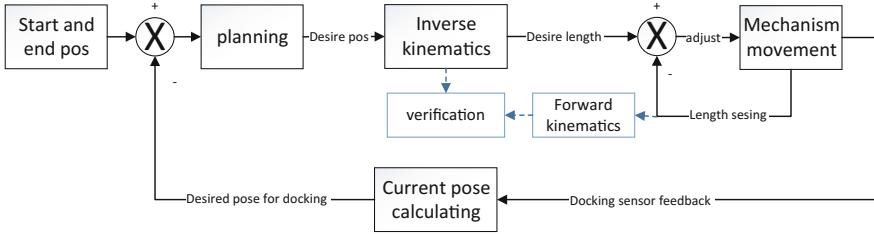


Fig. 6 The DP control chart

several times. The vernier caliper (brand: fushishan, accuracy: 0.02 mm) and the auxiliary tools (standard gauge block with thickness of 15.00 and 60.00 mm) is utilized for measuring the reference distance. Once for five points, 10 pairs data were measured for each one. The whole 150 pairs data were measured. Choosing 3 pos, the max error is listed in Table 1. The repeated positioning accuracy is about 0.88 mm along three axis, and the precision docking with the VVP can be accomplished by the design mechanism.

3.3 Double Seal Door Control

The DSD control primarily consists of the MD control, the CD control, and the DSD control. The MD control is used for the sealing between the MD and the VVP. The CD control is used for the sealing between the MD and the CD, and, also, for the attaching or detaching between the CD and the MD. Besides, the DSD control is used for the DSD opening or closing by the Tilting mechanism, and also for the sealing between the CD and the CC. The DSD is a very heavy load, and

Table 1 Repeated positioning accuracy measuring

Max error	x- axis ($\times 10^{-2}$ mm)					y- axis ($\times 10^{-2}$ mm)					z- axis ($\times 10^{-2}$ mm)				
1	38	78	56	38	52	72	30	72	52	64	20	20	16	54	48
2	22	18	38	32	42	74	88	60	34	82	80	60	16	28	20
3	20	4	14	22	30	10	12	20	48	14	8	14	10	24	20

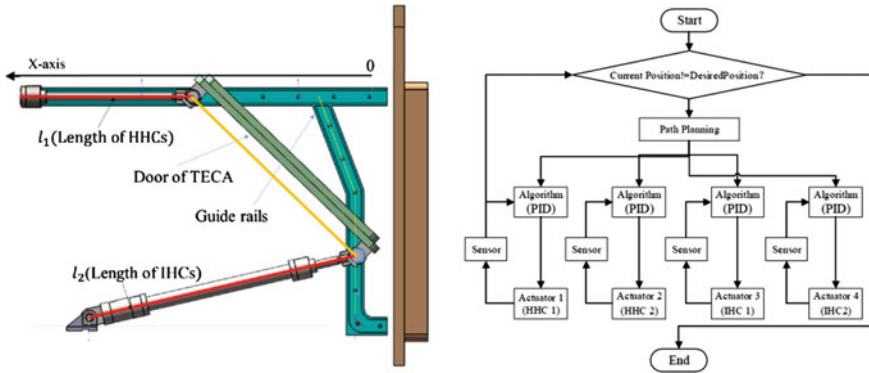


Fig. 7 The tilting mechanism and synchro control scheme

also, the Tilting mechanism is required of synchronized motion [17]. The Tilting mechanism and control chart is show in Fig. 7. The kinematic analysis is detailed listed in reference [15].

We perform several control experiments to validate the designed mechanism and the schemed control measures. The motion control results are shown in Fig. 8. It can be seen that, the four linear cylinders are able to move according to planning trajectory in a certain range of error. The maximum tracking error of the HHC (horizontal hydraulic cylinder) is about 3.60 mm, and the maximum tracking error of the IHC (inclined hydraulic cylinder) is 3.58 mm. And, the synchronization accuracy between the same set of hydraulic cylinders can to be accepted. Meanwhile, we can see that, since the dead zone of hydraulic valve, the deviation is a little larger during begin and end of the movement. It indicates that the PID parameters are not optimal, while the desired position of IHC has a bigger change. Maybe we should obtain optimal parameters though precisely modeling the controlling object, or use other control algorithms, such as intelligent measures as fuzzy or neural network.

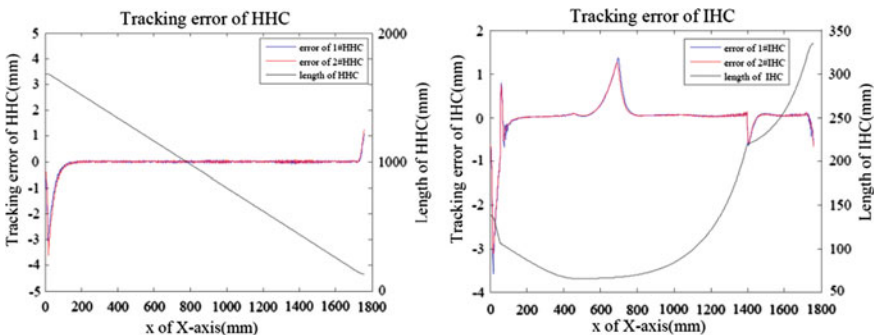


Fig. 8 The cylinder tracking error for the desire DSD motion

4 Conclusion

In the previous several years, Our group have developed the whole mechanism and control schemes for the TECA. And the whole mechanisms manufacturing and assembling are completed. The whole control systems are configured and programmed. The whole motion control experiments are performed again and again. Hence, the whole motion capability of the prototype are validated. Some mechanism parts are optimized and the control algorithms are improved. A lot of experiences on manufacturing and controlling are accumulated. The overall newer optimum mechanism and control measured will be improved and refreshed in the new generation of the TECA. The engineering and the practicability of the TECA will approach nearer.

Acknowledgements This research work was supported by the China National Magnetic-Confinement Fusion Energy R&D Program (ITER) and funded by the National Science and Technology Major Project of the Ministry of Science and Technology of China (2012GB102002, No.2012GB102007) and the National Natural Science Foundation of China (NSFC) under Grant No.61503361, The natural science foundation of Anhui Province 1308085MF86. The authors wish to thank the research group for their great contributions to the presented research. The views and opinions expressed in this paper are the sole responsibility of the authors.

References

1. <http://west-east.ipp.ac.cn/index.php>
2. Ribeiro, I., Damiani, C., Tesini, A., et al.: The remote handling systems for ITER. *Fusion Eng. Des.* **86**(6–8), 471–477 (2011)
3. Vale, A., Fonte, D., Valente, F., Ferreira, J., Ribeiro, I., Gonzalez, C.: Flexible path optimization for the cask and plug remote handling system in ITER. *Fusion Eng. Des.* **88**(9–10), 1900–1903 (2013)
4. Locke, D., Gutierrez, C.G., Damiani, C., Friconneau, J.-P., Martins, J.-P.: Progress in the conceptual design of the ITER cask and plug remote handling system. *Fusion Eng. Des.* (2014)
5. Tesini, A.: DDD2.3 Remote handling equipment. ITER Naka Joint Work Site (2004)
6. Zi-bo, Z., Da-mao, Y., Lei, C., Li, G.: The structural design and analysis of pallet in ITER transfer cask for remote handling operations. *Plasma Sci. Technol.* **11**(3), 352–358 (2009)
7. Wang, F.-F., Yao, D., Li, G.: Primary structure and seismic response analysis of the ITER transfer cask. *Nucl. Fusion Plasma Phys.* **29**(3), 270–274 (2009)
8. Zhou, Z.-B., Yao, D.-M., Li, G., Cao, L., Qin, S.-J.: Design on alignment system of ITER remote handling transfer cask. *Nucl. Fusion Plasma Phys.* **29**(4), 357–360 (2009)
9. Li, S.-Q., Cao, L., Li, G.: Design and simulation for the support legs of the ITER transfer cask system. *Nucl. Fusion Plasma Phys.* **31**(1), 57–61 (2011)
10. Cheng, T., Liu, C., Yao, D.: Double seal door design and analysis for ITER transfer cask. *Nucl. Fusion Plasma Phys.* **28**(4), 347–351 (2008)
11. Pan, S., Yao, D., Zibo, Z., Jing, W., Lei, C., Han, X et al.: The control system of remote transfer CASK for ITER. In: *Proceedings of the 14th Annual Conference on Nuclear Electronics and Detection Technology*, Wulumuqi, China, pp. 415–420 (2008)

12. Qin, S.-J., Song, Y., Li, G., Yao, D.: Kinematics analysis of double seal door for iter remote handling transfer cask system. In: Proceedings of the 18th International Conference on Nuclear Engineering: Volume 6, Xi'an, China, pp. 501–507, May 17–21 (2010)
13. Qin, S.-J., Song, Y., Yao, D., Li, G.: Theoretical calculation and kinematics analysis of double seal door for ITER remote handling transfer cask system. *J. Fusion Energ.* **33**, 435–443 (2014)
14. Wang, B., Wang, Y., Gao, L., et al.: Pneumatic design for Intelligent air transfer system based on AMESim. In: IEEE International Conference on Information and Automation (ICIA), pp. 758–762 (2014)
15. Guo, W., Gao, L., Wang, Z., et al.: Design and analysis of a double seal door for the tokamak equipment CAsk (TECA). *IEEE Int. Conf. Cyber Technol. Autom. Control Intell. Syst. (CYBER)* **64**(2), 35–42 (2015)
16. Liu, Y., Guo, W., Wang, D. et al.: Research on a type of parallel lifting mechanism for TECA transfer cask. robot. (in press)
17. Wang, D., Liu, Y., Ge, Y., Design of control system on the bidirectional sealing door for TECA. *IEEE Int. Conf. Inf. Autom. (ICIA)* (2015)

3D SLAM for Scenes with Repetitive Texture Inside Tokamak Chamber

Wei Liu, Zhong Zheng, Odbal, Bin Cai and Zengfu Wang

Abstract As the intra-scene of Tokamak chamber contains many repetitive textures, the traditional 3D reconstruction method based on the feature descriptor would be difficult to work well since the image matching algorithms based on feature descriptor are unstable and may fail sometimes in this environment. To address this problem, a novel multilevel matching algorithm is proposed, which uses the structural characteristics of Tokamak chamber as prior knowledge to find reliable correspondence points between two images. Firstly, each image is divided into basic structure regions. Then, to obtain the corresponding relation of structure regions from multiple images, we take the preliminary matching on the structural framework. The feature points is matched inner the structure regions to ensure the correctness of the feature matching. To testify the effectiveness of the proposed algorithm, it is applied to repetitive texture images captured in the Tokamak chamber, and the experimental results show that more correct matching points are acquired, smooth and clear 3D point-cloud data are generated, and high accurate and integrated reconstruct results are obtained.

Keywords 3D reconstruction · Repetitive texture · Structural characteristics · Structure from motion algorithm

1 Introduction

Because of the numerous equipment, the complexity of various piping and the narrow workspace of the Tokamak chamber, the maintainer should not operate directly on the relevant parts but complete the corresponding operations using the manipulator within the remote control strategy outside the tank even during the maintenance of the equipment since part of the cabin is polluted by radioactive and toxic substances. It is necessary to clear the accessible workspace of the manipu-

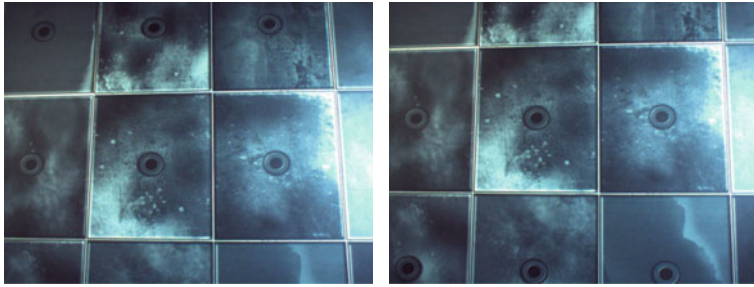
W. Liu · Z. Zheng · Odbal · B. Cai · Z. Wang (✉)
Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, China
e-mail: zfwang@ustc.edu.cn

lator in order to plan route on this basis. So, the 3D modeling of the scene in the chamber is extremely essential.

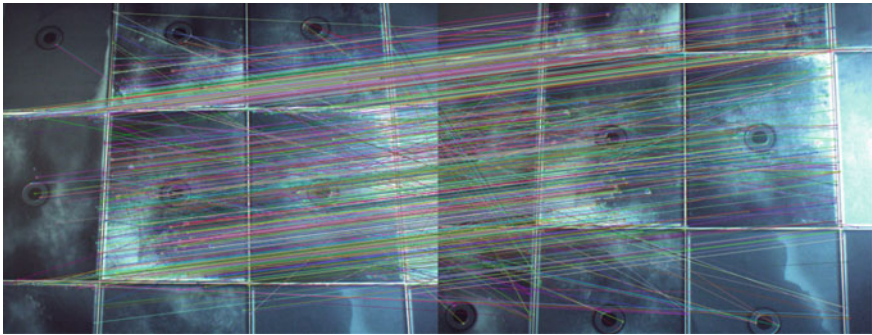
As one of the most important branch in the field of computer vision, multiple view 3D reconstruction technique analyzes the spatial location, focal length and distortion coefficient of the camera by taking the sequence images around the scene as the input and feature matching as the basis to reconstruct the 3D point cloud, and then obtains the realistic 3D model by using patch generation algorithm and texture mapping technology. The core content of this kind of method consists of two kinds of technology: recovering the scene Structure from motion (SFM) [1–4] and Multi-view stereo (MVS) [5, 6]. The SFM restores the motion parameters of the camera and the scene structure generally based on the low-level features (point features) of multi-view images [1, 2], basic geometry constraints (such as plane, rectangle, etc.) [3] or the combination of both [4]. And the method based on the low-level features always achieves better performance. However, the images from realistic scene usually put out monotonous and repetitive texture, for example, Tokamak chamber, as the application environment of this paper, is characterized by the highly repetitive texture. These methods relying on the image texture information in the process of feature matching cannot estimate the depth values since lacking of the match features on the surface with repetitive or weak texture especially the flat area of the images. It leads the reconstruction model to be a lot of holes. So, it is a critical step to find reliable matching points for SFM algorithm.

In terms of natural feature extraction and matching, SIFT operator based on the scale invariance, which is presented by David Lowe, is now recognized as the most outstanding algorithm about natural feature extraction and matching [7]. However, the main factors restricting the algorithm widely used are the poor real-time performance in feature extraction and matching. Though SIFT operator is improved by many researches, such as SURF [8], which is used to reduce the dimension of the SIFT operator descriptors, the SIFT and the SURF algorithm not only take up huge storage space but also are with pretty slow matching speed since the descriptors used are float vectors. In recent years, researchers have proposed to generate binary feature descriptor [9], the ORB features, by comparing the image grey value. Compared with the float descriptors, this kind of descriptors is characterized by less storage space occupation and fast matching speed, and thus become the hotspot in the field of computer vision. But, there appears severe mismatch phenomenon when processing the sense with repetitive texture using these algorithms directly, as shown in Fig. 1b.

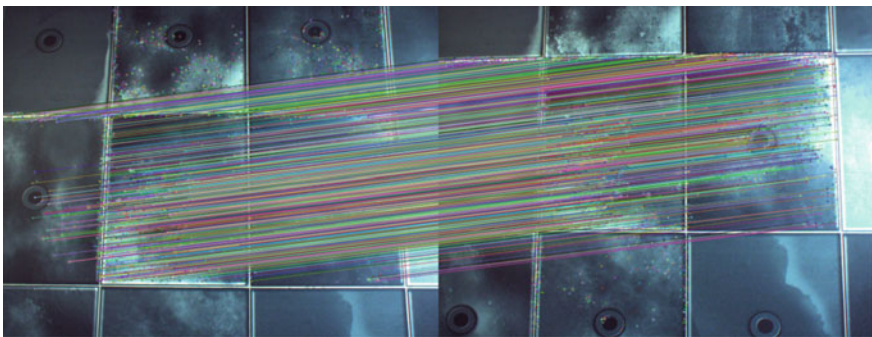
To solve the problem of mismatch about the interior scene image of Tokamak chamber, an image matching algorithm combining the structure matching with the feature matching is proposed by using the structural information of the scene. Compared with the traditional image matching algorithm based on the feature descriptors, the proposed method has the more powerful ability to acquire the robust and accurate matching results for the images with repetitive patterns. The high quality 3D point cloud data is acquired by applying the matching results to SFM algorithm. On this basis, 3D model reconstruction and texture synthesis is accomplished. Figure 1c shows the image matching results of the proposed



(a) two motion images



(b) fast library for approximate nearest neighbors matching method based on SIFT feature



(c) the proposed method

Fig. 1 The matching results of different matching methods

algorithm. It is observed that the proposed algorithm can get more correct matching point pairs.

The rest of this paper is organized as follows: the new image matching method is presented in Sect. 2, followed by the experiments in Sect. 3. Finally, we conclude this paper in Sect. 4.

2 Our Method

According to the structural characteristics of the experimental environment, an image matching method combining the structure matching with the feature matching is proposed. For a set of images captured during the motion, first of all, we have to carry out regional segmentation among the images, and extract the relevant information of each region. And then the various regions are corrected to the same size blocks according to the regional information. Thirdly, the corrected blocks are combined according to the special location. The region matching results is obtained by matching the adjacent images. Finally, the ORB feature is extracted from the region matching results to process the feature matching again, and the final image matching results is obtained. The workflow of our 3D reconstruction method is shown in Fig. 2.

2.1 Regional Information Retrieval and Correction

Figure 3 shows the single tile image. From the figure, it shows that tile image is a rectangular block consists of two concentric circles internally. In order to extract information from each region and rectify it, we need to obtain the complete information of tile image, including the four vertices of the rectangle and the center circle coordinates.

2.1.1 Center Circle Coordinate Extraction

Tile images come out strong directional information in the center circle area while the other areas of the tile images are relatively flat. The HOG feature [10] is acquired by doing statistics among the calculated image gradient direction of local area. Support vector machine (SVM) [11] is a new machine learning method based on the Vapnik-Chervonenkis Dimension theory and Structure Risk Minimization. It

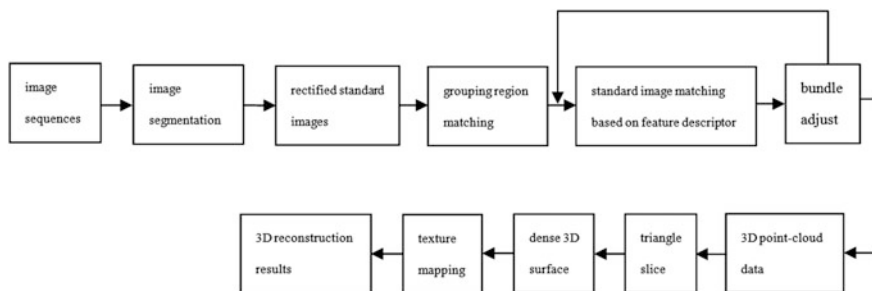


Fig. 2 The workflow of our 3D reconstruction method

Fig. 3 Tile image

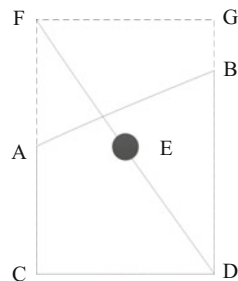


has the ability to solve the classification problems of two or more classes by mapping the input vector into a high dimensional feature space and constructing the optimal classification plane in this space. HOG feature combined with SVM classifier has been widely used in object detection task since HOG is not sensitive to light, direction, size, etc. and the SVM have good classification ability. Therefore, this paper utilizes HOG features combined with SVM classifier to detect the center circle and get the center circle coordinate information.

2.1.2 Get Vertex Coordinates of Tile

The characteristic of the actual tile patch that is surrounded by straight line is still in existence when the tile image obtained by the small distortion. Probabilistic Hough transform [12], which is the Hough transform in a certain range, so as to cut the amount of calculation and shorten the calculation time. Therefore, we use probabilistic Hough transform method to obtain the vertex information of segmented tile areas. We can be directly obtained the real vertex information from the complete tile area; since all points on a single tile are in a flat, we can estimate the missing vertex according to the vertex which has been obtained for some incomplete tiles. As shown in Fig. 4, ABCD are the four detected vertices of incomplete tile, E is the central point, and the missing vertex F can be obtained according to the intersection of the straight line AC and ED, and the vertex G can be calculated in the same way.

Fig. 4 Calculate the coordinate information of missing vertex

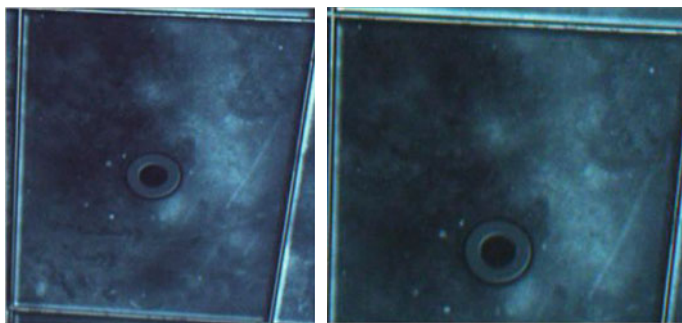


2.1.3 Image Region Rectification

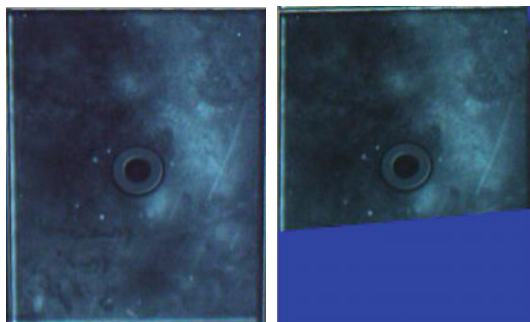
According to the Sects. 2.1.1 and 2.1.2, the complete information of each tile image which contains four vertex information and the corresponding center circle coordinate information is obtained. After the four real vertices coordinate information of each tile are obtained, it is mapped to the four vertices of the standard tile, through the perspective transformation, and then gets the rectified standard tile image. Figure 5 is the extraction of tile regions and the rectified standard images.

2.2 Grouping Region Matching

According to Fig. 1b, we can find that it is difficult to achieve satisfactory matching results for the intra-scene of Tokamak chamber, which use the traditional image matching algorithm based on feature descriptor. Analysis the images of intra-scene,



(a) the extraction of tile regions of two key frames
(complete and incomplete region)



(b) the corresponding of rectified standard images

Fig. 5 An example of tile rectification

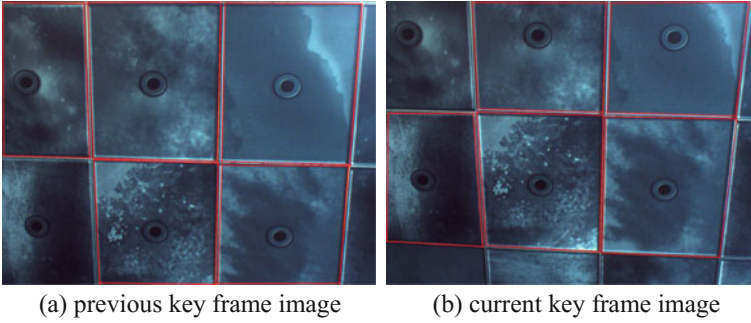


Fig. 6 An example of grouping region matching

we can be found that although its local area to show the weak texture feature, but global area has a strong structural features. For a standard rectification tile image A and B, the whole similarity value is:

$$\begin{aligned}
 HLDC = & \left(\sqrt{\sum GA_x \cdot GA_x} \cdot \sqrt{\sum GB_x \cdot GB_x} \right) / \sum GA_x \cdot GB_x \\
 & + \left(\sqrt{\sum GA_y \cdot GA_y} \cdot \sqrt{\sum GB_y \cdot GB_y} \right) / \sum GA_y \cdot GB_y
 \end{aligned} \tag{1}$$

where $GA_x(x, y)$, $GA_y(x, y)$, $GB_x(x, y)$ and $GB_y(x, y)$ are horizontal gradient and vertical gradient of the image A and B at (x, y) position, respectively.

Due to the standard tile image with high repeatability texture, it will produce some false matching cases when whole matching the standard tiles. In order to solve this problem, we grouping the division tile images according to spatial relations, and the similarity value are achieved by grouping matching method. Figure 6 is a grouping matching example, Fig. 6a, b can extract 5 rectified standard tiles (2 complete standard tiles and 3 incomplete standard tiles, the red box). Firstly, we individually calculate the similarity of tile image using Eq. (1). Then form a group according to spatial position, and get the grouping similarity value. Thereby we can obtain the structure matching results.

2.3 Rectification Image Matching Based on Feature Descriptor

3D reconstruction algorithm based on image sequence, which need accurate 3D point-cloud data. We can match features in the rectified regions, which will greatly reduce the search range. The ORB feature [9] is a binary feature description, which



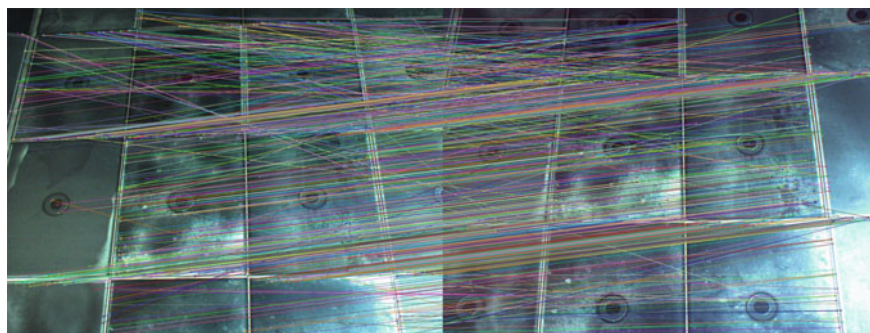
Fig. 7 An example of standard tile image matching

has the merits of occupy less storage space and fast matching. Thus, we use brute force matching method based on ORB feature for the rectified image region. Figure 7 is a matching sample for a pair of standard tile images.

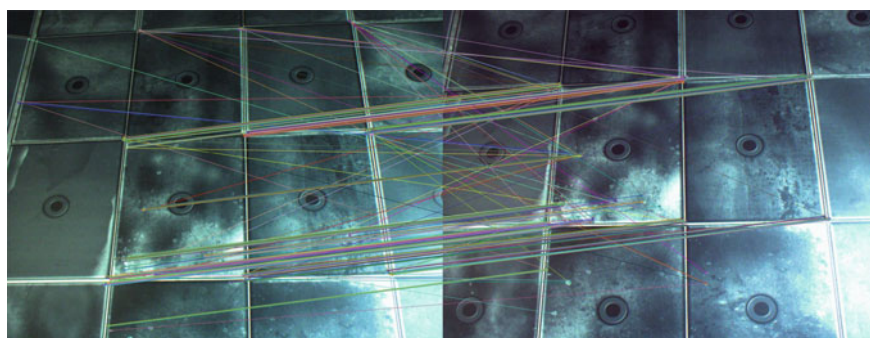
3 Experiments

In this section, we conduct experiments to show the validity of the proposed method. Firstly, we collect a series of motion images in the intra-scene of Tokamak chamber. Then, we test the proposed method on the repetitive texture images to show its superiority to the fast library for approximate nearest neighbors matching method based on SIFT feature descriptor [6] and brute force matching method based on ORB feature descriptor [8]. Figure 8 are part of the matching results. As shown in Fig. 8, the matching results directly based on SIFT, ORB features, there are many cases of mismatching, this is because there are many weak texture areas in the intra-scene, which show the image pixel value is little change. It caused there are very similar between each key point, thus it will produce a great ambiguity in the image matching based on local descriptor. And the proposed matching algorithm in this paper, it firstly matching the right area at the structure level, and then the feature matching method for the right matching area is used. The method makes full use of the structural characteristics of the experimental scenes, firstly, it obtain preliminary area matching results on a large scale, and then we use local feature matching in a certain range, all of this provide a certain assurance to the end of matching results. Thus the ideal matching results are obtained.

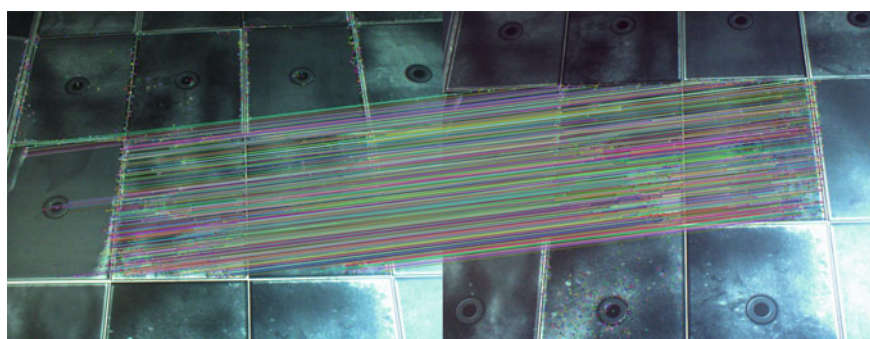
Figure 9 show the 3D point-cloud data which calculate according to the match point of Fig. 8, it can be find that the point-cloud data have clear boundary and



(a) fast library for approximate nearest neighbors matching method based on SIFT feature



(b) brute force matching method based on ORB feature



(c) the proposed matching method

Fig. 8 The matching results of different matching method

smooth contour. Figure 10 is the reconstruction results for the experimental scene, and it can completely display the scenario of the intra-scene. This can be provided good service for the subsequent teleoperation tasks.



Fig. 9 The point-cloud data of Fig. 8

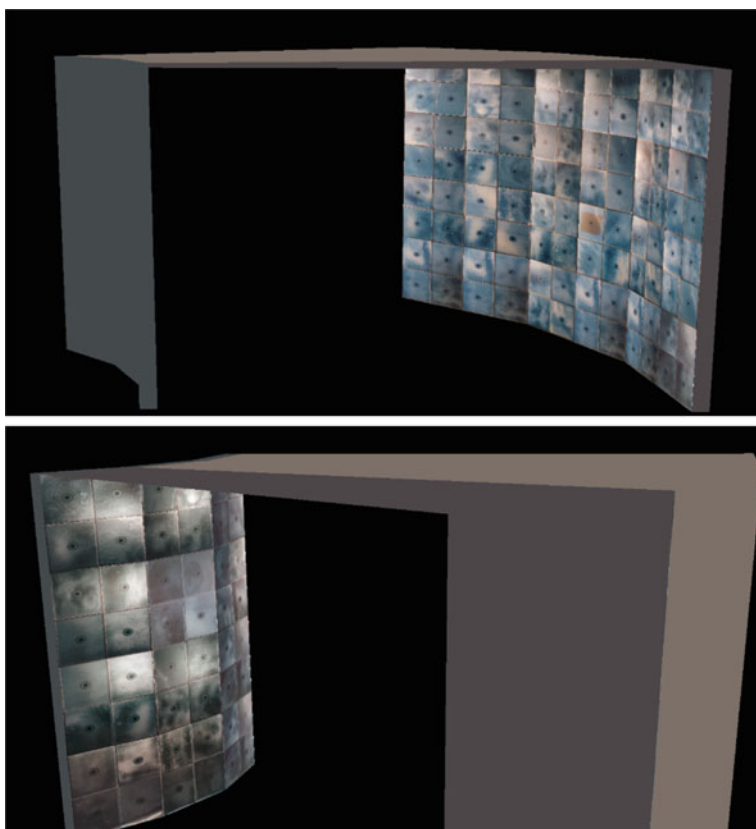


Fig. 10 The 3D reconstruction results

4 Conclusion

A novel 3D reconstruction algorithm based on image for intra-scene of Tokamak chamber is proposed. Compared with other algorithms, how to find right matching points is the biggest difference of the proposed algorithm. The traditional image matching algorithms based on feature descriptor are easy to give false matching results which bring considerable difficulties to subsequent 3D point-cloud computing. In order to solve this problem, according to the structural characteristics of the Tokamak chamber, a robustness and high precision image matching method based on structure information is designed. According to the matching results, a multistage SFM approach is used to reconstruct the structure. The high quality 3D point-cloud data are generated and high accuracy and complete modeling results are obtained.

Acknowledgements This research work was supported by the China National Magnetic-Confinement Fusion Energy R&D Program (ITER) and funded by the National Science and Technology Major Project of the Ministry of Science and Technology of China (No. 2012GB102007) and the National Natural Science Foundation of China (NSFC) under Grant No. 61503361. The authors wish to thank the research group for their great contributions to the presented research. The views and opinions expressed in this paper are the sole responsibility of the authors.

References

1. Agarwal, S., Furukawa, Y. et al.: Building Rome in a Day. In: Proceedings of the 12th International Conference on Computer Vision. Kyoto, Japan, pp. 72–79. IEEE (2009)
2. Snavely, N., Simon, I., et al.: Scene reconstruction and visualization from community photo collections. *Proc. IEEE* **98**(8), 1370–1390 (2010)
3. Bartoli, A., Sturm, P.: Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. *Int. J. Comput. Vis.* **52**(1), 45–64 (2003)
4. Zhou, Z., Jin, H. et al.: Robust plane-based structure from motion. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition. Rhode Island, USA, pp. 1482–1489. IEEE (2012)
5. Taniyai, T., Matsushita, Y. et al.: Graph cut based continuous stereo matching using locally shared labels. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1613–1620 (2014)
6. Zbontar, J., Lecun, Y.: Computing the stereo matching cost with a convolutional neural network. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1592–1599 (2015)
7. Lowe, D.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
8. Bay, H., Ess, A.T., et al.: SURF: speeded up robust features. *Comput. Vis. Image Underst.* **110**(3), 346–359 (2008)
9. Ethan, R., Vincent, R. et al.: ORB: an efficient alternative to SIFT or SURF. In: Proceedings of 2011 IEEE international conference on computer vision, vol. 58, no. 11, pp. 2564–2571 (2011)

10. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detect. In: Proceeding of IEEE International Conference on Computer Vision and Pattern Recognition, Vol. 1, pp. 886–893 (2005)
11. Sayanan, S., Mohan, M.T.: Active learning for on-road vehicle detection: a comparative study. *Mach. Vis. Appl.* **25**(3), 599–611 (2014)
12. Priyanka, M., Bidyut, B.C.: A survey of Hough transform. *Pattern Recogn.* **48**(3), 993–1010 (2015)

Design and Implementation of Wormlike Creeping Robot System Working at the Bottom of the Nuclear Fusion Vessel

Qiang Zhang, Ling Zhou, Yi Liu and Zengfu Wang

Abstract Maintenance for nuclear fusion vessel is crucial, yet it faces great difficulty due to the complex internal physical and geometric conditions. For this reason, such remote control mobile robot system is required to do the inboard detection and maintenance work instead of human beings. Under the above background, this paper presents a wormlike creeping robot system working on the V-shaped circular slot at the bottom of the EAST nuclear fusion vessel, and constructs the engineering prototype. The wormlike creeping robot adopts three-part structure, which consists of fore segment, mid segment and back segment connected by bidirectional universal joint. The fore and back segments stretch the paws to contact the surface of V-shaped slot, while the mid segment realizes the movement of robot by axial expansion. Functional analysis is devoted to modules of robot system. For the requirements of walking stability on the V-shaped slot, creeping gait planning is analyzed. Finally, the engineering prototype is running in the simulated vessel to test the synthesis movement properties and evaluate the mechanical properties. The test shows that the wormlike creeping robot system has good feasibility and effectiveness.

Keywords EAST nuclear fusion vessel · Wormlike creeping robot · Gait planning · Engineering prototype · Movement performance test

1 Introduction

The internal vessel of Experimental Advanced Superconducting Tokamak (EAST) is usually referred to as nuclear fusion vessel. In order to ensure its normal running, scheduled maintenance is taken to complete the following two tasks: one is visual

Q. Zhang · L. Zhou · Y. Liu · Z. Wang (✉)
Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, China
e-mail: zfwang@ustc.edu.cn

Q. Zhang · Z. Wang
Department of Automation, University of Science and Technology of China, Hefei, China

information acquisition and processing of the internal scene, another is replacement of damaged parts and fault correcting [1, 2]. However, for the extreme environment of the internal vessel, it's extremely challenging to complete the above tasks [3]. For example, in EAST nuclear fusion vessel, it has such physical characteristics as intense radiation, high temperature, intense magnetic field and high vacuum on the one hand; on the other hand, factors such as mass equipment, complex pipeline, narrow working space, devices polluted by radioactive and toxic substances, cause great difficulties for maintenance work in nuclear fusion vessel. So it's in urgent need to develop a remote control mobile robot system to reach the deep narrow space and complete given tasks, so as to promote the automation, flexibility and stability of remote operation and maintenance technology inside the nuclear fusion vessel [4].

The abroad existing research results for maintenance work in nuclear fusion vessel reflects in the manipulator robot. Manipulator robot is limited by its inherent supporting strength, load weight and size space to do the jobs [5, 6]. In view of the insufficiency of existing research results, combining with the above actual demand, this paper presents a wormlike creeping robot system working at the bottom of the EAST nuclear fusion vessel. This system can move along the V-shaped circular slot inside the EAST nuclear fusion vessel to make up some inherent defects of the manipulator robot, assist manipulator robot for inboard detection, component replacement and fault repairing. Based on the structure design and function analysis of modules, an engineering prototype of wormlike creeping robot system is constructed, completing relevant synthesis movement properties and feasibility verification. Furthermore, gait planning for wormlike creeping is researched to lay the foundation for the practical application of the wormlike creeping robot system.

2 Structure Design and Function Analysis

The mobile robot system has a high requirement to supporting ability and motion stationary supported by track carrier. It's a structured circular vessel with D-shaped section in the EAST nuclear fusion vessel, as shown in Fig. 1. The circular working space is horizontally symmetrical in the vessel, half bottom is covered with graphite or stainless steel tiles in difference shape [3]. The V-shaped deck is located at the middle of the half bottom, consists of gentle slope decks between wide gaps on both sides, suitable to be the supporting base of the robot platform. The space between V-shaped inward and outward deck forms the V-shaped slot.

General mobile robot according to the different walking style can be divided into legging, jumping, creeping, wheeled and tracked types. Facing EAST fusion vessel the irregular structured operation environment, creeping type is the most stable and adaptable of the gait types [7]. Meanwhile, for the sake of carrying all kinds of operation tools as much as possible, increasing generality of robot, wormlike creeping structure is adopted on the mobile robot system, as shown in Fig. 2.

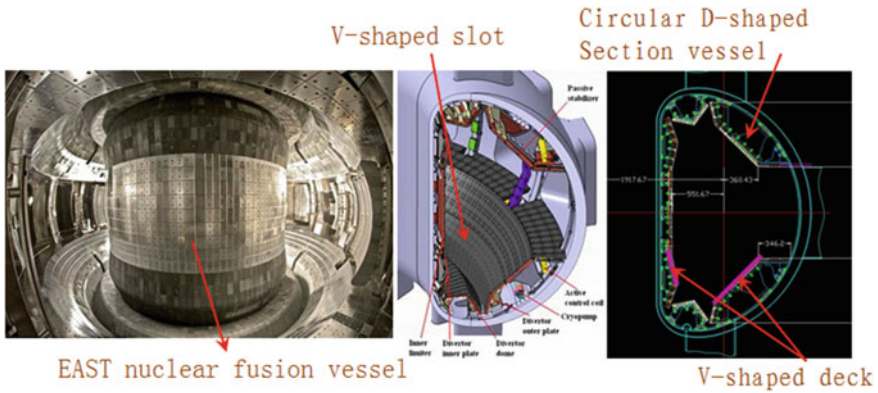


Fig. 1 Inner structure inside EAST nuclear fusion vessel

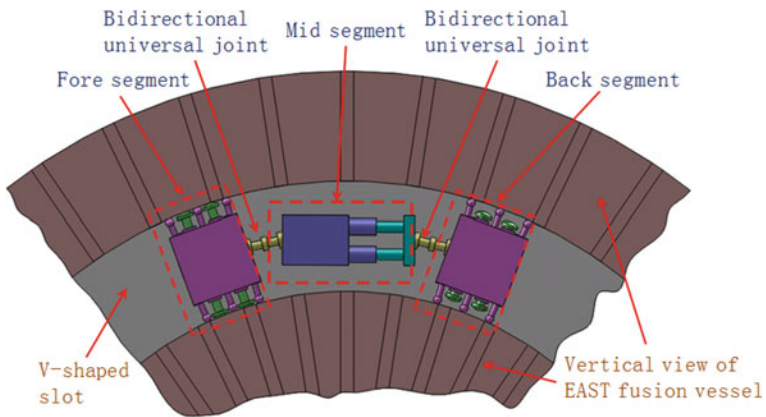


Fig. 2 Overall structure of wormlike creeping robot system

The mobile robot system adopts three-part structure, which consists of fore segment, mid segment and back segment connected by bidirectional universal joint. The fore and back segment has the same structure, symmetric at the two sides of mid segment, stretching the paws to contact the V-shaped deck to control the lock/unlock state with the V-shaped deck; the mid segment is connected with fore and mid segment by bidirectional universal joint to control axial expansion periodically. The coordinated motion between fore, mid, and back segment can drive the robot move along the V-shaped slot in wormlike style [8].

The fore/back segment is lateral positioning module of the robot system, constituted by supporting submodule, driving submodule, inward stretching submodule, outward stretching submodule and thermal control device. As shown in Fig. 3, driving submodule is installed in the supporting submodule, inward/outward stretching submodule are set symmetrically at both sides of the supporting

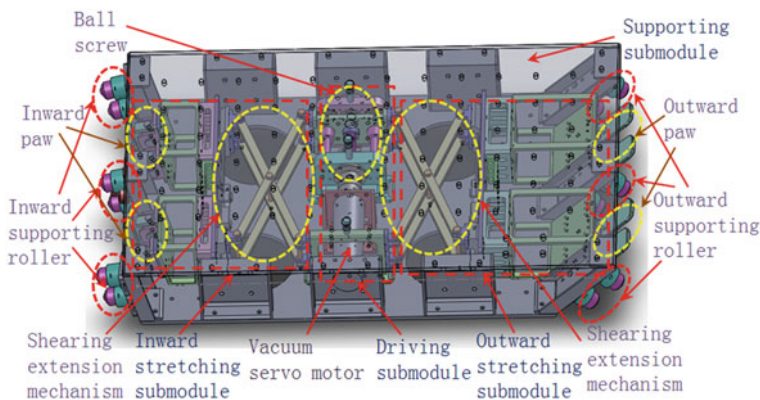


Fig. 3 Fore/back segment

submodule in the same structure, one end connected with driving submodule, another end stretching over the supporting submodule, which constitute the shearing extension mechanism; Driving submodule consists of ball screw mechanism driven by vacuum servo motor, driving inward/outward stretching submodule on both sides stretching out and back synchronously to control lock/unlock state from the V-shaped slot, as shown in Fig. 4.

The supporting submodule is for whole segment supporting, with ladder-shaped section. The normal direction of inward and outward end face, on which installed n ($n \geq 2$) sets of linear uniformed rollers as the supporting end of fore/back segment, are paralleled perpendicularly to normal direction of V-shaped deck. On the one

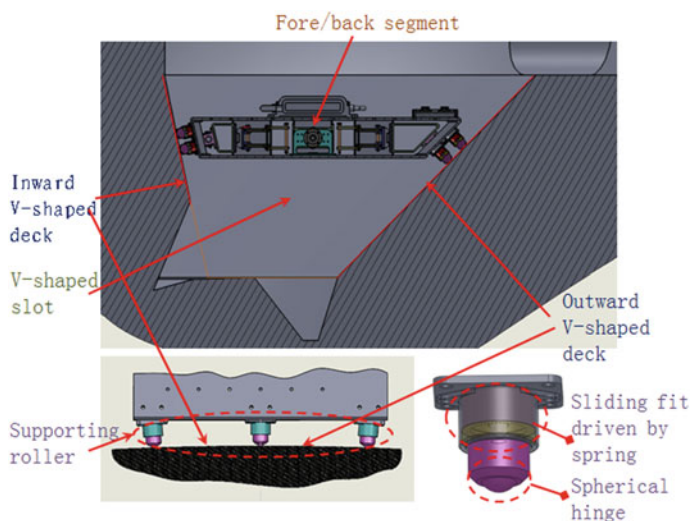


Fig. 4 Fore/back segment located on the v-shaped slot

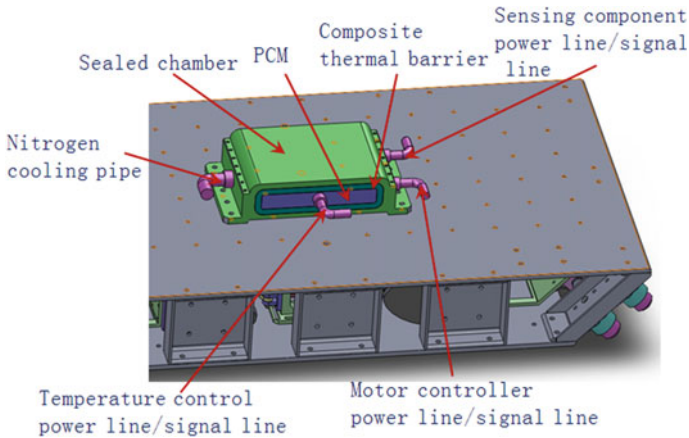


Fig. 5 Thermal control device

hand, it is beneficial to increase the contact area and ensure uniform stress, On the other hand it can improve bearing performance of the fore/back segment; the roller has spherical hinge and sliding fit driven by spring, which improve its self-adaptive ability while moving over the damaged deck; furthermore, the roller can minimize the friction resistance between robot and deck, optimize system driving performance, achieve the goal of saving energy consumption.

As shown in Fig. 5, thermal control device is installed on the supporting submodule, to ensure normal function of motors, sensors, electronic devices and the cables in the extreme environment of high temperature, high vacuum, and intense radiation inside the nuclear fusion vessel. It's constituted by sealed chamber, composite thermal barrier, and phase change materials (PCM). From the inner space of PCM come out respectively the nitrogen cooling pipe, power lines and signal lines of temperature control module, motor controller and sensing components.

The mid segment is the axial moving module of the robot system, constituted with supporting submodule, fore submodule and back submodule; as shown in Fig. 6, fore submodule is installed inside supporting submodule; fore submodule and back submodule move relatively along the axis direction as an elastic whole; the supporting submodule is the supporting frame assembled of rectangular plates; vacuum servo motor installed in the supporting submodule drives the gear pairs transmitting to ball screw, then transmitting to straight push rod, realizing the axial relative feed motion between fore submodule and back submodule; inside the supporting submodule, two optoelectronic switches are installed to sense axial position variation of the straight push rod, respectively giving position signal of two extreme position to the robot control system to precisely control periodic axial feeding distance of mid segment; the gear pairs are for power transmission and for shortening the axial length of mid segment to increase the moving capacity of robot on the circular V-shaped slot.

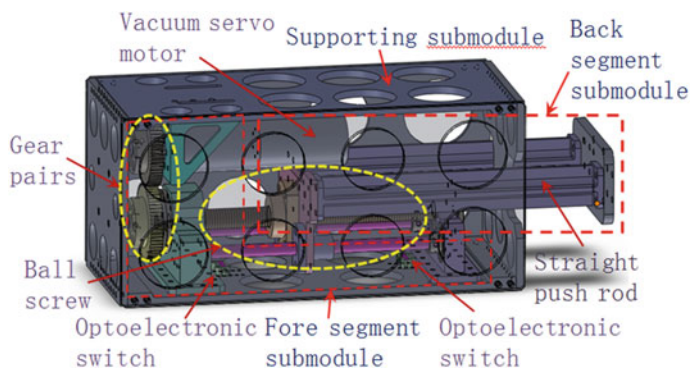


Fig. 6 Mid segment

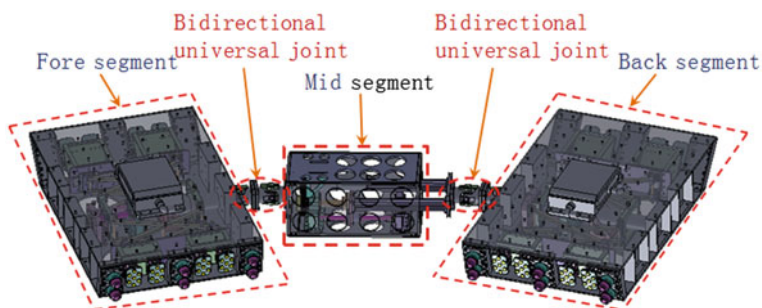


Fig. 7 Structure design of the wormlike creeping robot system

Mid segment and fore/back segments are connected with bidirectional universal joints to form the wormlike creeping robot system, as shown in Fig. 7. In order to ensure the robot passing the entrance along the horizontal direction into the EAST nuclear fusion vessel, the designed height of fore/back and mid segment shall be less than the width of the rectangular window. Carrying sorts of operation tools, it has strong practicability to complete the maintenance tasks inside the vessel. Running condition of the robot system on the V-shaped slot inside EAST nuclear fusion vessel is shown in Fig. 8.

3 Gait Planning and Control Method

To ensure robot moving steadily on the V-shaped slot, it requires reasonable gait planning and time series analysis, and on the basis building suitable movement control method [9]. The wormlike creeping mechanism is based on mutual

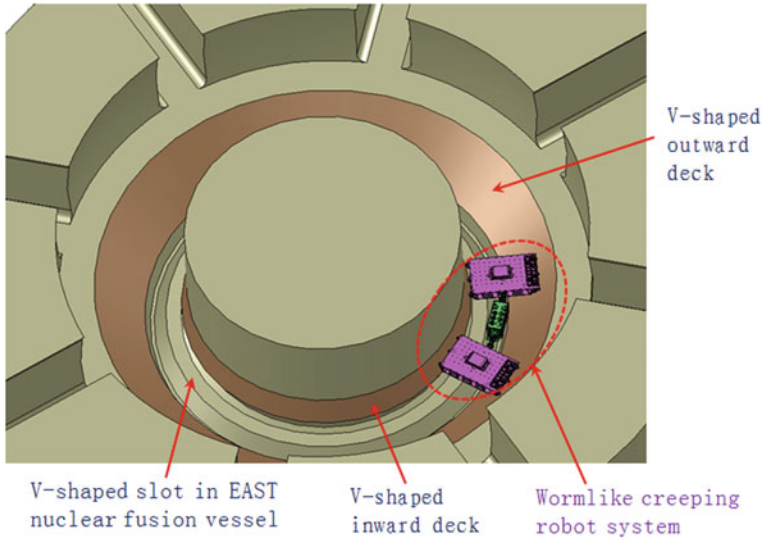


Fig. 8 Running condition on the V-shaped slot

coordination between fore/back and mid segment, belonging to periodical gait. In order to illustrate easily, the fore segment, mid segment, back segment are marked in number 1, 2, 3 respectively.

In the segment 1 and 3, vacuum servo motor drives the ball screw transmission do reciprocating motion, setting motor output shaft revolution corresponding to one stroke as n_1 ; in the segment 2, the motor output shaft revolution corresponding to one stroke is set as n_2 ; according to different motion state of segment 1, 2, 3, time cost of one creeping step, which is clockwise in overhead view, is set as one period $T(0 < t_1 < t_2 < t_3 < t_4 < t_5 < T)$. In one period T , control steps are as following:

At State 0(start state): segment 1 is locked on the slot deck, segment 2 is minimum shortened, and segment 3 is locked;

Process I. during $0 \sim t_1$, motors in segment 1 and 2 stop, paws in segment 1 stretch longest, submodules in segment 2 keep relative rest; motor in segment 3 rotates forward $n_1/2$, drives ball screw transmitting to shearing extension mechanism, then drives paws retracting back. At State t_1 , segment 1 is locked, segment 2 is minimum shortened, and segment 3 is unlocked;

Process II. during $t_1 \sim t_2$, motors in segment 1 and 3 stop, paws in segment 1 stretch longest, paws in segment 3 retract shortest; Motor in segment 2 rotates forward $n_2/2$, drives ball screw pushing back submodule moving one step towards the right, then transmits to bidirectional universal joint pushing segment 3 moving one step clockwise. At State t_2 , segment 1 is locked, segment 2 is maximum extended, and segment 3 is unlocked;

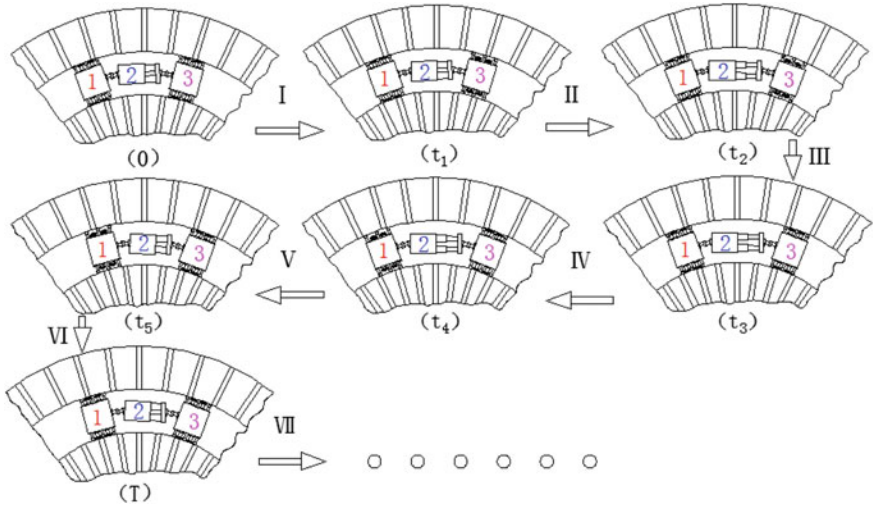


Fig. 9 Gait planning and analysis of wormlike creeping movement on V-shaped slot

Process III. during $t_2 \sim t_3$, motors in segment 1 and 2 stop, paws in segment 1 stretch longest, submodules in segment 2 keep relative rest; motor in segment 3 rotates backward $n_2/2$, drives paws stretching out. At State t_3 , segment 1 is locked, segment 2 is maximum extended, and segment 3 is locked;

Process IV. during $t_3 \sim t_4$, motors in segment 3 and 2 stop, paws in segment 3 stretch longest, submodules in segment 2 keep relative rest; motor in segment 1 rotates forward $n_1/2$, drives paws retracting back. At State t_4 , segment 1 is unlocked, segment 2 is maximum extended, and segment 3 is locked;

Process V. during $t_4 \sim t_5$, motors in segment 1 and 3 stop, paws in segment 1 retract shortest, paws in segment 3 stretch longest; motor in segment 2 rotates backward $n_2/2$, drives ball screw pulling back submodule moving one step towards the right, then transmits to bidirectional universal joint pushing segment 1 moving one step clockwise. At State t_5 , segment 1 is unlocked, segment 2 is minimum shortened, and segment 3 is locked;

Process VI. during $t_5 \sim T$, motors in segment 3 and 2 stop, paws in segment 3 stretch longest, submodules in segment 2 keep relative rest; motor in segment 1 rotates backward $n_1/2$, drives paws stretching out. At State T, segment 1 is locked, segment 2 is minimum shortened, and segment 3 is locked.

Thus, the robot completes an entire period of one clockwise step on the slot, as shown in Fig. 9. The anticlockwise step is similar to the clockwise step, merely the rotation direction of motor and control sequence are different.

Figure 10 shows motor control state of the creeping period, corresponding to the gait in Fig. 9. The horizontal axis denotes the time phase in period $0 \sim T$, and the vertical axis denotes the motor control state of segment 1,2,3 respectively,

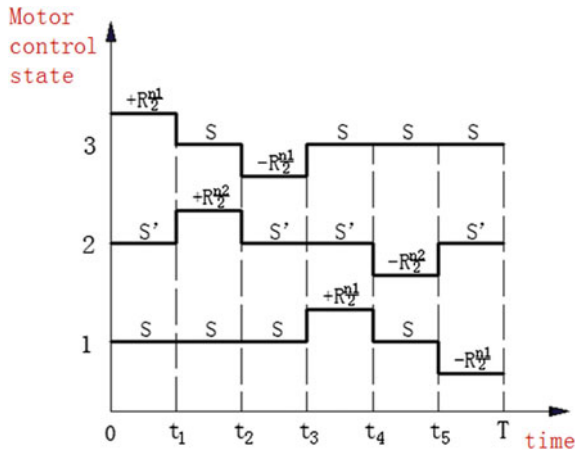


Fig. 10 Motor control state of the creeping period

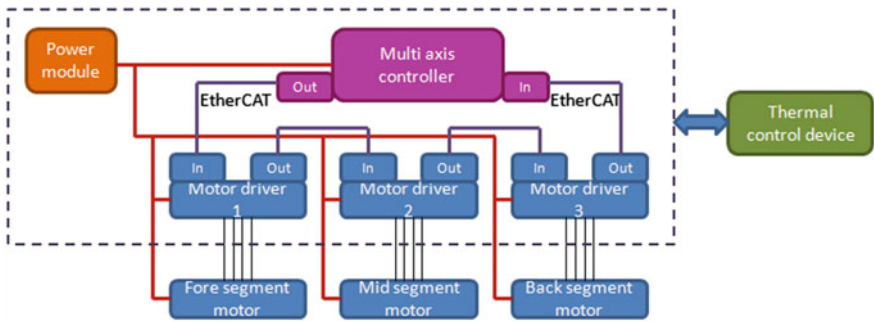


Fig. 11 Motor control system chart

“ $+R_{n_i}^1$ ”、“ S ”、“ $-R_{n_i}^1$ ” denotes that the motors in segment 1 and 3 rotate forward $n_1/2$, stop, and rotate backward $n_1/2$; “ $+R_{n_i}^2$ ”、“ S ”、“ $-R_{n_i}^2$ ” denotes that the motor in segment 2 rotates forward $n_2/2$, stops, and rotates backward $n_2/2$.

Motor control system chart is shown in Fig. 11. The gait of the creeping units is coordinately controlled by multi-axis controller, and the segments are controlled by motor drivers; power module is for power supply of electromechanical modules; thermal control device is to protect the core workpiece of modules from high temperature, high vacuum and intense radiation; real time ethernet protocols is adopted to communicate between multi-axis controller and motor drivers.

4 Engineering Prototype and the Movement Performance Test

Based on structure design, function analysis and gait planning, components and parts of the robot system are purchased, manufactured and assembled. Movement control hardware system is built. Engineering prototype and the coordinate imitative EAST vessel are constructed [3, 10], as shown in Fig. 12.

As shown in Fig. 13, the engineering prototype is set on the V-shaped slot of imitative EAST vessel to effectively assess mechanics performance of wormlike

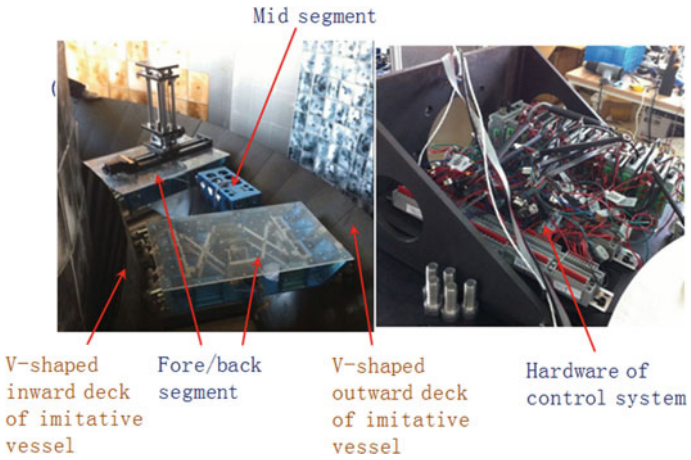


Fig. 12 Engineering prototype

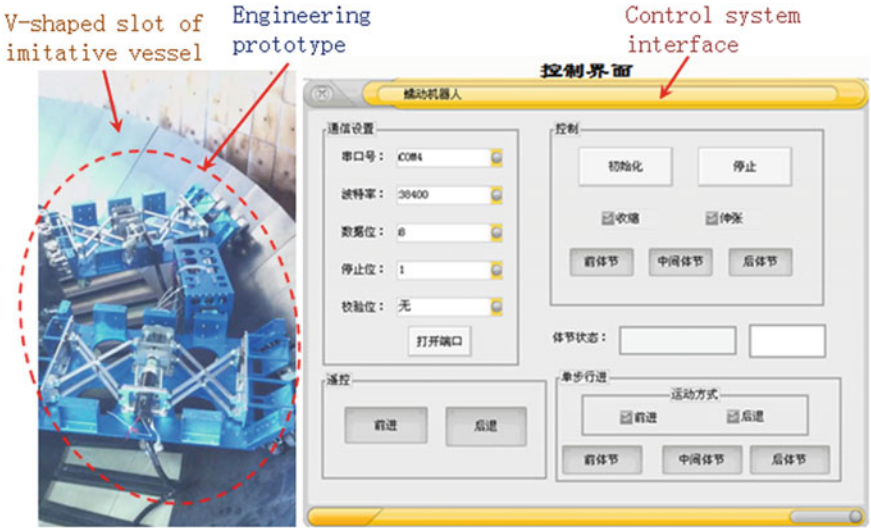


Fig. 13 Movement performance test of engineering prototype

Table 1 Test result of creeping reliability

Step order	Start time (s)	End time (s)	Actual moving time (s)	Theoretical moving time (s)	Absolute error (s)	Relative error (s) (%)
1	1.00	19.02	18.02	18	0.02	0.11
2	19.02	36.97	17.95	18	-0.05	0.28
3	36.97	54.98	18.01	18	0.01	0.06
4	54.98	73.00	18.02	18	0.02	0.11
5	73.00	91.02	18.02	18	0.02	0.11
6	91.02	108.99	17.97	18	-0.03	0.17
7	108.99	127.03	18.04	18	0.04	0.22
8	127.03	145.01	17.98	18	-0.02	0.11
9	145.01	162.97	17.96	18	-0.04	0.22
10	162.97	181.01	18.04	18	0.04	0.22

Table 1 shows the reliability test results of the creeping prototype. During the 3 min period, 10 creeping paces have been moving through, while recording the start time and end time of every pace in order, calculating the actual moving time, comparing it with the theoretical moving time, calculating the absolute error, then getting the relative error. From the Table 1, it shows that the maximum relative error is 0.28 %, does not exceed the design target 0.5 %, and proves that the wormlike creeping robot system has the very good feasibility and effectiveness.

5 Conclusion

This paper researched a remote control mobile robot system to assist manipulator robot for inboard detection, component replacement and fault repairing instead of human, mainly introduced structure design, modular function analysis, gait planning and control of the robot system, based on which a set of engineering prototype is constructed, realized the wormlike creeping function of robot on the V-shaped slot of EAST fusion vessel, constructed imitative vessel, carried on the synthetic movement performance test of engineering prototype, obtained the expected effect.

The design of wormlike creeping robot system adopts three-part structure, carrying sorts of operating tools, is applied to complete maintenance task inside vessel, has good engineering practicality; the robot has wide mobile range throughout whole circle of the V-shaped slot, meets the requirement of deeply and widely operation inside the vessel for nuclear environment maintenance robot. Owing the wormlike creeping gait, the control method is simple and convenient; and for the good self-adaptive capacity of roller supporting device contacting with V-shaped slot, the robot system has running stability and environmental adaptability.

Although having achieved detecting and maintenance functions in EAST nuclear fusion vessel from the basic principle, the researched wormlike robot engineering

prototype still has some shortcomings. For example, environmental perception module has not been designed so that the movement control system is in open loop control which could affect the repeated positioning accuracy of the system. In addition, restricted by current research and development costs, test of synthesis movement performance is only conducted in the geometric imitative environment instead of real physical environment. In the future research work, the present robot system can be developed by optimizing the design scheme and improving the technical level to promote the sustainable development of nuclear fusion experimental device automatic maintenance technology.

Acknowledgements This research work was supported by the China National Magnetic-Confinement Fusion Energy R&D Program (ITER) and funded by the National Natural Science Foundation of China (NSFC) under Grant No. 61503361 and the National Science and Technology Major Project of the Ministry of Science and Technology of China (No. 2012GB102007). The authors wish to thank the research group for their great contributions to the presented research. The views and opinions expressed in this paper are the sole responsibility of the authors.

References

1. Pampin, R., Davis, A., Izquierdo, J., et al.: Developments and needs in nuclear analysis of fusion technology. *Fusion Eng. Des.* **88**, 454–460 (2013)
2. Damiani, C., Annino, C., Balagué, S., et al.: The European contribution to the ITER Remote Maintenance. *Fusion Eng. Des.* **89**, 2251–2256 (2014)
3. Song, Yuntao, Yao, Damao, Songata, Wu, et al.: Structural analysis and manufacture for the vacuum vessel of experimental advanced superconducting tokamak (EAST) device. *Fusion Eng. Des.* **81**, 1117–1122 (2006)
4. Dubus, Gregory, Puiiu, Adrian, Bates, Philip, et al.: Progress in the design and R&D of the ITER In-Vessel Viewing and Metrology System (IVVS). *Fusion Eng. Des.* **89**, 2398–2403 (2014)
5. Peng, Xuebing, Yuan, Jianjun, Zhang, Weijun, et al.: Kinematic and dynamic analysis of a serial-link robot for inspection process in EAST vacuum vessel. *Fusion Eng. Des.* **87**, 905–909 (2012)
6. Peng, X.B., Song, Y.T., Li, C.C., et al.: Conceptual design of EAST flexible in-vessel inspection system. *Fusion Eng. Des.* **85**, 1362–1365 (2010)
7. Loc, V.-G., Roh, S-g, Koo, I.M., et al.: Sensing and gait planning of quadruped walking and climbing robot for traversing in complex environment. *Robot. Auton. Syst.* **58**, 666–675 (2010)
8. Nakamura, Taro, Kato, Takashi, Iwanaga, Tomohide, et al.: Peristaltic Crawling Robot Based on the Locomotion Mechanism of Earthworms. *IFAC Proc. Vol.* **39**, 139–144 (2006)
9. Liu, J., Ma, B.-Y., Fry, N., et al.: Exploration Robots for Harsh Environments and Safety. *IFAC-PapersOnLine* **48–10**, 041–045 (2015)
10. Long, P., Liu, S., Wu, Y.: The FDS team, design and testing of the Fusion Virtual Assembly System FVAS 1.0. *Fusion Eng. Des.* **82**, 2062–2066 (2007)
11. Li, Maoxun, Guo, Shuxiang, Hirata, Hideyuki, et al.: Design and performance evaluation of an amphibious spherical robot. *Robot. Auton. Syst.* **64**, 21–34 (2015)

Part XVIII
Sensor Network

Indoor Positioning System Based on Distributed Camera Sensor Networks for Mobile Robot

Yonghoon Ji, Atsushi Yamashita and Hajime Asama

Abstract An importance of accurate position estimation in the field of mobile robot navigation cannot be overemphasized. In case of an outdoor environment, a global positioning system (GPS) is widely used to measure the position of moving objects. However, the satellite based GPS does not work indoors. In this paper, we propose a novel indoor positioning system (IPS) that uses calibrated camera sensors and 3D map information. The IPS information is obtained by generating a bird's-eye image from multiple camera images; thus, our proposed IPS can provide accurate position information when the moving object is detected from multiple camera views. We evaluate the proposed IPS in a real environment in a wireless camera sensor network. The results demonstrate that the proposed IPS based on the camera sensor network can provide accurate position information of moving objects.

Keywords Global positioning system · Indoor positioning system · Camera network · Mobile robot

1 Introduction

This paper proposes an indoor positioning system (IPS) that uses calibrated camera sensor networks for mobile robot navigation. Most navigation functions that allow mobile robots to operate in indoor environments were performed based on only map information built by simultaneous localization and mapping (SLAM) schemes. Recently, however, robots are expected to be operated in human-robot coexistence environments. The map information is static; hence it is hard to deal with human-

Y. Ji (✉) · A. Yamashita · H. Asama
Department of Precision Engineering, The University of Tokyo, Bunkyo, Japan
e-mail: ji@robot.t.u-tokyo.ac.jp

A. Yamashita
e-mail: yamashita@robot.t.u-tokyo.ac.jp

H. Asama
e-mail: asama@robot.t.u-tokyo.ac.jp

robot coexistence environments since it cannot reflect dynamic changes in the environment.

On the other hand, a distributed camera network system can monitor what is occurring in the environment and many automatic calibration schemes for such systems have been proposed in [8, 9]. Such system is able to manage environmental changes (e.g., moving objects) by processing image data from the distributed camera networks in real time; therefore, many of the problems encountered by classical mobile robot navigation can be improved by integrating both the static map information and the dynamic information from the calibrated camera sensor networks. The satellite based global positioning system (GPS) can greatly improve solutions to the positioning problem in an outdoor environment; however, the GPS does not work indoors. In this respect, many IPS technologies that use different signals (e.g., Wi-Fi, electromagnetic, radio, and so on) are developed [3, 4]. However, many limitations such as the strength of the signal, accuracy, multi reflection phenomenon remain unanswered.

Therefore, in order to realize a reliable IPS, we propose a novel approach that uses both distributed camera sensor networks and 3D map information of the environment. The 3D map information contains the information for the entire environment based on the world coordinate frame, and thus it can provide information on any point. Such map information can be easily obtained from the blueprint of the artificial environment (e.g., CAD data) or a traditional SLAM schemes. In addition, generating a bird's-eye image from camera networks is one of the effective methods to provide IPS in the case of a typical indoor environment. A bird's-eye image is an elevated view of the environment. It is able to provide sufficient information when mobile robots navigate in a typical indoor environment because the ground is flat, and thus the height can be ignored. In our research, therefore, the position signal is calculated by using overlapping zones, which are detected from the generated bird's-eye image.

The remainder of this paper is organized as follows. Section 2 describes the generation method of the global bird's-eye image to generate IPS information based on the calibrated camera network system in detail. Then, positioning method for the IPS information with the generated global bird's-eye image is presented in Sect. 3. The proposed IPS is validated with experiments for path planning of the mobile robot in Sect. 4. Finally, Sect. 5 gives the conclusions.

2 Bird's-Eye Image Generation

In this paper, to facilitate the monitoring of the entire environment, each of the calibrated camera images is transformed to local bird's-eye images that are taken from a user-define virtual camera, and then integrated into a global bird's-eye image. Figure 1 shows the overall process to generate the global bird's-eye image. Here, by way of example, three image data from camera networks are considered. First, floor detection process is performed to remove areas other than the floor using real camera

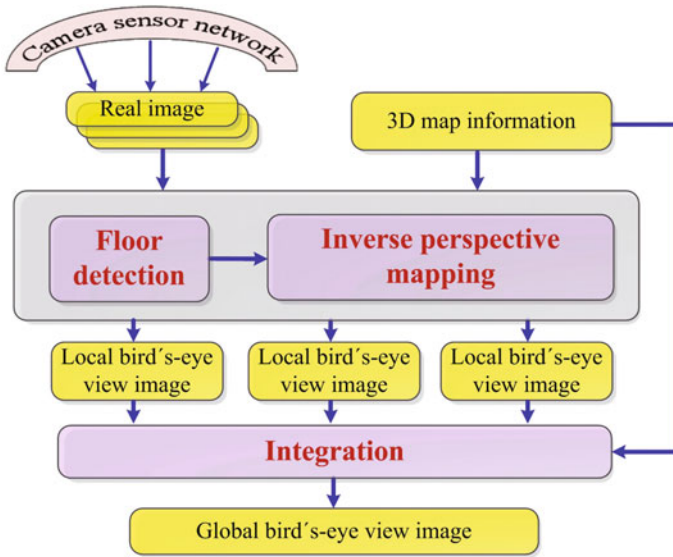


Fig. 1 Overall process to generate global bird's-eye image

images and predicted depth images generated from the camera poses (i.e., calibrated external camera parameters) and the 3D map information. Next, each of the generated floor images are converted into corresponding local bird's-eye images through an inverse perspective mapping and integrated into a global bird's-eye image. One of the notable features of this processing is that, with a distributed camera system, with accurately calibrated parameters based on the 3D map information, it is possible to generate RGB-D images (i.e., a RGB color image with a depth image) which provides 3D information, whereas only 2D image data cannot by itself provide 3D information. In this approach, the depth image generated from the 3D map information cannot mirror the dynamic information in real time because it is based on a static model; however, the RGB image is updated in real time. Therefore, even if this system cannot provide real time 3D information, it is able to handle 3D-like processing by utilizing overlapping areas from multiple camera images on an assumption that the ground is flat. Finally, the system performs positioning of IPS signal by using the generated global bird's-eye images.

2.1 Floor Detection

In order to detect the floor area in each of the real camera images captured by the camera sensor networks, first, height images $I_{R(w)}^H$ are generated from both, the camera poses $w = [x_c, y_c, z_c, \psi_c, \theta_c, \phi_c]^T$ and the 3D map information. Then, the real camera

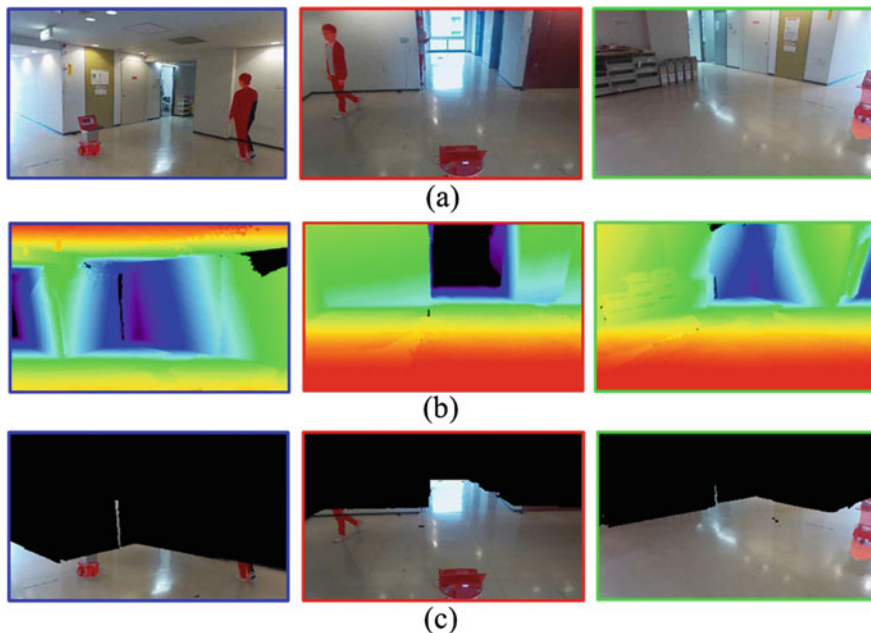


Fig. 2 Floor detection process: **a** real camera images $I_{R(w)}$ captured from camera sensor networks, **b** predicted depth images $I_{R(w)}^D$ that correspond to real camera images $I_{R(w)}$ generated by 3D map information, **c** height images $I_{R(w)}^H$ that correspond to real camera images $I_{R(w)}$, and **d** floor images $I_{R(w)}^F$ where regions other than the floor have been removed

images $I_{R(w)}$ (Fig. 2a) are converted into floor images $I_{R(w)}^F$ (Fig. 2d) using the height images $I_{R(w)}^H$ (Fig. 2c) as bounding values:

$$I_{R(w^{(k)})}^F(u, v) = \begin{cases} I_{R(w^{(k)})}(u, v) & I_{R(w^{(k)})}^H(u, v) < 0 + \epsilon \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

where k represents camera index. ϵ denotes the error constant which is added to take the camera calibration error into consideration. Thus, Eq. (1) removes regions other than the floor or those which heights are close to zero (the height of the floor surface is assumed to be zero in this study) from the real camera images $I_{R(w)}$. It is reasonable to remove areas other than the floor given that the moving objects generally walk on the floor in a typical indoor environment.

The detailed process for the floor detection for the real camera image is as follows. First, each depth image $I_{R(w)}^D$ is generated from the camera poses and the 3D map information as shown in Fig. 2b. Here, an important point to emphasize is that, with a distributed camera system with accurately calibrated parameters and 3D map information, it is possible to generate RGB-D-like images even if a typical optical camera cannot directly obtain the depth information as mentioned before.

Next, each of height images $\mathbf{I}_{R(w)}^H$ (Fig. 2c) that contain corresponding height information for each pixel are generated as follow:

$$\mathbf{I}_{R(w)}^H(u, v) = z_c - {}^C x \sin \theta_c + {}^C y \cos \theta_c \sin \psi_c + {}^C z \cos \theta_c \sin \psi_c, \quad (2)$$

$${}^C x = \mathbf{I}_{R(w)}^D(u, v), \quad (3)$$

$${}^C y = \frac{(u - c_u) \mathbf{I}_{R(w)}^D(u, v)}{f_u}, \quad (4)$$

$${}^C z = \frac{(c_v - v) \mathbf{I}_{R(w)}^D(u, v)}{f_v}. \quad (5)$$

Equations (3)–(5) convert 2D pixel information (u, v) to 3D data $({}^C x, {}^C y, {}^C z)$ based on the local camera coordinate frame using the pixel information of the depth image $\mathbf{I}_{R(w)}^D$. Here, the superscript C indicates the local camera coordinate frame. $\mathbf{w} = [x_c \ y_c \ z_c \ \psi_c \ \theta_c \ \phi_c]^\top$ represents the camera pose with respect to the world coordinate frame $\{\mathbf{W}\}$. f_u and f_v denote focal lengths. Equation (2) refers to the coordinate transformation from camera coordinate frame to world coordinate frame $\{\mathbf{W}\}$ for the height value. Finally, the floor images $\mathbf{I}_{R(w)}^F$ (Fig. 2d) are generated by applying Eq. (1). The coordinate system adopted in this study is described in detail in Fig. 3. Note that the optical axis is defined as x -axis in the camera coordinate frame in this paper.

2.2 Inverse Perspective Mapping

In order to generate local bird's-eye images from each camera, an inverse perspective mapping is exploited. The inverse perspective mapping removes perspective effects under the assumption that the ground is flat; therefore making environment monitoring more efficient [1]. In this study, the inverse perspective mapping is applied to each of the generated floor images $\mathbf{I}_{R(w)}^F$. This is done as it is assumed that the mobile robot navigates on the floor area in a typical indoor environment.

The bird's-eye image is generated by image processing based on the principle of geometrical perspective transformation for the captured real image. The principle used to convert real camera images to bird's-eye images for a virtual viewpoint is as follows. Figure 3 illustrates perspective transformations between each real camera and a user-define virtual camera that represents the relationship between a world coordinate frame $\{\mathbf{W}\}$, real camera images coordinate frames $\{\mathbf{R}\}$, and virtual bird's-eye image coordinate frame $\{\mathbf{B}\}$. Each relationship is described using transformation matrices. First, a relational expression between ${}^W \mathbf{x} = [x \ y \ z \ 1]^\top$ and ${}^R \mathbf{u} = [{}^R u \ {}^R v \ s]^\top$ is defined as follow:

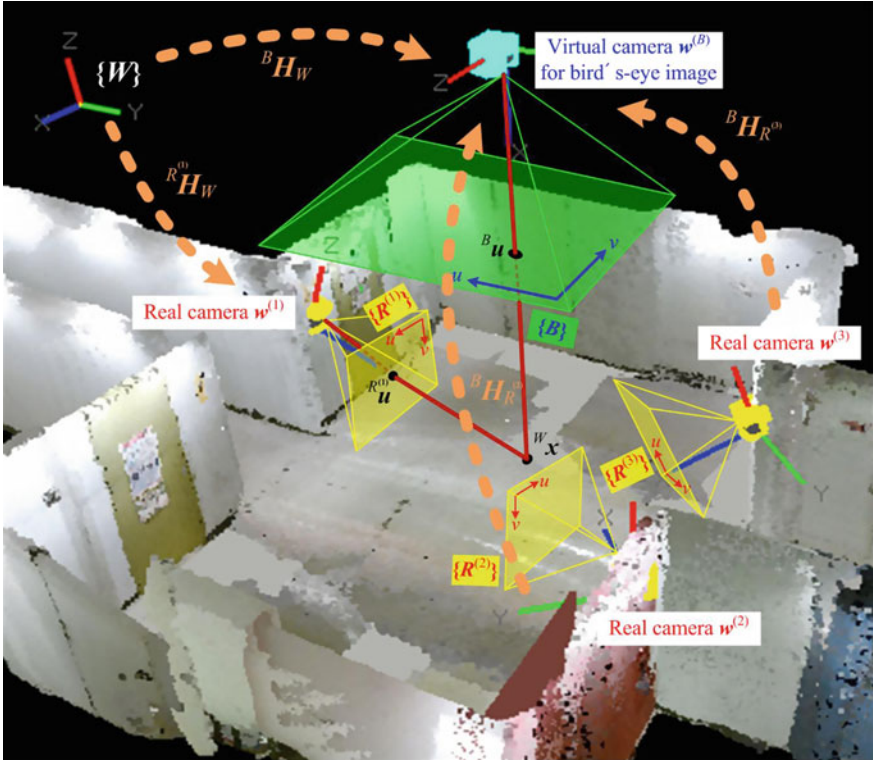


Fig. 3 Perspective transformations between floor plane, real camera images and bird's-eye image for virtual camera

$$\begin{aligned}
 R\mathbf{u} &= R\mathbf{Q}_W^W \mathbf{x} \\
 \begin{bmatrix} R_u \\ R_v \\ s \end{bmatrix} &= \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \end{bmatrix} \begin{bmatrix} W_x \\ W_y \\ W_z \\ 1 \end{bmatrix}, \tag{6}
 \end{aligned}$$

where s denotes a scale factor. The expression can be further simplified under the assumption that the height of the floor surface is zero (i.e., $W_z = 0$), then:

$$\begin{aligned}
 R\mathbf{u} &= R\mathbf{H}_W^W \tilde{\mathbf{x}} \\
 \begin{bmatrix} R_u \\ R_v \\ s \end{bmatrix} &= \begin{bmatrix} h_{11} & h_{12} & h_{14} \\ h_{21} & h_{22} & h_{24} \\ h_{31} & h_{32} & h_{34} \end{bmatrix} \begin{bmatrix} W_x \\ W_y \\ 1 \end{bmatrix}, \tag{7}
 \end{aligned}$$

where $W\tilde{\mathbf{x}} = [W_x \ W_y \ 1]^T$ is the position vector on the world coordinate frame excepting the z element. The matrix $R\mathbf{H}_W$ is called homography matrix in the geometric



Fig. 4 Local bird's-eye images $I_{B(w)}^F$ corresponding to floor images $I_{R(w)}^F$

field. It is possible to project one surface onto another through this homography matrix.

In the same manner, a relational expression between ${}^W\tilde{\mathbf{x}} = [{}^Wx \ {}^Wy \ 1]^T$ and ${}^B\mathbf{u} = [{}^Bu \ {}^Bv \ s]^T$ is also defined:

$${}^B\mathbf{u} = {}^B\mathbf{H}_W {}^W\tilde{\mathbf{x}}, \tag{8}$$

where ${}^B\mathbf{H}_W$ is the homography matrix between the world coordinate frame $\{W\}$ and the bird's-eye image coordinate frame $\{B\}$ when the height of the projection plane is zero.

Finally, substituting Eq. (8) into Eq. (7) yields the relation between the real image coordinate frame $\{R\}$ and the bird's-eye image coordinate frame $\{B\}$:

$${}^B\mathbf{u} = {}^B\mathbf{H}_R {}^R\mathbf{u}. \tag{9}$$

As shown in Eq. (9), the homography matrix ${}^B\mathbf{H}_R$ performs inverse perspective mapping (i.e., the transforming the real camera image into the local bird's-eye image). It is very easy to calibrate the 3×3 homography matrix ${}^B\mathbf{H}_R$ by randomly selecting more than 4 points (i.e., at least four ${}^W\mathbf{x}$) that are commonly observed in the real camera image $\{R\}$ and the bird's-eye image $\{B\}$ because pre-given 3D map information contains the information for the entire environment based on the world coordinate frame $\{W\}$, and thus it can provide information on any point.

Figure 4 shows each local bird's-eye image $I_{B(w)}^F$ that is generated by applying each calibrated homography matrix ${}^B\mathbf{H}_R$ to the floor image $I_{R(w)}^F$. In this case, the virtual camera pose was set to the high position at the center of the entire environment.

2.3 Integration of Local Bird's-Eye Images

After generating each of the local bird's-eye images $I_{B(w)}^F$ by applying the inverse perspective mapping, these images are combined together to create a global bird's-eye image ${}^G I_{B(w)}^F$ that is viewed from direct observation at high location, as follow:

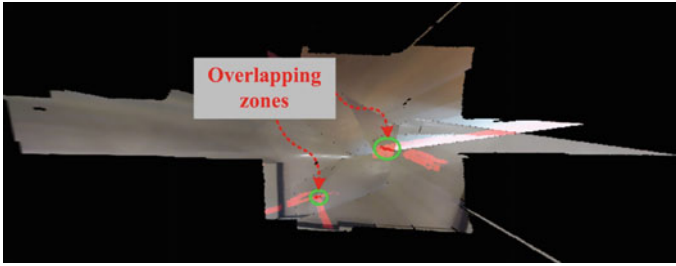


Fig. 5 Integrated global bird's-eye images ${}^G I_{B(w)}^F$

$${}^G I_{B(w)}^F(u, v) = {}^G I_{B(w^{(k^*)})}^F({}^B u^*, {}^B v^*), \quad (10)$$

$$\begin{bmatrix} {}^B u^* \\ {}^B v^* \\ 1 \end{bmatrix} = {}^B H_{R^{(k^*)}} \begin{bmatrix} {}^R u^* \\ {}^R v^* \\ 1 \end{bmatrix}, \quad (11)$$

$$(k^*, {}^R u^*, {}^R v^*) = \arg \min_{k, u, v} \left[I_{R(w^{(k)})}^D(u, v) \right]. \quad (12)$$

Equations (10)–(12) mean that when the same point is observed by multiple cameras, the information encoded from the pixel which has the shortest distance to its camera's optical axis is used. Here, k refers to the camera index (i.e., 1–3 in this example). To calculate the shortest distance, the depth images $I_{R(w)}^D$ for each camera are utilized again here. This is justified as the distortion of the bird's-eye image is smaller when the physical distance in real space is closer.

Figure 5 shows the combined global bird's-eye image ${}^G I_{B(w)}^F$ that is generated by applying Eqs. (10)–(12). Using this approach, it is possible to observe the entire area of the floor that is captured by the camera sensor networks intuitively. Furthermore, the proposed global bird's-eye image can be of great service to moving object's positioning. This will be presented in the following section.

3 Positioning Method for IPS

This section describes positioning method for the proposed IPS by using the generated global bird's-eye image. In this paper, the background subtraction method is used for recognizing moving objects in the image planes. Among the many background subtraction methods, mixture Gaussian model-based method has been used for this study [13]. Red areas in Figs. 2a, d, 4, and 5 show each detection result after applying the background subtraction. As we can see in Fig. 4, the detected objects from single cameras are significantly distorted in the local bird's-eye images because 3D viewpoint transformation is forcibly preformed for the 2D image data in order to

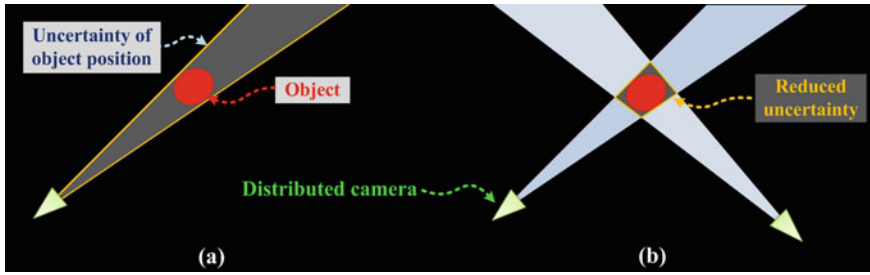


Fig. 6 Principle of distortion removal based on multiple camera observations from different viewpoints: **a** object detected from one camera, **b** object detected from two cameras

generate the local bird’s-eye images. These distortion effects are obviously inefficient in terms of reliable obstacle detection.

In Fig. 5, the dark red areas are the overlapping zones that represent detected moving objects from the multiple camera network. In this study, the distortion effect is solved by taking advantage of these overlapping areas. Thus, if the moving objects are observed from two or more cameras, the distortion effects can be corrected as shown in Fig. 6. 3D-like positioning can be performed in this case without a tedious and convoluted stereo measurement task when the height of the floor can be assumed constant value through the 3D map information.

The principle used to correct these distortions is illustrated in Fig. 6. In contrast to stereo cameras, a single camera cannot measure the distance to the object directly. However, if the object is observed from (i.e., detected from background subtraction in this study) multiple cameras from different viewpoints, then the uncertainty of the distance (i.e., existence region) will be significantly reduced. Note that there is a thread of connections between this principle and well-known monocular SLAM with a single camera [5]. In the global bird’s-eye image with the proposed combined method can remove distortion effects very intuitively, whereas monocular SLAM schemes use convoluted probabilistic processes.

The detected results are composed of pixel coordinates in the global bird’s-eye image frame $\{B\}$; thus, the detected regions ${}^B\mathbf{u}_{IPS}$ as pixel coordinates should be transformed to those of real coordinates ${}^W\tilde{\mathbf{x}}_{IPS}$ through the homography matrix ${}^W\mathbf{H}_B$:

$$\begin{aligned} {}^W\tilde{\mathbf{x}}_{IPS} &= ({}^B\mathbf{H}_W)^{-1}{}^B\mathbf{u}_{IPS} \\ &= {}^W\mathbf{H}_B^B\mathbf{u}_{IPS}. \end{aligned} \tag{13}$$

Here, ${}^B\mathbf{H}_W$ is defined on user-defined internal parameter matrix and external parameter matrix which represents virtual camera pose for generating the bird’s-eye image. Here, its 3rd row vector is excluded because the height of the ground plane is defined as zero. Next, the coordinates ${}^W\tilde{\mathbf{x}}_{IPS}$ detected as the moving objects are converted into a probability distribution function through Kernel density estimation:

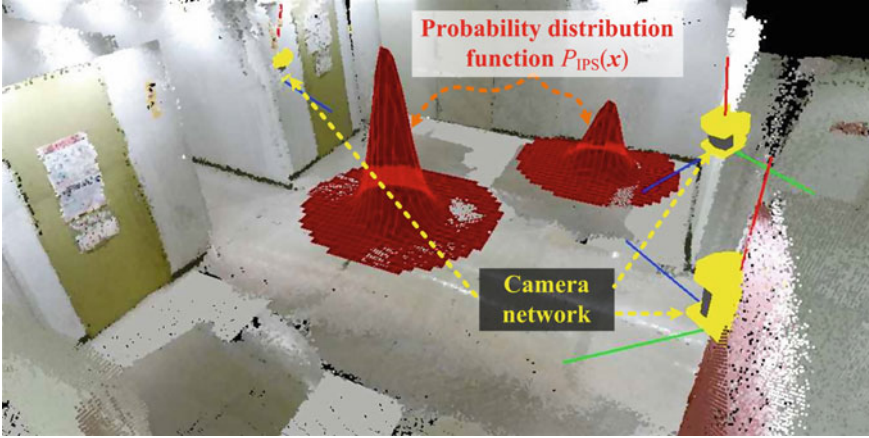


Fig. 7 Probability distribution $P_{\text{IPS}}(\mathbf{x})$ for the existence of moving object detected by proposed IPS system

$$P_{\text{IPS}}(\mathbf{x}) = \frac{1}{Nh^D} \sum_{i=1}^N \mathbf{K} \left(\frac{\mathbf{x} - \mathbf{x}_{\text{IPS}}^{(i)}}{h} \right), \quad (14)$$

$$\mathbf{K}(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}} \exp \left(-\frac{1}{2} \mathbf{x}^T \mathbf{x} \right), \quad (15)$$

where $\mathbf{x}_{\text{IPS}}^{(i)}$ and N denote the coordinates of the detected zone and the number of detected pixels ${}^B\mathbf{u}_{\text{IPS}}$ in the global bird's-eye image. From here, the superscript W which represents world coordinate frame is omitted. h and D are a smoothing parameter and the point dimensions (two in this case), respectively. Equation (15) represents typical \mathcal{O} mean and i covariance Gaussian Kernel function $\mathbf{K}(\cdot)$ which is applied to manage several noises owing to errors from camera calibration, background subtraction, and so on. Figure 7 shows generated final IPS information $P_{\text{IPS}}(\mathbf{x})$, which represents the existence probability distributions of the moving objects detected by multiple camera sensor networks.

4 Experimental Result for Safe Path Generation

This section presents an experimental result for safe path generation in a human-robot coexistence environment based on a gradient method with 3D map information and the information from camera sensor networks. Most obstacle avoidance schemes were based on processing on-board sensor (i.e., a sensor mounted on robot's body) information so far, and thus these cannot deal with invisible obstacles because of the sensing scope. In this study, therefore, in order to take invisible dynamic obstacles

into consideration, occluded obstacles are detected by processing the global bird's-eye view image generated from camera sensor networks installed in the human-robot coexistence environment. In other words, this section assumes that generated IPS information represents the positions of the obstacles. The possibility of collision is basically reflected to path generation and motion control algorithms. Major scope in this section is generating safe paths for the mobile robot, taking not only visible obstacles but also occluded moving obstacles into consideration. Most path planning methods generate optimal paths based on only visible obstacles and the optimality is basically defined in terms of distance from a current robot position to a goal position. However, the shortest path might be not the safest path for the robot motion. The path should reflect the risk of possible collision with occluded invisible obstacles. To this end, the proposed path generation scheme calculates posterior existence probabilities of dynamic obstacles including occluded ones to redefine the cost function of a gradient method [10]. By exploiting the redefined cost function, safe path taking dynamic obstacles' existence probabilities into account is generated.

The gradient method generates a minimum distance path without local minima problem, and it is the most widely used method. The original gradient method generates an optimal path based on intrinsic costs and adjacency costs which are allocated to every grids of the map information. The intrinsic costs are assigned for the distance from the static obstacles which are represented in the map information and the adjacency costs are assigned for the distance from the goal position. In addition to these costs, the modified gradient method proposed in this study calculates an additional risk costs that correspond to moving obstacles which are detected by the global bird's-eye view image generated from the multiple camera images. This is done in order to perform safer path planning also considering occluded zones in real time. Here, the probability distribution $P_{IPS}(x)$ (i.e., IPS information) which is computed by Eqs. (14) and (15) can be directly exploited, as the additional risk cost relates to the existence probability for the moving obstacles in the entire environment. What this entail is that positions which have high value of $P_{IPS}(x)$ are more likely have a moving obstacle.

The experiments were conducted under the same conditions using two different methods: the conventional gradient method considering only the intrinsic cost and the adjacency cost and the proposed modified gradient method considering the additional risk cost representing the moving obstacles. The generated path using each method is illustrated in Fig. 8. The generated path using the conventional gradient method generates an unsafe path (blue lines in Fig. 8) around the moving obstacles because the static model (i.e., the map information) cannot manage the dynamic information in real time, and thus it cannot be reflected in the intrinsic cost for the gradient method. On the other hand, the modified gradient method produced a safer path (red lines in Fig. 8) taking moving obstacles into account as the risk costs for the moving objects detected from the global bird's-eye view image were calculated and applied to the total navigation costs for optimal path generation. In conclusion, the proposed path planning scheme based on the modified gradient method considering occluded moving obstacles is expected to reduce collision risk in terms of the mobile robot's motion control.

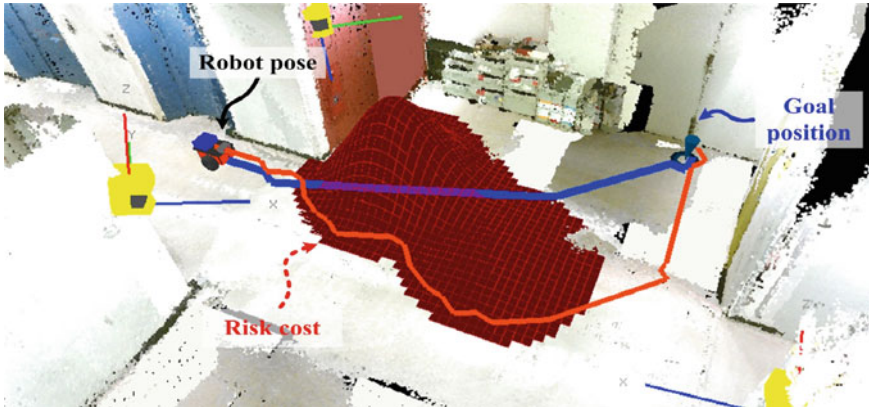


Fig. 8 Experimental results for path generation. *Blue* and *red* lines represent generated paths using typical gradient method and modified gradient method, respectively

5 Conclusion

This paper proposed a novel IPS that uses distributed camera sensor networks for the mobile robot navigation. In order to generate reliable IPS information from the camera networks, a novel method to generate a global bird's-eye image was proposed by using 3D map information. Here, homography matrices for transforming each real image to each local bird's-eye image were automatically calibrated using 3D map information which had all 3D coordinate data for the entire environment. The global bird's-eye view image generated from the multiple camera images was used to detect moving obstacles (i.e., generation of IPS information), and the following conclusion was drawn.

The typical path planning methods so far cannot manage occluded obstacles due to limitations of the on-board sensor's sensing scope. By using the information of the moving obstacles detected as an additional cost function to the gradient method, a safer path can be generated. Thus, mobile robots are expected to reduce their collision risks.

Future work will involve solving a problem of the optimal distributed camera placement by maximizing the coverage of the demands.

Acknowledgements This work was in part supported by Tough Robotics Challenge, ImPACT Program (Impulsing Paradigm Change through Disruptive Technologies Program).

References

1. Bertozzi, M., Broggi, A., Fascioli, A.: Stereo inverse perspective mapping: theory and applications. *Image Vis. Comput.* **16**(8), 585–590 (1998)
2. Bršćić, D., Hashimoto, H.: Model based robot localization using onboard and distributed laser range finders. In: Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2008), 1154–1159 (2008)
3. Chang, N., Rashidzadeh, R., Ahmadi, M.: Robust indoor positioning using differential Wi-Fi access points. *IEEE Trans. Consum. Electron.* **56**(3), 1860–1867 (2010)
4. Curran, K., Furey, E., Lunney, T., Santos, J., Woods, D., McCaughey, A.: An evaluation of indoor location determination technologies. *J. Locat. Based Serv.* **5**(2), 61–78 (2011)
5. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(6), 1052–1067 (2007)
6. Dellaert, F., Fox, D., Burgard, W., Thrun, S.: Monte carlo localization for mobile robots. In: Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA1999), 1322–1328 (1999)
7. Jayasekara, P.G., Hashimoto, H., Kubota, T.: Simultaneous localization assistance for mobile robot navigation in real, populated environments. *SICE J. Control, Meas. Syst. Integr.* **5**(6), 349–358 (2012)
8. Ji, Y., Yamashita, A., Asama, H.: Automatic calibration and trajectory reconstruction of mobile robot in camera sensor network. In: Proceedings of the 11th Annual IEEE International Conference on Automation Science and Engineering (CASE2015), 206–211 (2015)
9. Ji, Y., Yamashita, A., Asama, H.: Automatic calibration of camera sensor networks based on 3D texture map information. *Robot Auton Syst.* (2016). doi:[10.1016/j.robot.2016.09.015](https://doi.org/10.1016/j.robot.2016.09.015)
10. Konolige, K.: A gradient method for realtime robot control. In: Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2000), 639–646 (2000)
11. Rahimi, A., Dunagan, B., Darrell, T.: Simultaneous calibration and tracking with a network of non-overlapping sensors. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2004), 1–287 (2004)
12. Reza, A.W., Geok, T.K.: Investigation of indoor location sensing via RFID reader network utilizing grid covering algorithm. *Wirel. Pers. Commun.* **49**(1), 67–80 (2009)
13. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR1999), 246–252 (1999)

Precise and Reliable Localization of Intelligent Vehicles for Safe Driving

Liang Li, Ming Yang, Lindong Guo, Chunxiang Wang and Bing Wang

Abstract Autonomous driving technology has become a spotlight in recent years. Of all the factors related to autonomous driving, safety should be first considered. A safe global trajectory should be planned at beginning and local safe trajectory should be planned according to the situations in real time. Due to this, the intelligent vehicles must know where they are in real time to do the next control steps. In this paper, a high-precision localization framework for intelligent vehicles is proposed. A vertical low-cost LIDAR is used for mapping and live data collection. High-precision maps are generated by projecting laser scans along the survey trajectory produced by trajectory filter. When localizing, an improved matching method particle Iterative Closet Point is proposed. Using this particle ICP, not only the matching precision is improved, but also the computing time decreases remarkably, which helps to make the algorithm real-time. Decimeter-level precision can be achieved by the validation of experiments. The results show much benefit for safe driving by this Monte Carlo framework.

Keywords Precise localization · Intelligent vehicle · Safe autonomous driving · Particle icp · Vertical lidar

1 Introduction

Decades have been passed since the emergency of intelligent vehicles. All the developed technologies aim to achieve two main goals: (1) the autonomous vehicle travels correctly; (2) the autonomous vehicle travels safely. Much effort has been devoted

L. Li · M. Yang (✉) · L. Guo · B. Wang

Key Laboratory of System Control and Information Processing, Ministry of Education of China, Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China
e-mail: MingYang@sjtu.edu.cn

C. Wang

Research Institute of Robotics, Shanghai Jiao Tong University, Shanghai 200240, China

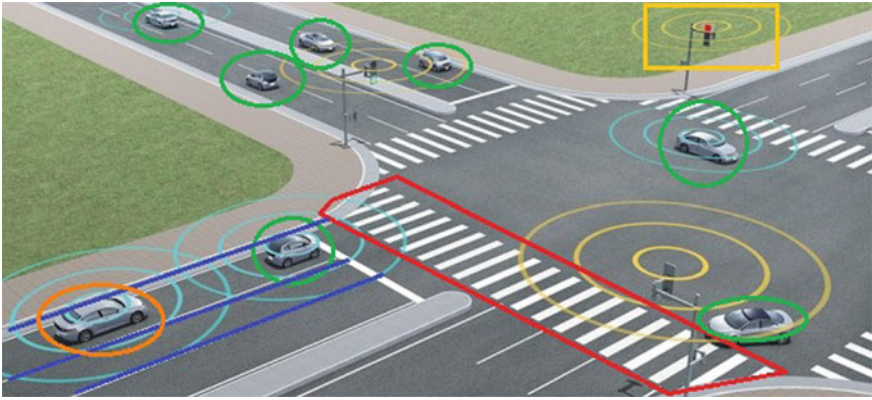


Fig. 1 An example of how high-precision localization improves the safety of autonomous driving. The vehicle in the *orange circle* is our vehicle, the vehicles in the *green circles* are other vehicles. The *yellow rectangle* circles the infrastructures. With Internet of Vehicles, V2V or V2I equipments, the relative position between our vehicle and others or the infrastructures can be known. With the help of high-precision maps, the zebra crossing (in the *red polygon*), the lane markings (denotes by the *purple lines*) and all the other signs on the road surface will be known to our vehicle too. Based on these, the fully autonomous driving is not hard to be imagined

into making the intelligent vehicles safer ever since. Many achievements have been made in this field, like Advanced Driving Assistance System (ADAS), collision avoidance system, etc. In the previous study for making intelligent vehicles safer, one factor is often ignored, that is precise localization. If precise localization can be achieved, many safety techniques could be simplified largely. When the exact position and pose of the vehicle is known, the relative position of vehicles around and infrastructures around will be known through Internet of Vehicles, thus collision avoidance will be much easier. Besides that, the high-precision localization result will benefit local and global safe path planning. Based on precise localization result, real-time perception will be easier at the same time. If the vehicle knows where it is exactly, the environment around can be inquired. The relative position between the vehicle and traffic signs will be clear, so the traffic sign detection result will be improved, which will make the vehicle driving autonomously more safely. This is vividly illustrated in Fig. 1. If high-precision localization could be achieved, together with high-precision maps and Internet of Vehicles, the surroundings of the vehicles is totally known to itself. Then autonomous driving safely is not a dream any more.

Much research has been done towards high-precision localization for intelligent vehicles. Vision based methods have a history of more than 10 years. These methods rely on feature maps, which is like Simultaneous Localization And Mapping (SLAM). The vision-based features such as SIFT, SURF, DIRD, etc. are extracted and stored in the prior-maps. When localising, some image registration methods are utilised. The previous studies show that localization precision can be good (maybe centimeter-level) when environment meets the requirements. While like many other

vision-based methods, the defects of it is also obvious. It maybe easily affected by environment changes, like weather, season, light condition changes. So robustness is the main obstacle to prevent vision-based methods to be widely used.

In order to overcome the shortcomings brought by vision-based methods, LIDAR-based high-precision localization methods [1] have been paid more and more attentions in recent years. 3D LIDAR units are favored by many authors, they can afford ample environmental information, which can be used to localise the vehicles directly. In this work, we do not rely on this expensive, complex actuated sensors. Some authors made use of much cheaper 2D LIDAR to achieve the same goal and have done some impressive work. In [2], Sheehan M et al. optimized the Kernelised Rényi Distance (KRD) function between the prior map and live measurements from a 3D laser sensor. After this, the continuous-time trajectory of a vehicle was estimated by finding a sequence of Catmull-Rom splines. I. Baldwin et al. [3] used a single pushbroom 2D LIDAR which was oriented downwards to produce 3D swathe and a prior 3D survey. The swathe generation was based on estimates of vehicle velocity which could be obtained from vehicle speedometers after calibration, i.e., removing the sensors' bias. Kullback-Leibler divergence was used to compare swathe and prior-maps. After a 26 Km of driven path, it was shown that the system outperformed a high caliber integrated DGPS/IMU system. In [4], A Monte Carlo Localization scheme was adopted for vehicle position estimation, based on synthetic LIDAR measurements and odometry information. It relied on features that were perpendicular to the ground(walls, etc.), which maybe a defect when considering its robustness. Maddern [5] et al. leveraged experience generated by 2D laser scans along the vehicle trajectory to optimize the cost function for large-scale localization in changing cities. They exploited GPU rendering pipeline to accelerate the computing process and can provide real-time localisation at 5 Hz.

Different from all the previous studies, in this paper we exploit the Particle Iterative Closest Point (PICP) method to match the live point cloud against the prior maps. The maps are built from laser scans and vehicle pose and position by trajectory filter, which is originated from Extended Kalman Filter. The hardware framework is low-cost. The remainder of the paper is organised as follows. Section 2 depicts the process of mapping and live point cloud generation. In Sect. 3, the localization process is presented. The experiments setup, results and data analysis are shown in Sect. 4. Finally, the paper is concluded in Sect. 5.

2 Mapping and Point Cloud Generation

In this section, the approach of mapping and live point cloud generation process will be presented. First the vehicle's trajectory is estimated by trajectory filter. Then the data collected by LIDAR are projected along the estimated trajectory. There is a bit difference between mapping and live point cloud generation. When mapping, the trajectory is estimated as a whole curve from beginning to the end. It means that if only relied on Inertial Measurement Unit (IMU) and odometry, accumulative error

will be the outstanding issue when the vehicle travels for a relative long distance. Due to this, high-precision GPS is induced. The effect of the high-precision GPS is obvious, for it will correct the accumulative error brought by IMU. While during live point cloud generation process, only a short length of point cloud (in this paper 20 m) is needed, the accumulative error can be neglected. So we don't rely on GPS in the process of live point cloud generation.

2.1 Vehicle Trajectory Estimation

In previous studies, vehicle trajectory is merely estimated by simple integration of data collected by inertial sensors. This will bring accumulative error problem when the scale of the map is large. Based on Extended Kalman Filter (EKF), the trajectory filter is used to estimate the vehicle trajectory with high precision in this paper, even if the scale of the map is large. The equations of EKF can be found in [6], due to space limit, we don't list them here. In the trajectory filter, three kinds of data are fused together, i.e., latitude and longitude provided by high-precision GPS, rotational data provided by IMU and translational data provided by odometer (wheel encoder). Since latitude and longitude cannot be input to trajectory filter directly, we need to transform the latitude and longitude into data in plane coordinate. In this paper, the cylinder model is used, which can be expressed as:

$$x = S \cdot lon \cdot \frac{\pi}{180} \cdot R - x_0 \quad (1)$$

$$y = S \cdot R \cdot \log(\tan(90 + lat)) \cdot \frac{\pi}{360} - y_0 \quad (2)$$

In the above equations, x and y are the transformed plane coordinates in two directions. lat and lon are latitude and longitude value of that point. x_0 and y_0 are the plane coordinate values of the origin. R is the radius of the earth. $S = \cos(x_0 \cdot \frac{\pi}{180})$. The origin is the selected mean point in that area. Using this method, the transformation result will be more precise. The next step of the trajectory filter is estimating the vehicle's position and pose using the transformation plane coordinates, IMU and odometry data. The data transformed from GPS are treated as observation data. The distance increment and angle increment are treated as prediction data. The Kalman gain which is very important in EKF here is

$$K_{gain} = P_{pre} G^T (G \cdot P_{pre} \cdot G^T + V \cdot R_{obv} \cdot V^T)^{-1} \quad (3)$$

In Eq. (3), P_{pre} is the covariance matrix of the last state, G is the observation matrix. The observation noises matrix is denoted as V . R_{obv} is the updated state vector. Besides avoiding from accumulative error problem, this trajectory filter can also solve the problem brought by different frequency of different sensors. For example,

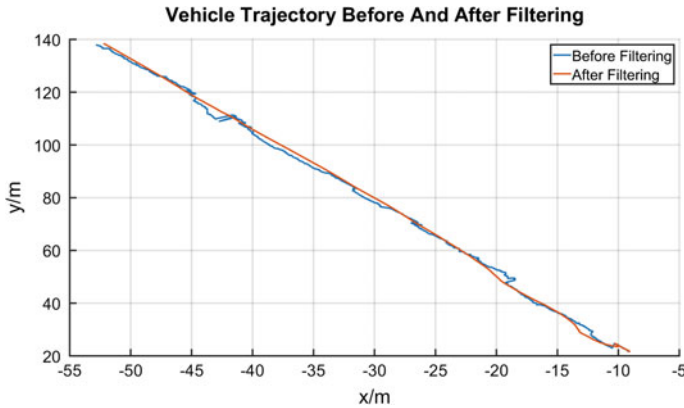


Fig. 2 An comparison between the original vehicle trajectory (*blue*) and vehicle trajectory after trajectory filtering (*orange*). After filtering, the trajectory is smoother. Moreover, it will not be influenced by accumulative errors brought by IMU or odometer. The heading yaw angle of the vehicle can also be estimated by this filter

the frequency of GPS is 10 Hz, while the frequency of Xsens MTi is 100 Hz. The online trajectory is also gotten by this filtering method, by with common GPS. An example of the vehicle trajectory produced by trajectory filter can be seen in Fig. 2. In the figure, the comparison of the trajectory before and after filtering is also presented.

2.2 Point Cloud Generation

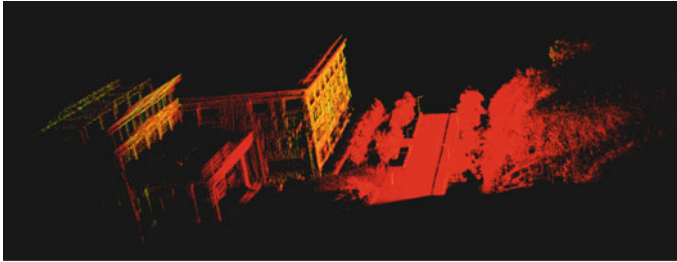
Due to space limit, the calibration of the LIDAR is not depicted in detail here. While it has been done before point cloud generation. First, every scan should be transformed with its corresponding position and pose. As the vehicle goes ahead, the position and pose the scan also changes. Then the scans are added together to generate the point cloud. The process can be expressed as:

$$\mathbf{Q} = \int_{t_0}^{t_f} T(s(t), p(t)) dt \quad (4)$$

In Eq. (4), \mathbf{Q} is the generated point cloud, $s(t)$ is the laser scan at time t , $p(t)$ is the pose and position at time t . t_0 and t_f are the beginning and end time of the scan time. When localizing, matching between the live point cloud and the whole map is time consuming and awkward. In this paper, we cut the map into small pieces. Only the most relative pieces will be loaded to match when localizing. So the final map is denoted as: $\mathbf{M} = \{m_1, m_2, \dots, m_n\}$, n is the number of the map pieces. An example of the built map piece can be seen in Fig. 3.



(a) A picture captured by camera



(b) The built map

Fig. 3 One of the built map pieces. **a** A picture captured by camera. **b** The built map of (a)

3 Localization

In this section, the localization method will be presented. The point cloud generated by the scans collected by multi-beam laser scanner is dense, which means there are many points in the point cloud. If using Iterative Closest Point (ICP) directly, it will take a long time to complete matching process. That means the real-time property is far more reachable. Due to this, a particle ICP is proposed to accelerate the matching process in a large scale.

3.1 Particle ICP

Particle Filter (PF) [7] is widely used in state estimation, object tracking, etc. Particle filtering methodology uses a genetic type mutation-selection sampling approach, with a set of particles (also called individuals, or samples) to represent the posterior distribution of some stochastic process given some noisy and/or partial observations. In reality, Particle Filters are a set of genetic-type particle Monte Carlo methodologies to solve the filtering problem. Due to this, it is also called Sequential Monte Carlo (SMC) method. So in this paper, we also draw on the experience of Monte Carlo method. Particle filter techniques provide a well-established methodology [7, 8] for generating samples from the required distribution without requiring

assumptions about the state-space model or the state distributions. The traditional ICP [9] traversals all the points in target points set to seek the closest point of one point in the source set. This process is extremely time-consuming, making the computational complexity $O(n^2)$. In this paper, the number of points to be matched is million-level, the original ICP is not suitable obviously. For the sake of real-time requirement, the particle ICP is proposed.

The core idea of particle ICP is that we don't need to traversal all the points in the target point set. Conversely, The aim of particle ICP is to scan the data points as less as possible at no cost of precision at the same time. The particle are based on two assumptions:

- The points in the point cloud are continuously distributed.
- The closest data points corresponding to two spatially close model points are also close each other.

According to PF, what particle ICP does is keeping sampling to convergence. At first, the particles are sampled equally in the point cloud. In the sampled points, the closest one will be selected, then keep sampling around this point. Doing the above until the convergence condition is fulfilled. The detail of this algorithm can be seen in Algorithm 1.

Algorithm 1 Particle Iterative Closest Point Algorithm

```

1: procedure PICP(P, Q)
2: Taking a point in P randomly, denoted as  $p_1$ .
3: Equally sampling in Q, producing k sampling seeds.
4:  $q_{11} = C(p_1, \{q_1^1, q_2^1, \dots, q_k^1\})$ .
5: repeat
6:   Equally sampling around  $q_{1i}$ .
7:   Do same as 4.
8: until  $d_1 < \delta$ .
9: for  $i = 2; i < num; i + +$  do.
10:    $l = \mathbf{d}(p_i, p_c)$ .
11:   Sampling Q according to  $l$  in 10.
12:   Do procedure 4-8.
13: end for
14: The corresponding points set Qc is gotten.
15: repeat
16:   Do procedure 2-14.
17: until The convergence condition is met.
18: T = Trans(P, Qc).
19: Return(T)
20: end procedure

```

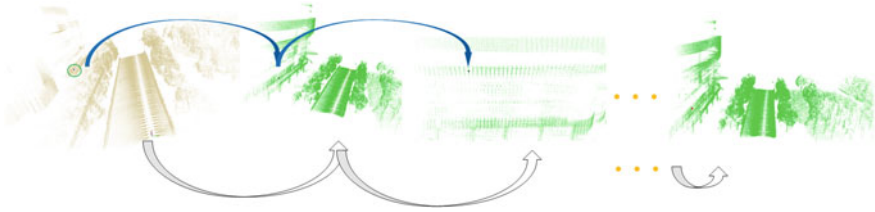


Fig. 4 Particle Iterative Closest Point Algorithm. First, randomly taking a point in the source point set, denoted as a *blue* point in the *green circle* of the first figure. Equally sampling in the target point set, denoted as *red* points in the second figure. Then computing the relative distance between the source point and every sampling point, selecting the closet point, denoted as *black* point marked in the second point \mathbf{P}_1 . After this, equally sampling around the marked point mentioned above in the envelop of the closet points around \mathbf{P}_1 . Repeat these procedures iteratively until convergence. Finally, the closet point is found, denoted as \mathbf{P} . When computing the other points in the source point set, considering the relative space relation between the already found closet points

In Algorithm 1, \mathbf{P} and \mathbf{Q} are the source point cloud and target point cloud, respectively. The function $\mathbf{C}(x, Y)$ finds the closest point to x in Y . d_1 is the distance between p_1 and its corresponding point in \mathbf{Q} , q_1 . num is the total number of points in \mathbf{P} . $\mathbf{d}(x, y)$ is a function to compute the distance between x and y . p_c is a set of points in \mathbf{P} which are selected as benchmark points. According to l , the resampling points in \mathbf{Q} can be estimated. This is based on the assumption that the closest data points corresponding to two spatially close model points are also close each other. In line 18, the transformation matrix \mathbf{T} between \mathbf{P} and \mathbf{Q} is estimated by computing the transformation relation between \mathbf{P} and its corresponding points set in \mathbf{Q} , \mathbf{Q}_c .

The whole process of the Particle Iterative Closet Point Algorithm is illustrated in Fig. 4.

3.2 Particle ICP Based Localization

The core process of localization is matching between the prior-maps and the live point cloud data. In this paper, the matching algorithm is Particle Iterative Closest Point (PICP) proposed above. The localization includes two procedures: (1) weak GPS localization, (2) PICP based localization. The weak GPS localization provides approximate position of the vehicle, i.e., the estimated position by this method is with high covariance. So it is called weak localization. Though weak, it will benefit the strong point cloud localization largely. According to weak localization result, the map pieces which will be loaded to be served as local map can be estimated at a high confidence level. It also can solve the “kidnapped robot” problem in a large degree.

Algorithm 2 Particle Iterative Closest Point Algorithm

```

1: procedure PICP based localization algorithm
2: repeat
3:   if GPS is available then
4:      $\mathbf{P}_w = W(\text{GPS}, \text{DR})$ 
5:   else
6:      $\mathbf{P}_w$  is estimated by DR
7:   end if
8:    $m_i = L(\mathbf{M}, \mathbf{P}_w)$ 
9:    $\mathbf{M}_c = \{m_{i-1}, m_i, m_{i+1}\}$ 
10:   $\mathbf{T} = \text{PICP}(\mathbf{P}, \mathbf{M}_c)$ 
11:   $\mathbf{P}_c = \text{Trans}(\mathbf{P}_m, \mathbf{T})$ 
12: until The localization module is closed
13: end procedure

```

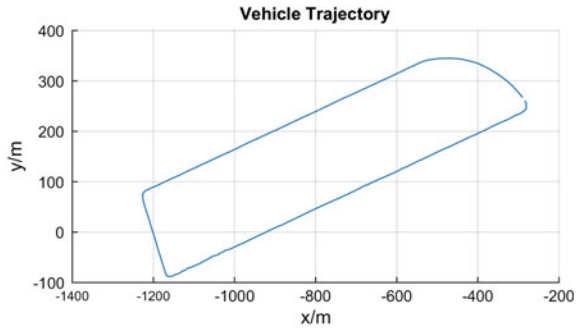
After weak GPS localization, the Particle ICP based method will be utilized to achieve centimeter-level localization. The detail explanation about the proposed Particle ICP based localization is given in Algorithm 2. In Algorithm 2, \mathbf{P}_w is the position of the vehicle estimated by weak localization. Function W computes the coarse position of the vehicle. m_i is the i -th map-piece which is nearest to the current position. \mathbf{M}_c is the current map constituted by m_{i-1} , m_i and m_{i+1} . \mathbf{T} is the transformation matrix produced by PICP method, which can be seen in Algorithm 1. Then the current precise pose and position of the vehicle can be estimated by transforming the pose and position labeled on the current map \mathbf{P}_m with transformation matrix \mathbf{T} .

4 Experimental Results and Analysis

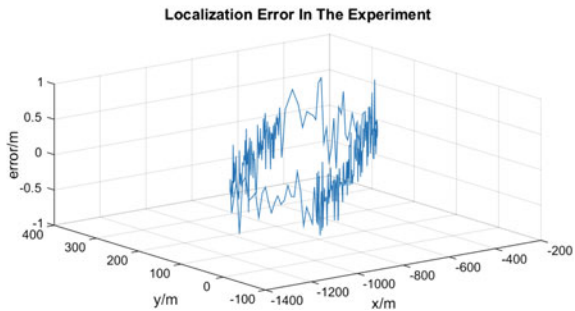
In this section, the experiments, results and data analysis will be presented. The experimental platform is a CyberTiggo Sport Utility Vehicle. A Velodyne VLP-16 LIDAR is mounted in front of the vehicle, its scanning surface is vertical to the road surface. The experimental environment is Shanghai Jiao Tong University campus. First, we validate the precision of localization by method in this paper. The ground truth is estimated by RTK-GPS. Usually, its precision is less than 10 cm. A comparison between the ground truth and the estimated result is illustrated in Fig. 5. Three degrees of freedom are considered, i.e., lateral, longitudinal and heading. From the figure, it can be seen that the lateral and longitudinal errors fall in 50 cm mostly. The vast majority of heading error falls in 1° .

The distribution of localization errors can be seen in Fig. 6a. The heading error distribution can be seen in Fig. 6b. From the above figure, it is obvious that with the high-precision localization, the distance error is limited to 1m, while the heading error is limited to 2° . With localization precision at this level, an autonomous vehicle will travel safely in most situations. Comparing with Dead Reckoning (DR), this method is much more precise and does not suffer from accumulative errors. Comparing to vision based localization, this method is not affected by light condition

Fig. 5 Localization error during the experiment. **a** The vehicle trajectory estimated by the localization method in this paper. **b** A plot of the localization error along the trajectory. The ground truth in the experiment is collected by RTK-GPS. The errors are all within 1 m



(a) Vehicle trajectory in the experiment



(b) Localization error during the experiment

changes. With this precision, it is enough to navigate the vehicle, optimize traffic sign detection, road curb detection, etc. The detection precision and robustness is significant to the safety of autonomous driving. A summary of the localization performance is presented in Table 1. From the table, it can be seen that the localizability parameter σ are small, which indicates the proposed method performs well in most cases.

The main aim of this paper is to make the autonomous driving more safe. So a set of experiment is done to verify validation of safe autonomous driving using method proposed in this paper. In the experiment, an emergency brake is considered as a dangerous behavior or not safe action. The less emergency brakes, the safer the autonomous driving is. In this experiment, the vehicle is navigated from start point to the finish point with high-precision localization proposed in this paper. Then navigating the same vehicle in the same route but without high-precision localization. The numbers of emergency brakes under these two situations are counted in Table 2. From the table, it can be seen that when utilizing high-precision localization algorithm along the traversal, the mean emergency brakes times decrease from 9.2 to 1.7, which indicates a remarkable effect of the high-precision localization for safely autonomous driving. One of the trajectory with emergency brake points are shown in Fig. 7, the different colors denote the navigating route with and without

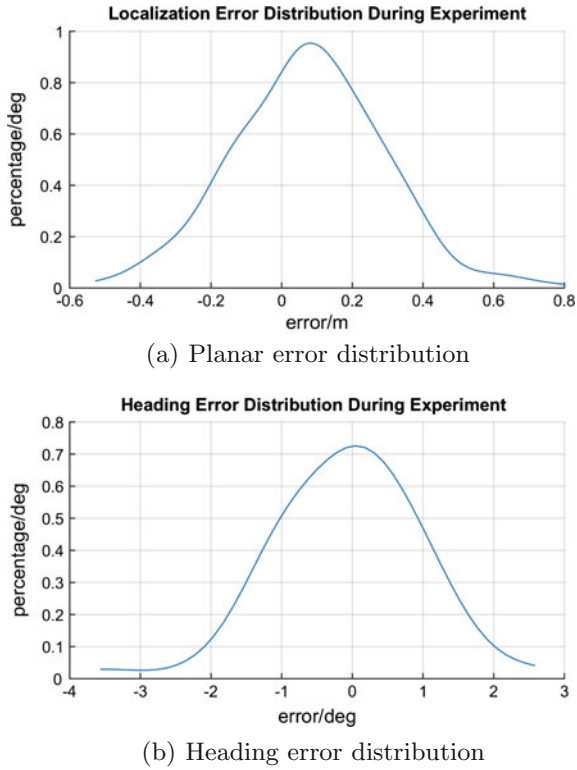


Fig. 6 Localization error distribution during the experiment. The errors lie in the interval $[-0.4, 0.6]$ and $[-0.2^\circ, 0.2^\circ]$ mostly

Table 1 Localization performance during experiment

Parameters	Mae	Max	90% CI
Lateral(x)	0.35 m	0.49 m	0.30 m
Longitudinal(y)	0.40 m	1.0 m	0.40 m
Heading(θ)	0.73°	2.57°	1.00°

Table 2 Number of emergency brakes and localizability with and without high-precision localization

Parameters	With	Without
Number of experiments	10	10
Total brakes	17	92
Mean brakes	1.7	9.2

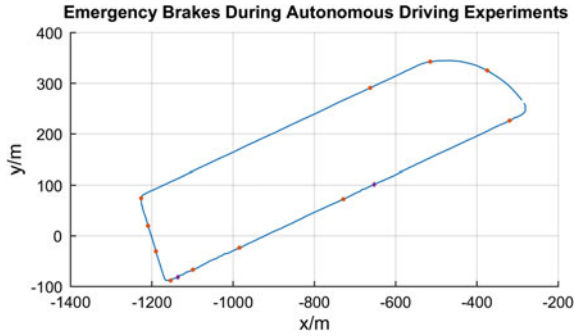


Fig. 7 The number of emergency brakes during experiments. The *blue lines* shows the trajectory estimated by the Particle based localization. The *orange dots* denote emergency brakes without the high-precision localization. While the *purple diamonds* denote emergency brakes with the high-precision localization. Clearly, with the high-precision localization, the emergency brakes fall from 11 down to 2

high-precision localization. In the figure, the red dot presents an emergency brake without high-precision localization. While the blue dot presents an emergency brake with high-precision localization. It can be seen that the blue dots are much less than the red dots, which indicates that using the high-precision localization method proposed in this paper, the autonomous driving is much safer.

5 Conclusion

In this paper, a novel high-precision localization framework is proposed using a low-cost 3D LIDAR which is vertical to the road surface. The whole process is matching the live point cloud data against the prior maps to get the transformation relation \mathbf{T} between them. Then the current high-precision pose and position of the vehicle can be estimated after referencing the position labeled on the maps and transforming it with the transformation matrix \mathbf{T} . When mapping, the vehicle trajectory is estimated by vehicle trajectory filter with temporal and spatial information collected by high-precision GPS, Inertial Measurement and odometer. The matching algorithm is Particle Iterative Closest Point proposed in this paper. PICP algorithm combines Particle Filter (PF) and ICP, which will decrease run time to fulfill real-time property. Experiments validate the precision and real-time property of this system. At the same time, the improvement of safety after using the high-precision localization method is also verified. In the future work, we will compress the size of map to localize in large areas e.g., city-level localization. Besides that, some moving objects detection algorithms should be adopted to decrease the influence of outliers.

Acknowledgements This work was supported by the National Natural Science Foundation of China(91420101), International Chair on automated driving of ground vehicle, National Magnetic Confinement Fusion Science Program(2012GB102002).

References

1. Levinson, J., Thrun, S.: Robust vehicle localization in urban environments using probabilistic maps. In: 2010 IEEE International Conference on Robotics and Automation (ICRA). IEEE (2010)
2. Sheehan, M., Harrison, A., Newman, P.: Continuous vehicle localisation using sparse 3d sensing, kernelised réyi distance and fast gauss transforms. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2013)
3. Baldwin, I., Newman, P.: Road vehicle localization with 2D push-broom LIDAR and 3D priors. In: 2012 IEEE international conference on Robotics and automation (ICRA). IEEE (2012)
4. Chong, Z.J., et al.: Synthetic 2D LIDAR for precise vehicle localization in 3D urban environment. In: 2013 IEEE International Conference on Robotics and Automation (ICRA). IEEE (2013)
5. Maddern, W., Pascoe, G., Newman, P.: Leveraging experience for large-scale LIDAR localisation in changing cities. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE (2015)
6. Fujii, K.: Extended Kalman Filter. Reference Manual (2013)
7. Del Moral, P.: Non-linear filtering: interacting particle resolution. *Markov Process. Relat. Fields* **2**(4), 555–581 (1996)
8. Del Moral, P.: Feynman-Kac formulae. *Genealogical and interacting particle approximations* (2004)
9. Best, P.J., McKay, Neil D.: A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 239–256 (1992)
10. Liu, Z., et al.: Action selection for active and cooperative global localization based on localizability estimation. In: 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE (2014)

A Unified Controller for the Connectivity Maintenance of a Robotic Router Networks

Li Xiangpeng, Huang Haibo, Yang Hao and Sun Dong

Abstract The robots equipped with wireless networking modules can act as mobile routers to bridge the communications of a network. How to adjust the robot network topology and the motion of mobile routers adaptively to support the communication connectivity to mobile users in group task execution is still remained challenging. In this paper, we addressed this challenging by developing a unified motion controller to drive the robots to approach their individual task region while maintaining the desired network topology and keeping collision-free with obstacles in environment. To achieve this, a new concept termed rubber communication model is first proposed to evaluate the real communication signal, which enables adding and removing communication links amongst robots. Then, a continuous model for collision avoidance is utilized for avoiding obstacles. Together with the rubber communication model and continuous model for collision avoidance, the tasks assigned to the robots are modeled as series of geometrical task regions which is formulated with the regional reaching constraint function. The three models are utilized in building the potential field function, based on which a bounded control input is generated for multirobot control. Simulations are finally performed on a group of mobile robots to demonstrate the effectiveness of the proposed controller.

Keywords Connectivity maintenance · Dynamic topology · Obstacle avoidance

L. Xiangpeng (✉) · H. Haibo
Robotics and Microsystems Center, Soochow University, Suzhou, China
e-mail: licool@suda.edu.cn

L. Xiangpeng · Y. Hao · S. Dong (✉)
Department of Mechanical and Biomedical Engineering, City University
of Hong Kong, Hong Kong, China
e-mail: medsun@cityu.edu.hk

1 Introductions

In recent years, mobile router configuration problem arises a new challenging in the robot network. The problem we investigated is how to control a group of robots to accomplish the task objective while maintaining the communication connectivity of the mobile user to a base station and keep collision free with obstacles in environment.

Several motion control schemes are closely relates to the problem we investigated. In [9], a control framework was designed for a group of mobile robots in making decision locally to maintain communication connectivity between a moving independent target and a station. Two matrixes were used to characterize the network connectivity in this study. Similar works also been done by [10], where a motion planning algorithm is proposed to keep the connectivity of a base station to a cooperative (or non-cooperative) user. In [2], a dual ascent algorithm and an augmented Bellman-Ford algorithm was proposed to maintain the connectivity between an UAV to a base station in surveillance task. The optimum router configuration approach for the connectivity maintenance between a transmitter and a receiver developed by [11] aiming to minimize the Bit Error Rate. The motion control of mobile routing configuration problem of multiple mobile users to a base station can be found in [1, 3, 8]. All these works above only investigate the problem of how to control the motion of mobile routers to fulfill the routing requirements, and only works in static network topology.

In this paper, we proposed a unified controller, which controls the motion of mobile routers adaptively to accomplish a cooperative exploring task while maintaining the connectivity of the exploring robot and a base station and keeping collision free with all obstacles in environment. A new concept termed rubber communication model was first proposed to evaluate the real communication signal. Based on which an objective function which enables adding and removing communication links was established for connectivity maintenance. Extending the concept of virtual obstacle in our formal work (Li et al. [5], a continuous model is designed for smooth obstacle avoidance. A regional reaching concept was then introduced in building the task reactive objective function for the robot routers. The three components are then incorporated into the design of a potential filed function, based on which a motion controller is designed as a negative gradient of this potential field function.

The contributions of this paper are twofold. First, we propose a new model termed rubber communication model to evaluate the real communication signal, based on which, the motion controller fulfills both connectivity maintenance and topology switching requirements simultaneously was proposed. Second, we firstly characterize the geometrical constraint for the mobile routers as specific geometrical regions, by incorporating the regional reaching concept into the potential field design, a new solution for solving motion control of the mobile networks is

provided. Compare to most of the existing works, the proposed motion controller drives mobile routers to maintain the communication connectivity in the presence of obstacles, meanwhile, enables the dynamic topology switching.

2 Problem Formulation

We consider a group of n mobile robots equipped with wireless communication devices and limited sensing capabilities in a workspace $W \in \mathbb{R}^2$ to accomplish a cooperative task specified by a target position g to be explored. During task execution, the robot network should stay connected all the time with a static base station.

2.1 Definitions and Notations

The robot is modeled by a point with position coordinate as $q_i(t) \in \mathbb{R}^2$ with a single integrator dynamics expressed as follows

$$i \in \{1, \dots, n\} \tag{1}$$

where u_i denotes the velocity control input for robot i .

For convenience, the static base station is indexed by 0 with position coordinate as $q_0 \in \mathbb{R}^2$. Define the robot selected to explore the target position as the “mobile user” and labeled with symbol R_u , and the other robots are called as mobile routers.

Consider that there are m sphere obstacles located in the workspace. The coordinate of the ζ th obstacle is denoted by $q_o(\zeta) \in \mathbb{R}^2$ with radius of $\rho_o(\zeta) \in \mathbb{R}^2$, where $\zeta \in \{1, \dots, m\}$ is the index of obstacles.

Due to the limited communication capabilities of robots, the communication topology of the robot network is associated with a proximity graph. For simplicity, we assume that the communication relationship between two robots is symmetric.

Definition 1 (Proximity Communication Graph) The proximity communication graph $C(t) = (V, E)$ is an dynamic undirected time varying graph, which consists of a set of vertices $V = \{0, 1, \dots, n\}$ with 0 indicting the base station and the rest indicting the mobile routers, as well as a set of edges $E(t) \subset V \times V$ indices of available communication links amongst the robots, where $E(t) = \{(i, j)\}$ consists of a pair of robots i and j . Suppose that self-loop is not valid, thus $(i, i) \notin E(t)$. We define $N_c(i) = \{j | (i, j) \in E\}$ as the set containing all neighboring robots of the i th robot in proximity communication graph.

Denote $\hat{A}(t) \in \mathfrak{R}^{n \times n}$ as the weighted adjacent matrix of the proximity communication graph $C(t)$, with normalized weight entry $\hat{A}_{ij} \rightarrow [0, 1]$ encodes the link quality of a communication link (i, j) . Suppose that $\hat{A}_{ij} > 0$ if $(i, j) \in E$, and $\hat{A}_{ij} = 0$ if $(i, j) \notin E$.

2.2 Problem Formulation

The problem we investigated now can be formulated as follows:

Problem 1 (*Mobile routing configuration control*) Considering n networked robots which is connected initially, determine the control input $u_i, \forall i \in \{1, \dots, n\}$ such that, (1) $R_u(t) \rightarrow g$ as $t \rightarrow \infty$. (2) the communication connectivity between robot R_u and the base station stay connected all the time. (3) All the robots keep collision free with obstacles in environment.

3 Modeling

3.1 Rubber Communication Model

Considering the limited communication capability, robots only can communication with the other robots located nearby. In practical application, the communication links amongst robots usually do not disappear suddenly. The radio signal strength transmitted by a wireless network device decreases as the distance increases, no signal can be reliably transmitted if the distance is far to exerts a certain value r , as illustrated in Fig. 1a. To capture this property, we associate each communication link with a rubber. A rubber may undergo three status as stretched as illustrated in Fig. 1b as follows

1. When $d \leq d_R$, the rubber is in relax state, where d denotes the length of the rubber and d_R denotes the boundary of the relax state. The rubber suffers no spring force in relax state.
2. When $d_R < d \leq d_T$, the rubber is in tension state, where d_T denotes the boundary of the tension state. The spring force in the rubber increases as the length of the rubber increases to prevent the rubber to be split.
3. When $d > d_T$, the rubber is in broken state. It will be disconnected at any time.

Associate each communication link amongst robots with a rubber, we propose a Rubber communication model to evaluate the link quality of a communication link, which is formulated as follows.

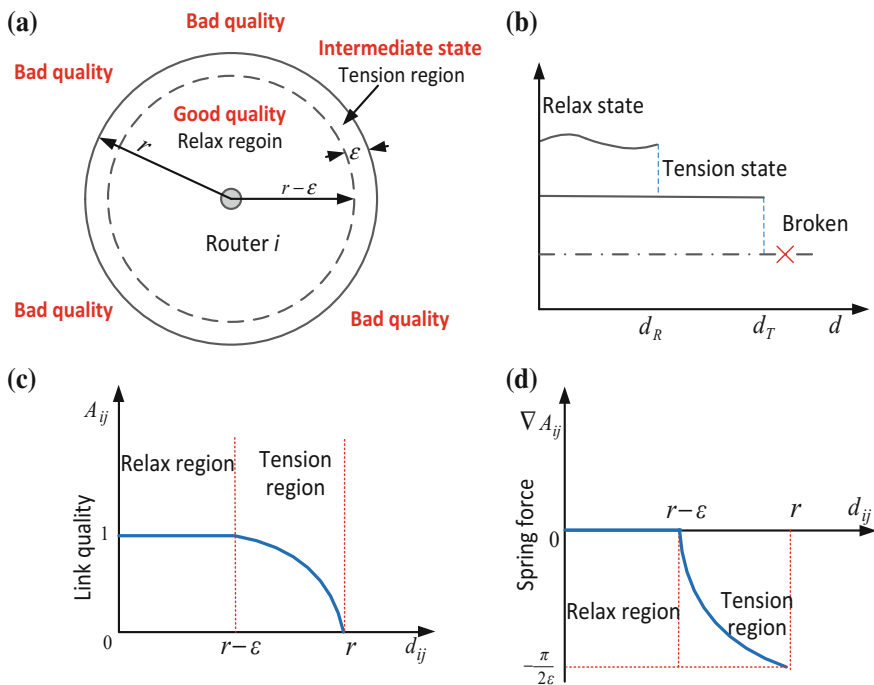


Fig. 1 Rubber tensile model

Rubber communication model Define a differentiable monotone step function $A_{ij}: [0, r] \rightarrow [0, 1]$ to evaluate the link quality of the communication link (i, j) , where $A_{ij} = 1$ means “good” quality corresponding to relax state of the rubber, $A_{ij} = 0$ means “bad” quality corresponding to the broken state of the rubber. $A_{ij} \in (0, 1)$ indicates that the link quality is in intermediate state corresponding to tension state of the rubber.

As illustrated in Fig. 1c, a candidate of the function A_{ij} can be formulated as follows

$$A_{ij} = \begin{cases} -\sin \frac{\pi(\|q_i - q_j\| - r)}{2\varepsilon}, & q_j \in S_T(i) \\ 1, & q_j \in S_R(i) \end{cases} \quad (2)$$

where $S_R(i) = \{q \mid \|q - q_i\| \leq r - \varepsilon\}$ denotes the relax region of the mobile router i , $\varepsilon \in \mathbb{R}^+$ is a positive threshold; $S_T(i) = \{q \mid r - \varepsilon < \|q - q_i\| < r\}$ denotes as the Tension region of the mobile router i , as illustrated in Fig. 1c. Note that the gradient of A_{ij} qualitatively simulates the spring force in each communication link (i, j) .

3.2 Connectivity Maintenance

We define a connectivity space for connectivity maintenance of the mobile router i and its neighboring mobile routers in the information flow graph, which is formulated as follows

Definition (*Connectivity Space $F_c(i)$*) The connectivity space for the i th mobile router is defined as a cross region expressed by $F_c(i) = \left\{ q \mid \bigcap_{j \in N_I(i)} (S_R(j) \cup S_T(j)) \right\}$. A connectivity constraint function $G_c(i)$ is given to formulate the connectivity space:

$$G_c(i) = \prod_{j \in N_I(i)} A_{ij} > 0 \tag{3}$$

To maintain the i th mobile router staying connected with all its neighboring routers in $N_I(i)$, the i th mobile router should locate inside $F_c(i)$. There is $G_c(i) > 0$ all the time.

Remark 1 When a mobile router j locates within the relax region of the i th mobile router, there is $A_{ij} = 1$. The value of $G_c(i)$ remains unchanged whether the router j is counted into the neighboring set or not. This property enables smoothly add and remove communication links, which will discussed in the follow-on development.

3.3 Obstacle Avoidance Model

As illustrated in Fig. 2a, three regions are defined around the obstacle ζ and formulated as follows

1. *Collision region*: $S_c(\zeta) = \{q \mid \|q - q_o(\zeta)\| \leq \xi\}$. Suppose that collision happens when a router locates inside $S_c(\zeta)$, where $\xi \in \mathfrak{R}^+$ is a positive threshold which satisfies $\xi > \rho(\zeta)$.

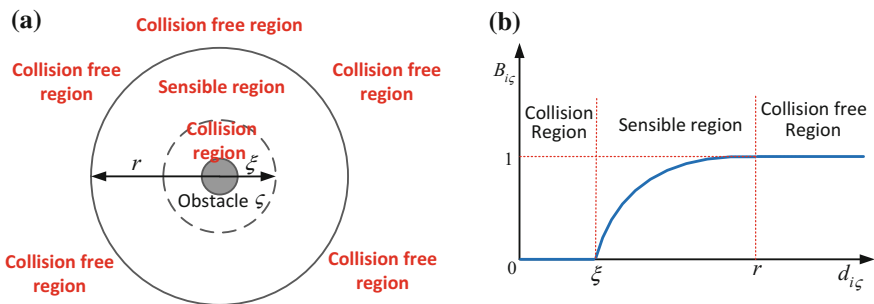


Fig. 2 Modeling for obstacle avoidance

2. *Collision free region:* $S_f(\zeta) = \{q \mid \|q - q_o(\zeta)\| > r\}$. When a router locates inside $S_f(\zeta)$, the obstacle ζ cannot be detected by the router sensor and the router keeps collision free with the obstacle ζ .
3. *Sensible region:* $S_s(\zeta) = \{q \mid \xi < \|q - q_o(\zeta)\| \leq r\}$. When a router locates inside $S_s(\zeta)$, the obstacle can be detected by the router and the obstacle ζ is taken into account for obstacle avoidance.

Obstacle avoidance model: The interaction between routers and obstacles are evaluated by a differentiable monotone step function $B_{i\zeta}: [\xi, \infty) \rightarrow [0, 1]$, where $B_{i\zeta} = 1$ implies that the mobile router i locates inside the collision free region and is collision free with the obstacle ζ , $B_{i\zeta} = 0$ implies that the mobile router i locates inside the collision region of the obstacle ζ and the collision happens, and $B_{i\zeta} \in (0, 1)$ indicates the obstacle ζ is detected by mobile router i . Thus is taken into account for obstacle avoidance.

As shown in Fig. 2b, a candidate function of B_{ij} is formulated as follows

$$B_{i\zeta} = \begin{cases} \sin \frac{\pi(d_{i\zeta} - \xi)}{2(r - \xi)}, & \xi < d_{i\zeta} \leq r \\ 1, & d_{i\zeta} > r \end{cases} \quad (4)$$

The interaction amongst routers and obstacles can be formulated by a weighted Robot-Obstacle Adjacent Matrix, denoted by $B(t) \in \mathfrak{R}^{n \times m}$. It is a $n \times m$ matrix with rows indexing robots and columns indexing obstacles, and entries by $B_{i\zeta}$. Specifically, denotes $O(i) = \{\zeta \mid 0 < B_{i\zeta} < 1\}$ as a set of obstacles detected by the mobile router i .

3.4 Collision Avoidance

We introduce a collision free space for the obstacle avoidance of the i th mobile router, expressed as follows

Definition (Collision Free Space $F_o(i)$) The collision free space for the mobile router i is defined a cross region expressed by $F_o(i) = \left\{ q \mid \left(\bigcap_{\zeta \in O(i)} (S_f(\zeta) \cup S_s(\zeta)) \right) \right\}$, which is formulated with the collision avoidance constraint function:

$$G_o(i) = \prod_{\zeta \in O(i)} B_{i\zeta} > 0 \quad (5)$$

If the mobile router i locates inside the collision free space $F_o(i)$, there is $G_o(i) > 0$ and the mobile router i keeps collision free with all the obstacles in environment.

3.5 Free Workspace

We now introduce the free workspace $F(i)$ which is defined as a subset of the robot position that meets the connectivity constraint while avoiding obstacles.

Definition (*The free workspace $F(i)$*) The free workspace for the mobile router i is defined a cross region expressed by $F(i) = \{q | F_c(i) \cap F_o(i)\}$, and formulated with a constraint function as follows

$$F(i) = \{q | G(i) = G_c(i)G_o(i) > 0\} \tag{6}$$

The above definition implies that in the free workspace, the robot maintaining connectivity with its neighbors $N_r(i)$ while keeping collision free with obstacles.

3.6 Regions Reaching Modeling

Motivated by the regional reaching concept proposed by [4], an objective function $\gamma(i) \in \mathfrak{R}$ is designed for driving the mobile routers to approach their specifically task regions, expressed by

$$\gamma(i) = \sum_{f_{\varphi} \in T(i)} \frac{1}{2} (\max[0, f_{\varphi}(q_i)])^2 \tag{7}$$

where $T(i) = [f_1(q_i), \dots, f_{\varphi}(q_i), \dots] \leq 0$ denotes the task region, which is consisted by the sphere sub-regions $f_{\varphi}(q_i) \leq 0$. Note that the task region converges to a point as the size of the task region decreases to zero.

The partial differentiating of $\gamma(i)$ with respect to q_i is as follows

$$\frac{\partial \gamma(i)}{\partial q_i} = \sum_{f_{\varphi} \in T(i)} \max[0, f_{\varphi}(q_i)] \frac{\partial f_{\varphi}(q_i)}{\partial q_i} \tag{8}$$

The control objective aims to minimize $\gamma(i)$, thus drives each mobile router i converges to the specific task region $T(i)$.

By the introducing of control objective function, the control objective function for each mobile router i can be expressed by

$$\begin{aligned} \gamma(i) = & \frac{1}{2} \left(\max \left[0, \|q_i - q_{i+}\|^2 - (r - \varepsilon)^2 \right] \right)^2 \\ & + \sum_{\varsigma \in O(i)} \frac{1}{2} \left(\max \left[0, \xi^2 - \|q_i - q_o(\varsigma)\|^2 \right] \right)^2 \end{aligned} \tag{9}$$

Remark The details of the definitions for different task regions can be find in Li et al. [7].

It should note that the control objective for the mobile user in exploring maneuver is formulated by

$$\gamma(R_u) = \frac{1}{2} \|q_{R_u} - g\|^4 \tag{10}$$

where $e = q_{R_u} - g$ denotes the exploring error.

4 Controller Design

In a similar manner to work of Li et al. [6], a decentralized potential field based function $\varphi_i \in [0, 1]$ was designed, which is expressed by

$$\varphi_i = \frac{\gamma(i)}{(\gamma^\delta(i) + G(i))^{\frac{1}{\delta}}} \tag{11}$$

where δ is a positive turning parameter with a low bound.

The control input for the mobile router i is designed as the negative gradient of φ_i , expressed as

$$u_i = -k \frac{\partial \varphi_i}{\partial q_i} \tag{12}$$

where $k \in \mathfrak{R}^+$ denotes a positive control gain.

Theorem 1 Consider n mobile routers with dynamics (1). Under the initial condition that $G(0) > 0$, the control law (12) gives rise to: (i) $G(i) > 0, \forall t > 0$; (ii) the underlying communication topology remains unchanged for all times; and (iii) all robots keeps collision free with the obstacles in environment.

Theorem 2 A positive lower bound δ' exists on parameter δ (see (11)), such that the system (1) is asymptotically stable under the controller (12).

$$\delta' \triangleq \max \left[\sqrt{2\hbar^{-1}\gamma(i)\gamma(j)\|\nabla G(i)\|\|\nabla G(j)\|}, 2\hbar^{-1}G(i)\gamma(j)\|\nabla G(j)\|\|\nabla \gamma(i)\| \right] \tag{13}$$

where $\hbar = |N_I(i)|^{-1} \|\nabla_{q_i} \varphi_i\|^2 (\gamma(i)^\delta + G(i))^{\frac{1}{\delta}+1} (\gamma(j)^\delta + G(j))^{\frac{1}{\delta}+1}$, $\nabla_{q_i}(\cdot)$ denotes the parital derivation of (\cdot) with respect to q_i .

Theorem 3 A positive lower bound δ'' exists on parameter δ such that $\mathfrak{SCT}(i)$, where $\mathfrak{S} = \{q | \nabla_{q_i} \varphi_i = 0\}$ is the largest invariant set of φ_i .

$$\delta'' \triangleq \frac{\pi\gamma(i)}{2\|\nabla_{q_i}\gamma(i)\|} \left(\frac{1}{\varepsilon} \sum_{j \in N_I(i)} \left\| \text{ctg} \frac{\pi(d_{ij} - r)}{2\varepsilon} \right\| + \frac{1}{r - \xi} \sum_{\zeta \in O(i)} \left\| \text{ctg} \frac{\pi(d_{i\zeta} - \xi)}{2(r - \xi)} \right\| \right) \quad (14)$$

The proof of Theorems 1–3 are omitted due to space limitation, interested reader can refer Li et al. [5] for details. In practical application, the turning parameter δ can be chosen as $\delta = \max\{\delta', \delta''\}$.

5 Simulations

To evaluate the effectiveness of the proposed approach, simulations were performed on a group of eight mobile routers to execute an exploring task cooperatively while maintaining the communication connectivity with a base station. As illustrated in Fig. 3a, the mobile routers are denoted by points and the obstacles are denoted by discs. The initial configuration of mobile routers and obstacles are listed in Table 1. The base station is denoted by a triangle with configuration of $q_0^T = (0, 0)$. The goal position is denoted by a cross with configuration of $g^T = (7, 2)$. The communication radius is set as $r = 1.6$ m and the positive threshold ε is set as $\varepsilon = 0.4$ m. The radius of obstacles were set as $\rho_o(1) = \dots = \rho_o(4) = 0.2$ m and the positive threshold ξ is set as $\xi = 0.3$ m.

As shown in Fig. 3, the topology of the mobile router network is designed as a hierarchic backbone chain-cluster topology and initialized as $\{\text{router } 0 \rightarrow \text{router } 2 \rightarrow \text{router } 3 \rightarrow \text{router } 4 \rightarrow \text{router } 1\}$ by using the topology design algorithm proposed by Li et al. [7]. The mobile user is selected as mobile router. The solid lines denote the communication links amongst routers. For easy distinguish, the idle routers are denoted by cycles with solid line.

Figure 4 shows the exploring maneuver of the mobile router network under controller (12) and topology design algorithm proposed by Li et al. [7]. The arrow

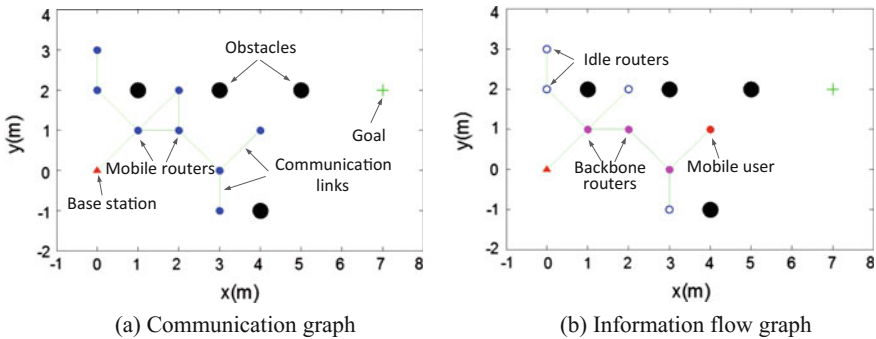


Fig. 3 The Hierarchic backbone chain-cluster topology design

Table 1 Initial Configuration

Position	Index							
	1	2	3	4	5	6	7	8
q_i (m)	(1, 4)	(1, 1)	(1, 2)	(3, 0)	(3, -1)	(2, 2)	(0, 2)	(0, 3)
$q_o(\zeta)$ (m)	(2, 5)	(2, 3)	(4, -1)	(1, 2)	NA	NA	NA	NA

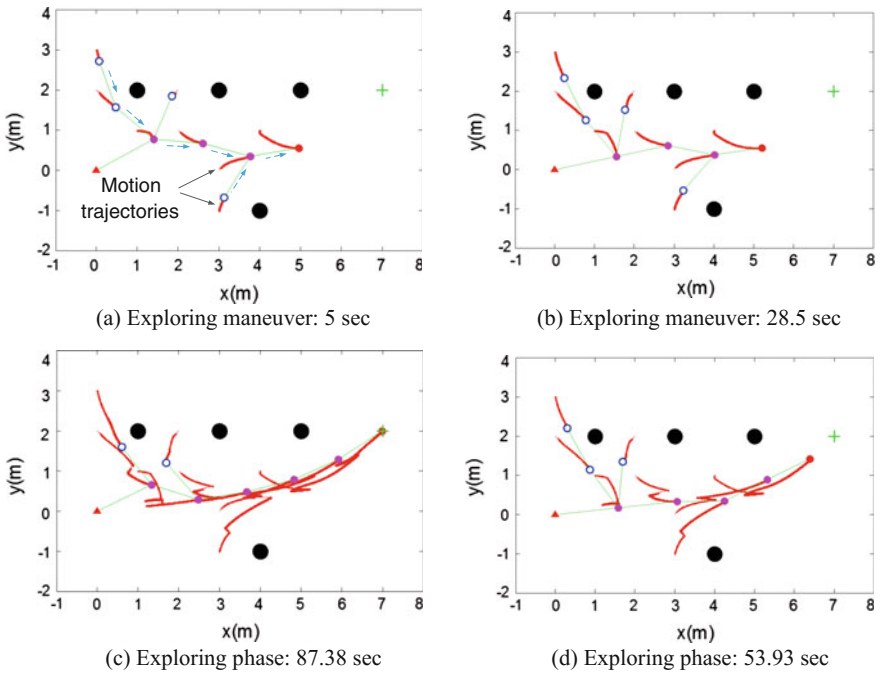


Fig. 4 Exploring maneuver

with dash line indicates the parent child relationship. The mobile user moves to the target position and the other mobile routers adjust their motions and topologies to support network connectivity between the mobile user and the base station. Figures 4a–d illustrate the motion evolution of exploring maneuver, where the mobile router approached the goal position successfully.

6 Conclusion

This paper presents a unified system, which controls the motion of mobile routers adaptively to accomplish a cooperative exploring task while maintaining the connectivity of the exploring robot and a base station and keeping collision free with all

obstacles in environment. Compare to most of the existing works, the proposed motion controller drives mobile routers to maintain the communication connectivity in the presence of obstacles, meanwhile, enables the dynamic topology switching. Both simulations and experiments were performed to demonstrate the effectiveness of the proposed approach.

Acknowledgements This work was supported in part by a grant from NSFC under Grant No. 61503270, a grant from the Natural Science Foundation of Jiangsu Province under Grant No. BK20150326, and a grant from Natural science fund for colleges and universities of Jiangsu Province under Grant No. 15KJB510029.

References

1. Bezzo, N., Fierro, R.: Tethering of mobile router networks. In: American Control Conference, 6828–6833 (2010)
2. Burdakov, O., Doherty, P., Holmberg, K., Kvarnstrom, J., Olsson, P.R.: Positioning unmanned aerial vehicles as communication relays for surveillance tasks. In: Proceedings of Robotics: Science and Systems (2009)
3. Chakraborty, N., Sycara, K. Reconfiguration algorithm for mobile robotic network. In: 2010 IEEE International Conference on Robotics and Automation, 5484–5489 (2010)
4. Cheah, C.C., Hou, S.P., Slotine, J.J.E.: Region-based shape control for a swarm of robots. *Automatica* **45**(10), 2406–2411 (2009)
5. Li, X., Sun, D., Yang, J.: Bounded controller for multirobot navigation while maintaining network connectivity in the presence of obstacles. *Automatica* **49**(1), 285–292 (2013)
6. Li, X., Sun, D., Yang, J.: Preserving multirobot connectivity in rendezvous tasks in the presence of obstacles with bounded control input. *IEEE Trans. Control Syst. Technol.* **21**(6), 2306–2314 (2013)
7. Li, X., Sun, D.: Topology design for router networks to accomplish a cooperative exploring task. In: 2014 IEEE International Conference on Robotics and Biomimetics, 884–888 (2014)
8. Mosteo, A.R., Montano, L., Lagoudakis, M.G.: Multi-robot routing under communication range. In: 2010 IEEE International Conference on Robotics and Automation, 1531–1536 (2008)
9. Stump, E., Jadbabaie, A., Kumar, V.: Connectivity management in mobile robot teams. In: Proceedings of IEEE International Conference on Robotics and Automation, 1525–1530 (2008)
10. Tekdas, O., Yang, W., Isler, V.: Robotic routers: algorithms and implementation. *Int. J. Robot. Res.* **29**(1), 110–126 (2009)
11. Yan, Y., Mostofi, Y.: Robotic router formation—a bit error rate approach. In: 2010 Military Communication Conference, 1411–1416 (2010)

The Elderly's Falling Motion Recognition Based on Kinect and Wearable Sensors

Pang Nana, Dong Min, Zhao Yue, Chen Xin and Bi Sheng

Abstract Intelligent wearable device is currently a hot spot, while the majority of smart wearable devices are concerned about the health of human information monitoring. In the study of the elderly wearable devices, the task is to find the elderly who accidentally fall, and give warning to others immediately. In order to effectively recognize the falling posture, this paper proposes a falling motion recognition algorithm combined with Kinect and wearable sensors. Although acceleration amplitude detected by three-axis accelerometer inside the wearable device can also recognize the falling motion, it is not accurate enough to distinguish the weightlessness and people's posture after falling. Therefore, Kinect's skeleton data is used to construct the feature vector of falling posture, and Support Vector Machines (SVM) is used to classify it. The experiments show that the accuracy of falling recognition is over 98 %, the real-time performance has been greatly improved as well.

Keywords Wearable device · Falling motion recognition · Elderly

1 Introduction

Nowadays, with the increase of population aging in China, the elderly's health problems have attracted more and more attention. Some intelligent health monitoring technologies [1, 2] gradually arise with the rapid development of computer technology.

P. Nana · D. Min (✉) · Z. Yue · C. Xin · B. Sheng (✉)
School of Computer Science and Engineering, South China University of Technology,
Guangzhou, China
e-mail: hollymin@scut.edu.cn

B. Sheng
e-mail: picy@scut.edu.cn

Some researches [3] are concerned with color images and gray images. Even if they put forward lots of advanced image processing algorithm, they can't avoid some noise problems. Kinect, a RGB-D camera Microsoft launched in 2010, involves color camera, infrared camera, and infrared projector and so on, so it can acquire the depth of field data of color image, that is, 3D image, which can be used for motion recognition [4].

The most important feature of Kinect is that it can perceive the depth information awareness and collect the three-dimensional data of human joints [5]. Using Kinect to recognize human behavior, we can combine the advantages of visual and non-visual research methods. It can obtain the three-dimensional human joints precisely like the wearable sensor, but without inconveniencing people's life.

The research of human behavior recognition on Kinect is mainly about human body parts distinction [6], gesture recognition [7] and head tracking, and the whole body's posture and behavior analysis [8].

2 Experimental Platform

2.1 Data Acquisition Terminal

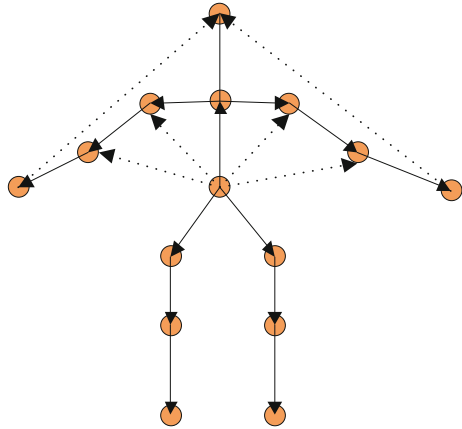
(1) Kinect collects human action feature variables

Kinect can produce color image, depth image and human joint points for upper layer developers. And through human skeletal tracking technology, the key points of human body can be identified and shown in a three-dimensional space [9].

The Kinect joint points' sets are used as an ideal human motion characteristic representation vector [10]. Because these joint points' data belong to human advanced information, each joint point's set can be directly expressed as a body [11, 12]. However, due to different human shape, like height and figure, and different coordinate of the Kinect, different people in the camera scene will produce different coordinate value and ballistic trajectory even if they perform the same action. Consequently, in order to express human action more precisely, the human feature vector based on these joints need to be further constructed.

Starting from human body structure, Sung J et al. proposed a feature representation method based on constructing human joint point vector and the angle information between vectors [11]. The method considered the relationship between human joint points, satisfying shift invariance and scaling invariance. So different human shape need not to be considered. Besides, the changing track of human body structure is nearly the same when different people complete the same action. Therefore the angle changing track among vectors can be used as the characteristics of human behavior recognition. Some improvements have been made based on the method, which extract the characteristics of human behavior (Fig. 1).

Fig. 1 Human body vector structure based on Kinect joint points



In order to identify the details in human actions, the modulus ratio of the two vectors have been added into the feature vector, and then the action feature can be identified.

The rules of naming these vectors are as follows: Their abbreviate English name is used to name the point joints. Except head (HEAD) and spine (SP), the last letter (L, C, R) of other point joints’ abbreviate name respectively represents left (L), center(C) or right(R).

The rules of naming the angle between vectors are as follows: the angle between each two vectors is named as “vector 1_ _vector 2”, for example, “SHC_SHR” and “SHR_ELR”, the angle between them can be simplified as “SHC_SHR_ELR”.

The rules of naming the modulus ratio of two vectors are as follows: Because the modulus ratio represents other parts relative to the position of the human body, the denominator of the whole modulus ratio is |SP_HEAD|. Therefore, the numerator can be directly used to represent the modulus ratio.

According to the above rules, the feature variables based on Kinect bone data are picked up as shown in Table 1.

(2) Acquire the acceleration amplitude SMV of the wearable sensor.

The sensor used in our system is INemo, which is a three- axis linear acceleration and angular inertia sensor module integrated in a miniature package released in 2011.The real-time data produced by INemo’s integrated linear accelerometer and gyroscope are used here to identify the gesture.

Table 1 The feature variables based on Kinect bone data

SHC_SHR_ELR	SHR_ELR_HANDR
SHC_SHR__SP_SHR	SHR_ELR__SP_ELR
HIPR_KNEER_FOOTR	SP_SHL__SP_HIPL
.....

According to Ref. [13], it is known that when people fall, there are two key features, weightlessness and collision. So the two features are regarded as the judgment criterion of falling. What's more, the specific feature of weightlessness is the change of acceleration, which is easily tested by the acceleration amplitude SMV [10]. We can use SMV to test falling motion.

$$SMV_t = \sqrt{a_{x,t}^2 + a_{y,t}^2 + a_{z,t}^2} \tag{1}$$

The acceleration is set at time t in a three-dimensional space (XYZ) respectively as

Usually a fall may take 0.5 ~ 1 s. We select 2/3 s. Data sampling rate is set as 50 times per second; COUNTER represents the number of times of weightlessness continuously detected when falling; therefore,

$$\begin{aligned} SampleRate &= 50 \\ Counter &= 2/3 * SampleRate \end{aligned} \tag{2}$$

If SMVCOUNTER is continuously detected out, it concludes that a fall has occurred. The Flow Chart is as following (Fig. 2).

At the same time, through Coalman filtering algorithm, our SMV curve figure after filtering is finally obtained.

From the Fig. 3, it is seen that after Coalman filtering remove noises in the three-axis accelerations data, SMV curve appears a valley, which is the SMV value when falling, and the feature is very clear. The above algorithms are implemented

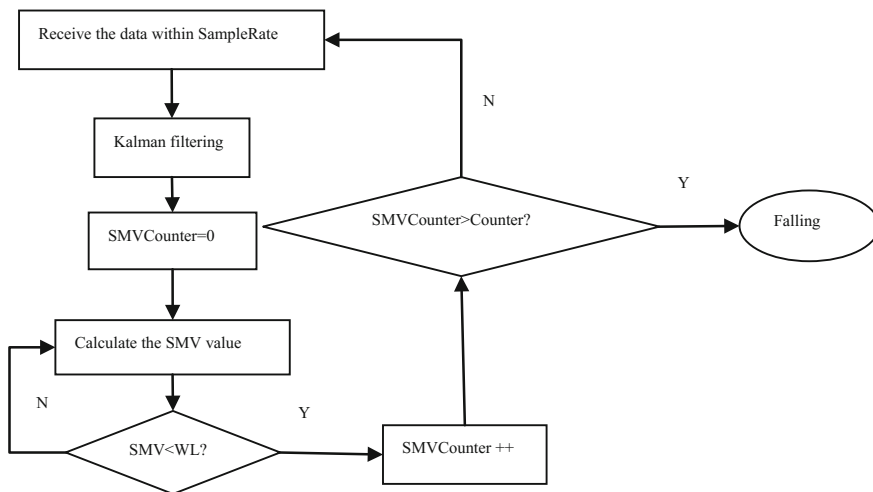


Fig. 2 Flow chart figure

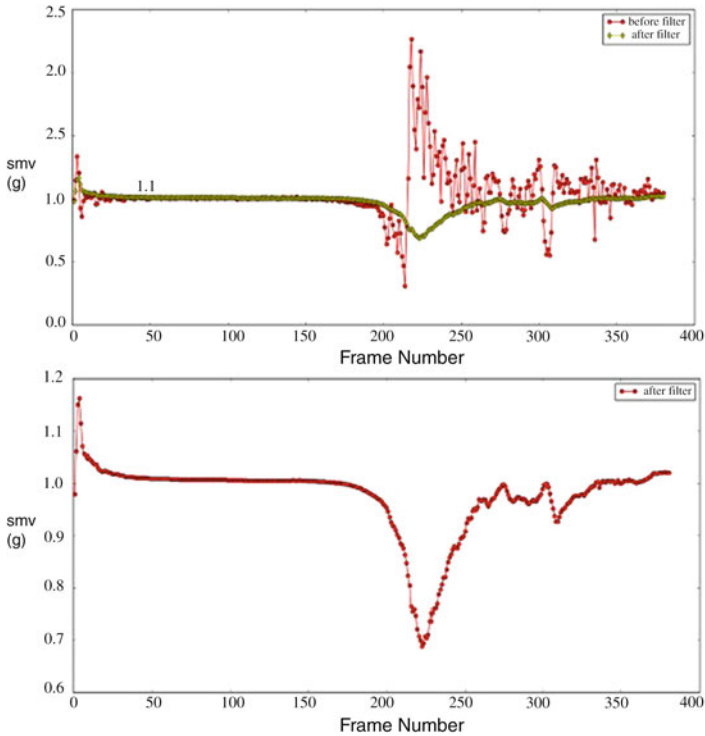


Fig. 3 SMV curve figure after filtering

on the platform; multiple experiments have been carried out many times. And we find out that the fittest WL, actually not the 0 g in theory but 0.8 g instead.

$$WL = 0.8g \tag{3}$$

Because sitting down and lying down slowly may both cause weight loss, two “false fall” actions are specially set as interference term. In this way it can test whether the algorithm in the paper can distinguish “true fall” and “false fall” or not. The experiment results are shown in Table 2.

Table 2 Falling test experiment

Experiment	Times		
	Alarming	No Alarming	Precision (%)
Sit Down/20	0	20	100
Lie Down/20	0	20	100
Fall Down/50	48	2	96

It shows that when lying down and sitting down under normal condition, it may both make SMV less than 0.8 g. However, it can't appear many continuous valleys.

2.2 Server Data Collection and Client Information Feedback

A platform is constructed as shown in Fig. 4. The elderly health monitoring platform consists of mobile phone side and browser side. Mobile phone side is used to receive the feedback information from server. When the elderly's abnormal condition is sent to the server, or the server tests out the abnormal condition, it will feedback to the corresponding mobile phone. Browser side is mainly used to obtain the elderly's health condition and geographical location over a period of time, through analyzing the information; it can also evaluate the elderly's health level.

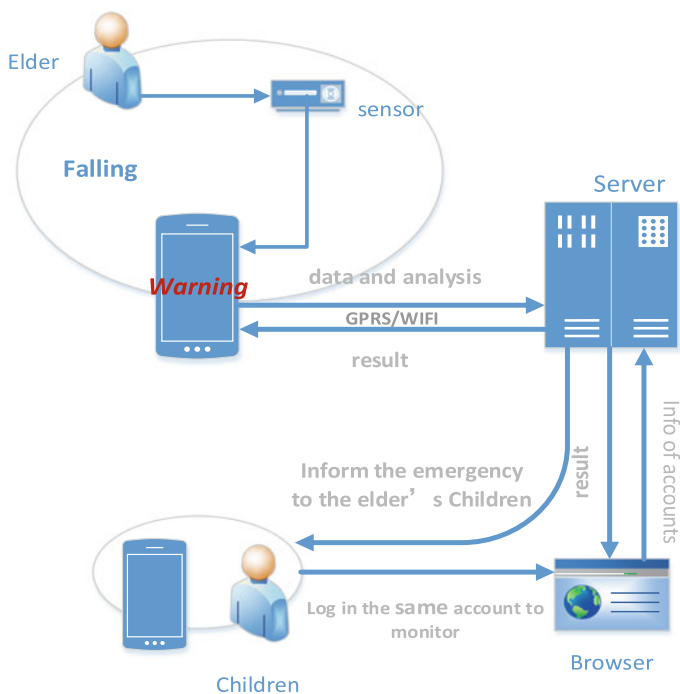


Fig. 4 Falling detection module system framework

3 Falling Detection Based on Characteristic Variables and Acceleration Amplitude

3.1 Classification of Falling Detection Using SVM of Kinect’s Feature Variables

In the fields of machine learning, Support Vector Machine (SVM) is a supervised learning model. It is usually used for pattern recognition, classification, and regression analysis. Vapnik et al., proposed a new design of linear classifier whose principle can also be divided into nonlinear. This new design makes our work easier.

SVM algorithm can be concluded as following:

(1) According to Logistic function $g(z) = \frac{1}{1+e^{-z}}$

We let $h_{\theta}(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$,

where $h_{\theta}(x)$ represents the classification eigenvalue, and therefore, when $y = 1$, we get $\theta^T x \gg 0$.

Since $\theta^T x = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n$, $\theta_0 = b$, $x_0 = 1$

Therefore $h_{\theta}(x) = g(\theta^T x) = g(W^T x + b)$

$$g(z) = \begin{cases} 1, & z \geq 0 \\ -1, & z < 0 \end{cases}$$

(2) In order to use linear classifier to identify human action, a hyper plane

$f(x) = W^T x + b$ needs to be found, which divides the data space of dimension n into two regions by finding the boundary points with maximum distance $|W^T x + b|$, and that boundary points are, namely, Support Vector x_i .

(3) W , respectively, is the normal vector of hyper plane. Therefore, we use

geometric interval $\tilde{y} = y \frac{W^T x + b}{\|W\|}$ to represent the distance as to deprive the shrinking or enlarging situation of hyper plane by different W and b combination. Since $y_i(W^T x + b) \geq 1$ and $\tilde{y} = y \frac{W^T x + b}{\|W\|} \geq \frac{1}{\|W\|}$ needs to be maximum, therefore $\|W\|^2$ is minimum.

(4) Also the outlier of the problem is taken into account, adding slack variables ξ_i , constant c represents the penalty coefficient, namely if x belongs to a certain class, but it deviates from that class to the boundary on the other place, but you do not want to give up this point, then boundary will be reduced, and less wrong equinox for the over fitting would be more serious. And the constraint conditions and objective function are as follows:

$$\begin{cases} y_i = 1, \text{ when } W^T x + b \geq 0 \\ y_i = -1, \text{ when } W^T x + b < 0 \\ y_i(W^T x + b) \geq 1 - \xi_i, \text{ where } \xi_i \geq 0, i = 1 \dots n \\ \min \frac{1}{2} \|W\|^2 + c \sum_{i=1}^n \xi_i \end{cases}$$

(5) Constructed Lagrange functions are as follows:

$$L(W, b, \alpha) = \frac{1}{2} \|W\|^2 - \sum_{i=1}^m \alpha_i [y_i(W^T x_i + b) - 1]$$

In order to calculate the minimum of $L(W, b, \alpha)$, and since α_i is stationary, then we can only calculate partial derivative of W and b .

$$\text{Let } \frac{\partial}{\partial W} L(W, b, \alpha) = W - \sum_{i=1}^m \alpha_i y_i x_i = 0,$$

$$\text{Then we get } W = \sum_{i=1}^m \alpha_i y_i x_i$$

$$\text{Let } \frac{\partial}{\partial b} L(W, b, \alpha) = \sum_{i=1}^m \alpha_i y_i = 0$$

After obtaining the upper function, the simplification process is as follows:

$$L(W, b, \alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m y_i y_j \alpha_i \alpha_j x_i^T x_j$$

(6) SMO algorithm can quickly solve the quadratic programming problem of SVM.

It is decomposed into some sub problems to solve. In each step, the SMO algorithm selects only two Lagrange multipliers for optimization, and then updates the SVM to react to the new optimized values.

Since the feature variables are obtained, SVM algorithm can be used to analyze the accuracy of Kinect’s feature variables. Firstly, the data in the paper comes from UTKinect-Action Dataset [14]. The data in UTKinect-Action Dataset is collected by a fixed Kinect. Ten people participate in the experiment, in which there are totally ten kinds of actions: walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands and clap hands. Each person does the same action twice. One of the data can be used to train and the other to test. The whole sample can be put in a text according to the data form a “libsvm” need, in order to train or test (Fig. 5).

After the data are classified and forecasted, the results of each forecast are recorded as shown in Table 3.

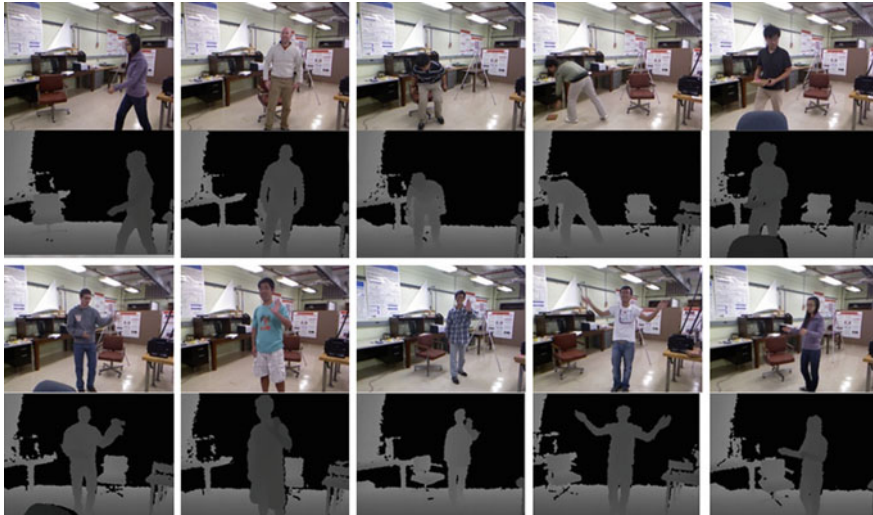


Fig. 5 UTKinect dataset

Table 3 Behavior recognition results

Real	Output					Precision (%)
	Walk	Sit down	Stand	Pick	Throw	
Walk	49	0	0	1	0	98
Sit Down	0	47	2	0	1	94
Stand	0	2	48	0	0	96
Pick	0	0	1	47	2	94
Throw	1	0	1	1	47	94

From the recognition results, it is found that many kinds of behavior's precision are more than 90 %. It concludes that our method has high recognition accuracy, and the effect is comparably remarkable.

In the characteristic variables of Kinect discussed in section II, the highly related variables associated with falling posture are SHC_SHR__SP_SHR, SHC_SHL__SP_SHL, SP_SHL__SP_HIPL and SP_SHR__SP_HIPR. While constructing the feature vector used in SVM classifier, the characteristic variables of Kinect with highest relation are combined as the feature vector X :

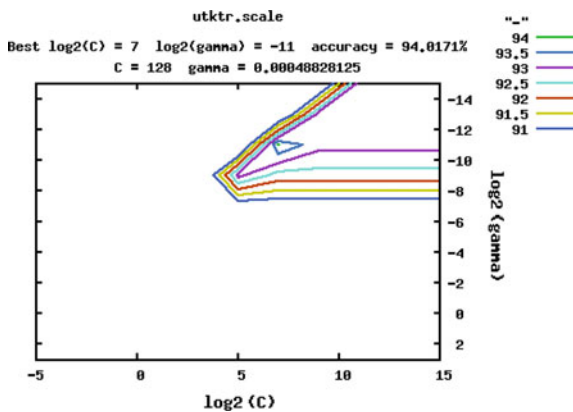
$$X = [x_1, x_2, x_3, x_4]$$

In order to use linear classifier to identify falling posture with dimension $n = 4$, a hyper plane needs to be found, which divides the data space of dimension n into two regions by finding the boundary points with maximum distance, namely, Support Vector x_i .

Table 4 Kernel test experiment

Kernel function	Linear	Binomial	RBF	Sigmoid
Accuracy (%)	95	54	95	90

Fig. 6 SVM optimal parameter



Based on SVM model properties:

- (1) It is in the situation of linearly separable analysis, but for the nonlinearly separable case, a nonlinear mapping algorithm can be used, known as kernel function, to transform a low dimensional input sample into a high dimensional feature sample, which will make high dimensional feature space using linear algorithm possible. Therefore, the sample is tested with several common kernel functions. The results are shown as follows: (Table 4)

So the most suitable one is RBF kernel function: $K(x, y) = \exp(-\gamma \|x - y\|^2)$

- (2) It is based on the theory of Structural Risk Minimization, constructing optimal hyper plane to the get global optimization. Therefore a certain upper bound is expected to meet with the risk of a certain probability. And in SVM model, which can solve the problems in small sample, high dimension, nonlinear separate and case of avoiding over-fitting, the optimal penalty coefficient C and the radius G of kernel function need to be found, as to and approach the highest classification accuracy. The experimental results are shown in Figs. 6 and 7.

3.2 Classification of Falling Detection Combined with Kinect and Wearable Sensors

As mentioned in the Sects. 2 and 3, Kinect can identify the specific behavior of a body at a time, and the sensor can determine whether the body at a specific moment

```

1 svm_type c_svc
2 kernel_type rbf
3 gamma 0.000488281
4 nr_class 5
5 total_sv 82
6 rho -1.2163 -0.82311 -0.0582902 -0.12209 -0.0427329 2.271 -0.0533935 1.25296 -0.0637498 -0.259114
7 label 1 2 3 4 5
8 nr_sv 15 19 16 19 13
9 SV
10 5.898086139531314 0 12.59743164204403 0 1:0.0676447 2:0.115455 3:0.385787 4:0.0341002 5:0.990183 6:0.583725
11 0.03725376024995226 14.42266480640365 11.4180241848908 0 1:0.10373 2:0.15876 3:0.282926 4:0.302366 5:0.84070
12 2.778347543118677 0 0 1:0.289583 2:0.0405707 3:0.215353 4:0.201255 5:0.475366 6:0.601552 7:0.348504 8:0.61
13 0 0 5.614198530873513 0 1:0.335601 2:0.0908912 3:0.280091 4:0.0741266 5:0.4786 6:0.638986 7:0.452691 8:0.372
14 32.046378533107 24.69717899679717 6.49237115963301 0 1:0.196971 2:0.12173 3:0.258672 4:0.103114 5:0.797905 6
15 6.223728808802215 0 0.7736085116245935 0 1:0.144257 2:0.0928191 3:0.158211 4:0.0365514 5:0.957441 6:0.776025
16 25.62721715844116 28.98133776838483 17.2982886795541 14.43331283085502 1:0.642841 2:0.0748973 3:0.422412 4:0

```

Fig. 7 SVM Train set

in the fall state. Both in the time state or time status of the relationship, and therefore the combination of the two will play a better role in testing.

On the other hand, the SMV algorithm presented in the Sect. 2 is according to the condition of weightlessness to determine whether it is falling. However, the body falls has another necessary condition which is that after a fall the body should be in a state of lying. Only using the acceleration data is unable to determine what the posture is after weightlessness happened, but with the use of Kinect, it is very easy to achieve. Based on this point, if the fall algorithm to join the Kinect data, you will be able to better improve the accuracy and robustness of the fall algorithm.

The details of the process are shown in Fig. 8. SMV can determine a point is in a situation of fall, when to detect falls (weightlessness), using Kinect identification

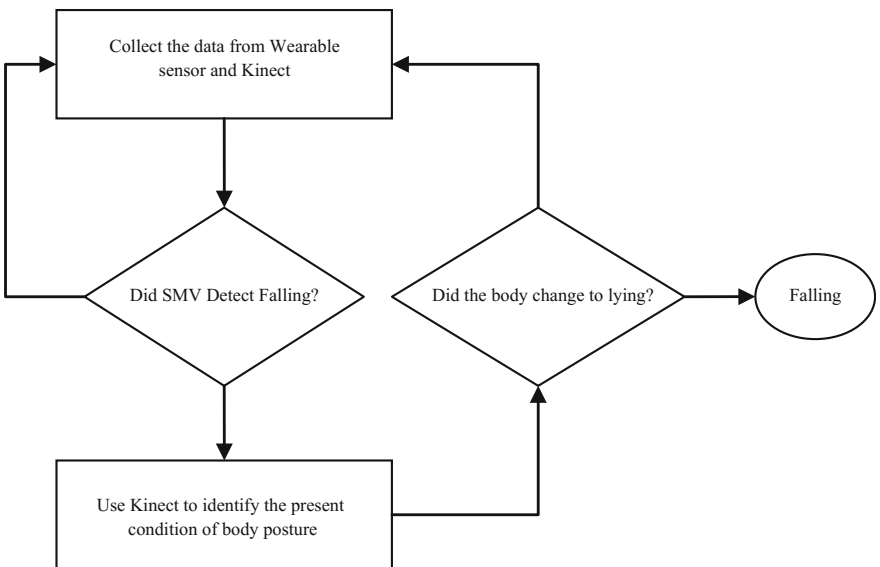


Fig. 8 Flow chart of falling detection based on Kinect and Wearable sensor

data fall before and after a period of time of human behavior, if the posture is change from standing to lying state, you can confirm to the fall, on the contrary, that is “false fall”.

4 Conclusions

This paper introduced an algorithm, based on Kinect and wearable sensor, which can identify old man falling down. Firstly, the Kinect’s skeleton data is used to construct the human body structure vector, and then the angle information between vectors is used as characteristic value of human behavior recognition. Then 4 vectors, which are highest level of falling posture are extracted, and divided into 2 situations: falling and No falling by SVM classifier. Finally, we use “libsvm” software obtain RBF trained model which can be used to test. The experimental data are collected from a “Data Set” of Kinect and sensor created by author. The experiment result verifies the effectiveness of the proposed algorithm.

In this paper, it has proved that human body behavior can be effectively recognized by using angel between human body structure vectors. When this research get further, we can realize the detection of human action as well. With developing of computer, the integration of smart device and living environment has become more touchable and convenient [8]. In future research, more wearable device will be integrated into our platform, such as surface EMG sensor [15]. With increase in the sensor, analysis of the data will be increased, so we need continue improving model data in SVM training Data set, so that computer can become more accurate in detecting. Human behavior can be understood and computer can observe correct service.

Acknowledgement This paper is a funded by Guangdong provincial science and technology plan projects(No. 2015A020219001). Guangdong Ministry of Education Foundation (No. 2013B090500093). Tianhe District science and technology project (201201YH038), Guangdong Ministry of Education Foundation (No. 2011B090400590). The National Undergraduate Innovative program.

References

1. Zhao, Y., Liu, Z., Yang, L., et al.: Combing RGB and depth map features for human activity recognition. In: Asia-pacific Signal & Information Processing Association Summit & Conference, **8345**(11), 1–4 (2012)
2. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(3), 257–267 (2001)
3. Alexiadis, D.: Evaluating a dancer’s performance using Kinect-based skeleton tracking. In: Proceedings of the 19th ACM International Conference on Multimedia, 142–152 (2011)
4. Yu, X., et al.: Children tantrum behaviour analysis based on Kinect sensor. In: Conference on Intelligent Visual Surveillance. *IEEE* 49–52 (2011)

5. Schwarz, L.A., Mkhitarian, A., Mateus, D., et al.: Human skeleton tracking from depth data using geodesic distances and optical flow. *Image Vis. Comput.* 217–226 (2012)
6. Li, Y., Ho, K.C., Popescu, M.: A microphone array system for automatic fall detection. *IEEE Trans. Biomed. Eng.* **59**(5), 1291–1301 (2012)
7. Zigel, Y., Litvak, D., Gannot, I.: A method for automatic fall detection of elderly people using floor vibrations and sound-proof of concept on human mimicking doll falls. *IEEE Trans. Biomed. Eng.* **56**(12), 2858–2867 (2009)
8. Shieh, W.Y., Huang, J.C.: Falling-incident detection and throughput enhancement in a multi-camera video-surveillance system. *Med. Eng. Phys.* **34**(7), 954–963 (2011)
9. Yang, X., Zhang, C., Tian, Y.: Recognizing actions using depth motion maps-based histograms of oriented gradients. In: *Acm International Conference on Multimedia*, 1057–1060 (2012)
10. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3D points. In: *IEEE Computer Society Conference on Computer Vision & Pattern Recognition Workshops*, 2010, Piscataway, New Jersey, USA. 9–14
11. Sung, J., Ponce, C., Selman, B. et al.: Unstructured human activity detection from RGBD images. In: *IEEE International Conference on Robotics & Automation*, **44**(8), 842–849 (2011)
12. Wang, J., Liu, Z., Yuan, J., et al.: Mining actionlet ensemble for action recognition with depth cameras. In: *Cvpr*. 1290–1297 (2012)
13. Farella, E., Pieracci, A., Benini, L., Acquaviva, A.: A wireless body area sensor network for posture detection. In: *Proceedings of 11th IEEE Symposium on Computers and Communications (ISCC'06)*, 454–459 (June 2006)
14. Xia, L., Chen, C., Aggarwal, J.K.: View invariant human action recognition using histograms of 3D joints. In: *Conference on Computer Vision and Pattern Recognition*, 20–27 (2012)
15. Yoo, J., Yan, L., Lee, S., Kim, H., Yoo, H.: A wearable ECG acquisition system with compact planar-fashionable circuit board-based shirt. *IEEE Trans. Inf. Technol. Biomed.* **13**(6), 897–902 (2009)

Online Adjusting Task Models for Ubiquitous Robotic Systems

Wenshan Wang, Qixin Cao, Qiang Qiu and Gilbert Cheruiyot

Abstract Task modeling and task planning are very important in robotic systems especially for large-scale nondeterministic problems. Two widely studied models (the classical planning model and the Markov Decision Process (MDP) model) are inapplicable to such problems due to either inherently assumed determinism or dimensional explosion. An amalgamation of these two results in a new model which is proposed in this study under the name “Reduced Markov Decision Process” (RMDP) model. This new model simplifies the conventional MDP model by reducing the branching factor of state transitions. Further, based on the RMDP model, a modified Dynamic Programming (DP) algorithm is proposed. The RMDP model also facilitates online learning that adapts the model to environmental changes. A “forgetting” model is employed for this online adjustment. In the experiment, a ubiquitous robotic system is implemented for robotic bar-tending task. The results demonstrate that the model conveniently facilitates online-updating to better match the real environment.

Keywords Task modeling · Task planning · Online learning · Ubiquitous robotics

W. Wang (✉) · Q. Cao · Q. Qiu · G. Cheruiyot
Research Institute of Robotics, Shanghai Jiao Tong University, Shanghai, China
e-mail: amigo@sjtu.edu.cn

Q. Cao
e-mail: qxcao@sjtu.edu.cn

Q. Qiu
e-mail: qiu6401@163.com

G. Cheruiyot
e-mail: gcherrie@gmail.com

1 Introduction

In recent years ubiquitous robotics has gained the attention of many researchers [1]. This technology is characterized by robotic devices dispersed in the environment such that they become a ubiquitous part of our daily lives. These distributed sensing and acting components provide services by communicating and collaborating over a network.

Such a system should have the capacity to translate user-commands to low-level actions, which can be directly executed by the distributed components. This calls for a task planning method that is capable of planning at the symbolic level for different tasks in a complex and dynamic environment.

Most of the earlier ubiquitous robotic systems are task-specific. For example: urban cleaning [2]; self-localization and mapping [3]; or public guidance [4]; etc. In these task-specific situations, the task planner is left out, because the designers tend to prefer simpler methods such as finite state machine for the action selection. However, when faced with a dynamic environment and the need to provide general-purpose services, it may be impossible to pre-consider all the situations, in which case a good task planner becomes critical.

The most commonly employed techniques are based on Artificial Intelligence (AI). Young-Guk Ha et al. used SHOP2 planner to decompose services based on semantic knowledge [5]. Robert Lundh et al. implemented a configuration approach for their network robot system also based on SHOP planner [6]. These classical planners such as SHOP2 [7] and Fast-downward [8] are efficient. However, they cannot deal with dynamic situations with uncertainties as is the case in the real world. In response to this, some researchers have used probabilistic models in task planning problems. For example, Marco Barbosa et al. used Partially Observable Markov Decision Processes (POMDP) to model the tasks with uncertainty [9]. Marcello Cirillo et al. implemented RTLplan for probabilistic domains [10]. However, planning methods based on probabilistic models suffer dimension explosion, which limits the size of the state space to impractical applications. Furthermore, the real environment is not only nondeterministic but also dynamic. The stationary model would soon be inaccurate for the nonstationary environment, and need a revision.

In view of the foregoing, we propose in this paper a Reduced MDP (RMDP) model, which is arrived at by introducing further assumptions to the MDP model. It is discussed in detail how this model can accelerate the planning process without compromising its expressivity on nondeterministic problems for real environments. Furthermore, using the RMDP model, we propose an online learning algorithm, which allows the planning model to adapt to the system changes. Consequently, the planning results could be improved as well. A ubiquitous robotic system for bar-tending scenario is implemented as a demonstration platform for these algorithms.

This paper starts with a brief introduction to our ubiquitous robotic system and its component-based structure in Sect. 2. Subsequently, in Sect. 3, the RMDP

model is proposed. The online learning capability that makes this method adaptive to dynamic situations is presented in Sect. 4 after which the results of the experiments are discussed in Sect. 5. Finally, conclusions are drawn in Sect. 6.

2 System Overview

In contrast to the monolithic robot, the ubiquitous robotic system offers the advantage of distributed robotic sensing and acting devices in the environment to complete different tasks through collaboration [11]. This paper focuses on how to cooperate different robotic devices to complete complex tasks.

The symbolic task planning module consists of a device manager, a task planner and an online learning module as Fig. 1 shows. The device manager is responsible for transferring high-level tasks and monitoring low-level status. The planner turns users' abstract commands into sub-task sequences, which can be directly carried out by corresponding robotic components. The online learning module adjusts the domain knowledge used by the task planner in real time, so that the planner could adapt to the environmental changes.

The robotic devices in the system are developed based on middleware technology [12, 13]. This component-based structure is the foundation of the system. It facilitates the execution of the planning results to fulfill realistic tasks in physical environment. Robotic devices are highly heterogeneous with respect to platforms such as operating system, programming language and communication media. Middleware is thus employed to generalize the components into a uniform abstraction which enables dynamic communication and coordination between any two of the modules. Figure 2 explains how the middleware "transparentizes" the

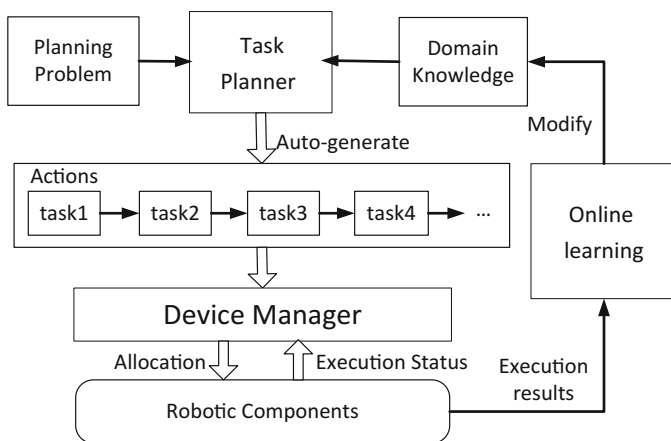


Fig. 1 System diagram for task planner and online learning

hardware and software platforms, and offers standardized access to the robotic devices.

We implement our system based on (Robotic Technology) RT middleware [14, 15]. In RT component, the ports are categorized into data ports and service ports. The data port is responsible for the continuous exchange of data. Each component can have any number of data in-ports and out-ports. A data out-port sends the data to a corresponding in-port which receives the data. The service port provides the command based communication. The component with a service port, offering a set of services, listens for requests for those services via a connector.

We further define three kinds of service ports, namely FuncGet, FuncSet and ExeStatusGet as Fig. 2 shows. The service port is responsible for the interaction with the task planner and the learning module. FuncGet port reports to the upper layer about the components' state. For example, the environmental camera provides the index of current robots within the field of vision; the mobile platform feeds back its coordinates, etc. FuncSet port provides the functionality invoking, such as setting the target position for the mobile table, setting the localizing target for the environmental camera, etc. ExeStatusGet port returns the execution status, for example whether or not the mobile table has reached its destination, or whether the environmental camera is locking onto its target.

Each component may have any number of data ports for continuous data exchange between components. For instance, the localization information is transferred from the data out-port of environmental camera to the data in-port of the path planning component. Once two data ports are connected, those two components are able to perform real-time communication to accomplish the task collaboratively.

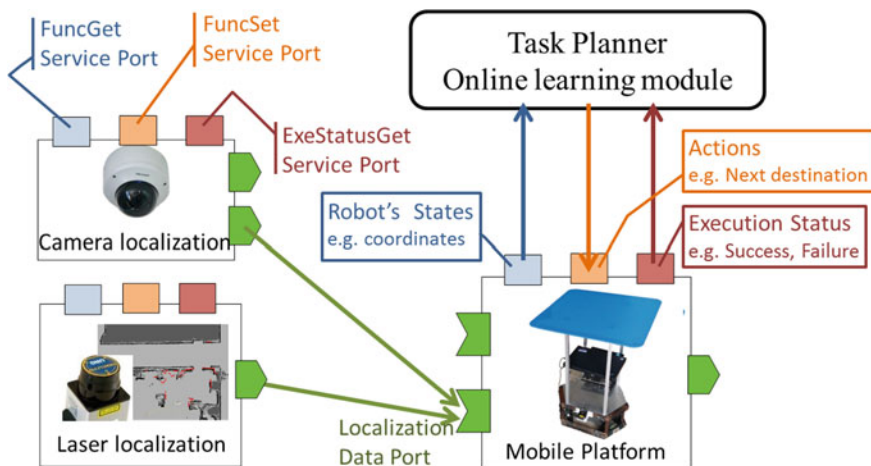


Fig. 2 The distributed components are in the uniformed structure using middleware technology

3 Task Modeling and Task Planning

As mentioned above, the different robotic functionalities are abstracted, and ready to be called through the service ports and data ports. As a result, the tasks in the service layer, can be modeled, analyzed and solved at symbolic level. This paper follows the techniques of automated planning derived from the AI field. The Task planning problem is modeled as a state transition system. Depending on different assumptions, various models are proposed. The two most commonly used models are the classical planning model [16–18] and the MDP model [19].

The classical planning model is not particularly well-suited to practical problems due to its strict assumptions. The MDP model has good expressivity but is less suitable for this study due to the large state space of ubiquitous robotic problems. To achieve better efficiency without sacrificing the expressivity on realistic problems, we propose the RMDP model by reducing the branching factor of the state transitions. The reason for the difficulty of solving MDP model comes from its large branching factor. It is assumed in RMDP model that after actions are executed by robotic components, the outcome could be a few predictable states, which are the successful state and a few failed states. Take grasping action for instance, if the manipulator fails to pick up one object, the object will either remain where it is or drop on the table or floor. This assumption dramatically decreases the branching factor of the state space. As shown in Fig. 3, in MDP model, the branching factor of state transition is as large as the state number $|S|$. While in RMDP model, the branching factor is much smaller. Further, the RMDP employs multi-valued state variables instead of binary variables commonly applied in the existing automated planning domain languages [20, 21].

Definition 1 RMDP model is defined as a five-tuple $\Pi = (V, D, A, I, G)$:

- $V = \{v_1, v_2, \dots\}$ is a finite set of state variables;
- $D = \{d_1, d_2, \dots, d_n\}$ is a finite set of variable domains, each $v_i \in V$ with a finite domain $d_i \in D$. V and D define the planning space S , where state $s \in S$ is represented as a vector $[x_1, x_2, \dots, x_n]$, where $x_i \in d_i$ is the value of variable v_i ;

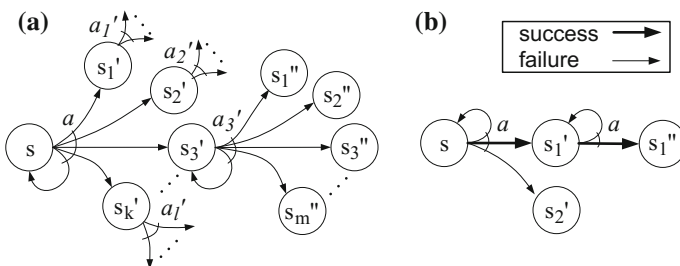


Fig. 3 a MDP model for state transition, b RMDP model for state transition

- $A = \{a_1, a_2, \dots, a_m\}$ is a finite set of actions, each $a_i \in A$ is a triple (Pc, Ef, c) referred to the action's preconditions, effects and cost respectively.
- $I \in S$ denotes the initial state;
- $G \subseteq S$ denotes the set of goal states.

Definition 2 A partial state ξ is a value assignment of a subset of state variables $V_\xi \subseteq V$. Assuming $V_\xi = \{v_{\xi 1}, v_{\xi 2}, \dots, v_{\xi k}\}$, ξ is represented with a set of variable-value pair $\{(v_{\xi 1}, x_{\xi 1}), (v_{\xi 2}, x_{\xi 2}), \dots, (v_{\xi k}, x_{\xi k})\}$.

Definition 3 A partial assignment is a function $\Gamma: S \times \xi \mapsto S$ represents the state transition on a partial set of state variables $V_\xi = \{v_{\xi 1}, v_{\xi 2}, \dots, v_{\xi k}\}$. Assuming $\Gamma(s, \xi) = s'$ and $(v_{\xi i}, x_{\xi i}) \in \xi$, s' is obtained by setting each variable $v_{\xi i}$ to $x_{\xi i}$ in state s .

The precondition $Pc(a)$ of action a is defined with a partial state $\xi^{Pc(a)}$, where $(v_{\xi i}, x_{\xi i}) \in \xi^{Pc(a)}$ denotes the value of variable $v_{\xi i}$ should be $x_{\xi i}$ to satisfy the precondition. The effects $Ef(a)$ of action a is defined with an effect list $[ef_1, ef_2, \dots, ef_k]$, where $ef_i = (p_i, \xi_i^{Ef(a)})$ denotes the outcome state will be the partial assignment $\Gamma(s, \xi_i^{Ef(a)})$ with probability p_i , after the action's execution. Each action has a cost c , which acts like a reward function in MDP model.

The planning result of RMDP is also a policy $\pi: S \mapsto A$. The expected value $VF^\pi(s)$ of a policy π at a given state s satisfies:

$$\begin{aligned}
 VF^\pi(s) &= E\left\{ \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s \right\} \\
 &= c(\pi(s_0)) + \gamma c(\pi(s_1)) + \gamma^2 c(\pi(s_2)) + \dots \\
 &= c(\pi(s)) + \gamma \sum_{(p_i, \xi_i) \in Ef(\pi(s))} p_i \cdot VF^\pi(\Gamma(s, \xi_i^{Ef(\pi(s))})),
 \end{aligned} \tag{1}$$

where $c(a)$ represents the cost of action a ; γ is the discount factor; $Ef(a)$ represents the effects of action a . Similar to the MDP model, the optimal value function satisfies:

$$VF^*(s) = \min_{a \in A} [c(a) + \gamma \sum_{(p_i, \xi_i) \in Ef(a)} p_i \cdot VF^*(\Gamma(s, \xi_i^{Ef(a)}))] \tag{2}$$

The optimal policy π^* is the one that minimizes the value function:

$$\pi^*(s) = \arg \min_{a \in A} [c(a) + \gamma \sum_{(p_i, \xi_i) \in Ef(a)} p_i \cdot VF^*(\Gamma(s, \xi_i^{Ef(a)}))] \tag{3}$$

With RMDP model, we derive the following modified Dynamic Programming (DP) algorithm according to Eqs. (2) and (3):

Algorithm 1. Modified DP

-
1. initialize value function $VF(s) = 0$ for all $s \in S$
 2. repeat for each episode:
 3. initialize s as the initial state
 4. repeat for each step of episode:
 5. for each a that is applicable to s :
 6. $(s_i, \xi_i) \leftarrow Ef(s, a)$
 7. $a \leftarrow \arg \min_{a \in A} [c(a) + \gamma \sum p_i \cdot VF(\Gamma(s, \xi_i))]$
 8. $VF(s) \leftarrow (1 - \alpha)V(s) + \alpha[c(a) + \gamma \sum p_i \cdot VF(\Gamma(s, \xi_i))]$
 9. $s \leftarrow \text{execute}(s, a)$
 10. until s is the goal state
 11. until VF converges
 12. $\pi(s) \leftarrow \arg \min_{a \in A} [c(a) + \gamma \sum_{(p_i, \xi_i) \in Ef(a)} p_i \cdot VF(\Gamma(s, \xi_i))]$
-

This algorithm modifies the value function iteratively as depicted in line 7 and line 8, where $\alpha \in (0, 1]$ is the step size, which controls the iteration speed. The value function is guaranteed to converge to VF^* given a sufficiently large number of iterations.

4 Online Model Learning

The automated planning methods in AI only consider the offline problems. However for practical problems, the model should adapt to dynamic and nonstationary environments. For example, changing the position of furniture would alter the moving ability of the mobile robots. Consequently, as new information gained, the model should be improved online to make it more accurately match the real environment. This means the planner will gradually compute a new way of behaving to match the new model.

The RMDP model describes the environmental uncertainty with the probabilistic effects of actions. This naturally allows a method for online learning of the transition probabilities. Recall that the probabilistic effects $Ef(a)$ is defined with an effect list $[e_1, e_2, \dots, e_k]$, where $e_i = (p_i, \xi_i)$ denotes the state will be changed by partial assignment $s'_i = \Gamma(s, \xi_i)$ with probability p_i , after the executing action a . Intuitively, the transition probability p_i can be estimated by the number of times that it takes action a in state s to get to the state s' , divided by the total number of times that take action a in state s :

$$p_i = \frac{\#[(s, a) \mapsto s'_i]}{\#[(s, a)]} \quad (4)$$

Following each execution instance of action a , the execution result is recorded by the online learning module. This module calculates the transition probabilities as more execution results are recorded according to Eq. (4). However, since this method factors in each instance with equal weight, there could be great latency if the environment suddenly changes. A solution is to make the system more “forgetful”, and give higher credit to recent experiences.

Suppose the action a has been executed n times since the system was started. Further supposing these executions happened on time steps $t_1^a, t_2^a, \dots, t_n^a$ and assuming the execution results are $r_1^a, r_2^a, \dots, r_n^a$, where each $r_i^a \in \{1, 2, \dots, |Ef(a)|\}$. Define a binary function as follows:

$$t^a(k, i) = \begin{cases} 0 & \text{if } r_i^a = k \\ 1 & \text{elsewise} \end{cases} \quad (5)$$

An exponential forgetting model is defined as:

$$p_k^a(n) = \frac{\sum_{i=1}^n t^a(k, i) e^{-\alpha(t_n^a - t_i^a)}}{\sum_{i=1}^n e^{-\alpha(t_n^a - t_i^a)}}, \quad (6)$$

where $\alpha \in \mathbb{R}^+$ is the forgetting factor. The larger the value of α , the more the system is biased towards the recent experiences. α is set to 0.1 in the experiment described in the following section.

An iterative version which is easier to implement is as follows:

$$p_k^a(n) = p_k^a(n-1) + \frac{1}{\varphi(n)} (t^a(k, n) - p_k^a(n-1)) \quad (7)$$

$$\text{where } \varphi(n) = e^{-\alpha(t_n^a - t_{n-1}^a)} \varphi(n-1) + 1 \quad (8)$$

5 Experiment Results

5.1 The Robot Bar-Tending Experiment

A bar-tending scenario was implemented as ubiquitous robotic system based on the middleware technology discussed in Sect. 2. The task was to serve drinks to customers. Different robotic devices were developed into components as shown in Fig. 4. There were two mobile tables with different color and size, one dual-arm

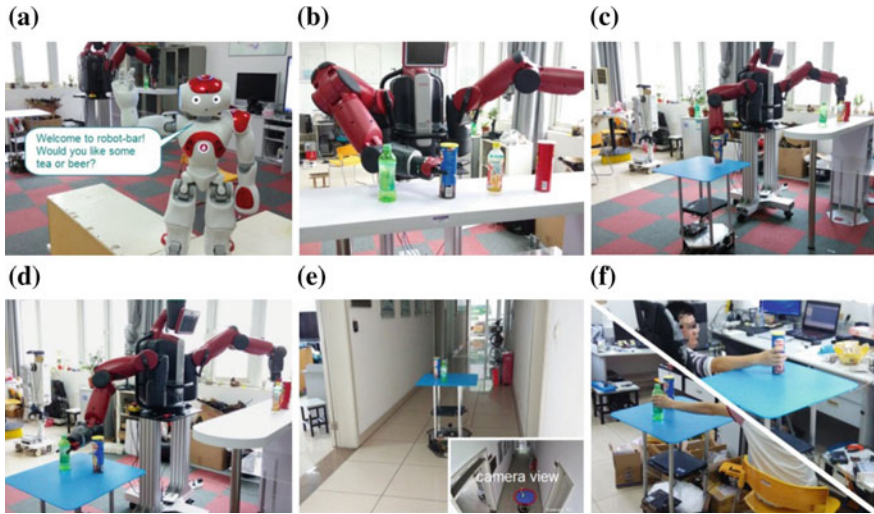


Fig. 4 Execution process of one bar-tending task

robot for grasping the drinks and snacks on the bar counter, three environmental cameras to help the localization, two laser sensors mounted on the mobile tables for localization and obstacles avoiding, and other software components such as path planning, object recognition, etc.

This task environment comprised two rooms connected by a corridor. One humanoid robot standing by the entrance took the orders from patrons. The tasks were generated according to the customers' orders.

The states and actions are predefined as domain knowledge. The initial state is reported by each component through the FuncGet service port. The goal state is translated from the customer's order. When all these information is ready, the planner calculates the policy using modified DP algorithm as detailed in Sect. 3.

The actions were then sent to each component for execution. Figure 4 illustrates the execution process of one such task. First, the mobile table moved to the bar counter through laser localization. Next the dual-arm robot picked up the drink and snack ordered by customers, and placed them on the mobile table as Fig. 4b–d show. The mobile table then moved to the other room where two customers picked their food and drink while at their tables. As the mobile table traveled through the corridor, the laser failed to provide the localization information, so the environmental camera mounted on the ceiling started to localize the robot as Fig. 4e. The execution status of each action was monitored by querying on the ExeStatusGet service port. Once any error or failure occurred, a new action was executed according to the policy without re-planning the whole task.

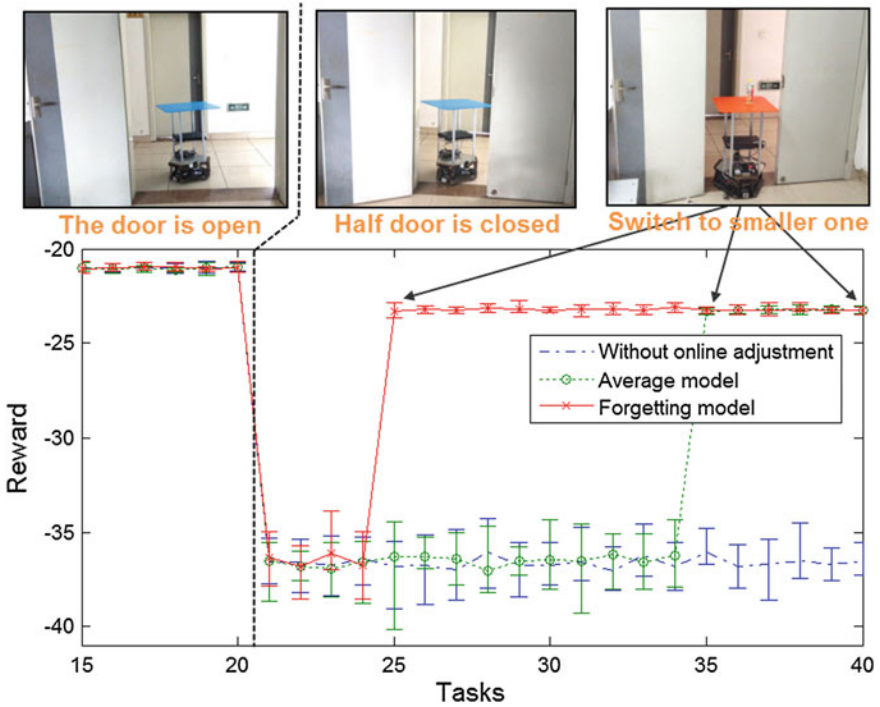


Fig. 5 Online adjusting the model when the environment changes

5.2 Online Adapting to Environment Changes

The online learning method was tested by the following scenario: The objective is to have the drinks delivered to the next room. Initially the doorway was fully open, hence the mobile tables had no difficulty passing through it. Sometime later half of the doorway was closed, so that the larger mobile robot often failed to pass through.

The same task was repeated 40 times. In the first 20 times, the door was open. The bigger robot—which is initially located closer to the bar counter—was sent to deliver the drinks. From the 21st episode onwards, the doorway was half closed. As Fig. 5 shows, a sharp drop in the reward was registered as a result of execution failure. Three methods were compared. Namely: using the average model; forgetting model; and the one without online adjustment. The two methods with online learning altered the success rate of the corresponding actions. Consequently, they switched to another policy that employed the smaller sized mobile table to do the task. Conversely, the method without online learning still stuck to the poor policy. It may be readily noted that the one with forgetting model had a quicker response to the environmental changes when compared to the one based on the average model. In fact this effect is more pronounced when more trials are made before the environment changes.

6 Conclusion

Given the increasing popularity of the ubiquitous robotic system as a solution offering better environmental intelligence, the RMDP model proposed in this work has been proved to be suitable for the symbolic planning problems that are especially large-scale and with uncertainties.

This result has been achieved by retaining the probabilistic features of the MDP model while at the same time decreasing the branching factor of the state transitions. A robotic bar-tending task was implemented based on RMDP model and was solved using a modified DP algorithm. The experiments demonstrated that the RMDP model facilitates online learning, which adapts the model to environmental changes. A “forgetting” model is employed for this online learning process. The results showed that the improved model more accurately matched the real environment and the planning results also improved accordingly.

Acknowledgements The authors gratefully acknowledge YASKAWA Electric Corporation for supporting the collaborative research funds under the project of “Research and Development of Key Technologies for Smart Factory”.

References

1. Chibani, A., Amirat, Y., Mohammed, S., Matson, E., Hagita, N., Barreto, M.: Ubiquitous robotics: recent challenges and future trends. *Robot. Auton. Syst.* **61**, 1162–1172 (2013)
2. Reggente, M., Mondini, A., Ferri, G., Mazzolai, B., Manzi, A., Gabelletti, M., Dario, P., Lilienthal, A.J.: The dustbot system: using mobile robots to monitor pollution in pedestrian area. *Proc. NOSE* **23**, 273–278 (2010)
3. Kim, B.K., Tomokuni, N., Ohara, K., Tanikawa, T., Ohba, K., Hirai, S.: Ubiquitous localization and mapping for robots with ambient intelligence. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4809–4814. IEEE
4. Koide, Y., Kanda, T., Sumi, Y., Kogure, K., Ishiguro, H.: An approach to integrating an interactive guide robot with ubiquitous sensors. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, pp. 2500–2505. IEEE
5. Ha, Y.G., Sohn, J.C., Cho, Y.J., Yoon, H.: A robotic service framework supporting automated integration of ubiquitous sensors and devices. *Inf. Sci.* **177**, 657–679 (2007)
6. Lundh, R., Karlsson, L., Saffiotti, A.: Autonomous functional configuration of a network robot system. *Robot. Auton. Syst.* **56**, 819–830 (2008)
7. Nau, D.S., Au, T.C., Ilghami, O., Kuter, U., Murdock, J.W., Wu, D., Yaman, F.: SHOP2: an HTN planning system. *J. Artif. Intell. Res. (JAIR)* **20**, 379–404 (2003)
8. Helmert, M.: The fast downward planning system. *J. Artif. Intell. Res. (JAIR)* **26**, 191–246 (2006)
9. Barbosa, M., Bernardino, A., Figueira, D., Gaspar, J., Gonçalves, N., Lima, P.U., Moreno, P., Pahlhiani, A., Santos-Victor, J., Spaan, M.T.J.: ISRobotNet: a testbed for sensor and robot network systems. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2827–2833. IEEE
10. Cirillo, M., Karlsson, L., Saffiotti, A.: A human-aware robot task planner. In: *Proceedings of the 19th International Conference on Automated Planning and Scheduling*, pp. 58–65. AAAI Press (2009)

11. Wang, W., Cao, Q., Zhu, X., Liang, S.: A framework for intelligent service environments based on middleware and general purpose task planner. In: 2015 International Conference on Intelligent Environments (IE), pp. 184–187. IEEE
12. Zhai, M., McKenna, G.B.: Surface energy of a polyurethane as a function of film thickness. In: Proceedings of the Annual Technical Conference, Society of Plastics Engineers, ANTEC, Las Vegas (2014)
13. Wang, W., Cao, Q., Zhu, X., Adachi, M.: An automatic switching approach of robotic components for improving robot localization reliability in complicated environment. *Ind. Robot: Int. J.* **41**(2), 135–144 (2014)
14. Zhai, M., McKenna, G.B.: Elastic modulus and surface tension of a polyurethane rubber in nanometer thick films. *Polymer* **55**(11), 2725–2733 (2014)
15. Ando, N., Suehiro, T., Kitagaki, K., Kotoku, T., Yoon, W.-K.: RT-middleware: distributed component middleware for RT (robot technology). In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp. 3933–3938. IEEE
16. Blum, A.L., Furst, M.L.: Fast planning through planning graph analysis. *Artif. Intell.* **90**, 281–300 (1997)
17. Kautz, H.A., Selman, B.: Planning as satisfiability. In: Proceedings of the 10th European Conference on Artificial Intelligence, pp. 359–363. Wiley (1992)
18. Bonet, B., Geffner, H.: Planning as heuristic search. *Artif. Intell.* **129**, 5–33 (2001)
19. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
20. Sanner, S.: Relational dynamic influence diagram language (RDDI): language description, Unpublished MS. Australian National University (2010)
21. Fox, M., Long, D.: PDDL. 1: an extension to PDDL for expressing temporal planning domains. *J. Artif. Intell. Res. (JAIR)*, **20**, 61–124 (2003)

Improved Skeleton Estimation by Means of Depth Data Fusion from Multiple Depth Cameras

Marco Carraro, Matteo Munaro, Alina Roitberg
and Emanuele Menegatti

Abstract In this work, we address the problem of human skeleton estimation when multiple depth cameras are available. We propose a system that takes advantage of the knowledge of the camera poses to create a collaborative virtual depth image of the person in the scene which consists of points from all the cameras and that represents the person in a frontal pose. This depth image is fed as input to the open-source body part detector in the *Point Cloud Library*. A further contribution of this work is the improvement of this detector obtained by introducing two new components: as a pre-processing, a people detector is applied to remove the background from the depth map before estimating the skeleton, while an alpha-beta tracking is added as a post-processing step for filtering the obtained joint positions over time. The overall system has been proven to effectively improve the skeleton estimation on two sequences of people in different poses acquired from two first-generation Microsoft Kinect.

Keywords Skeleton estimation • Body parts estimation • Multiple depth cameras • PCL • OpenPTrack

M. Carraro (✉) • M. Munaro (✉) • E. Menegatti (✉)
Department of Information Engineering, University of Padova,
Via Gradenigo 6/A, 35131 Padua, Italy
e-mail: marco.carraro@dei.unipd.it; carraro1@dei.unipd.it

M. Munaro
e-mail: matteo.munaro@dei.unipd.it

E. Menegatti
e-mail: emg@dei.unipd.it

A. Roitberg
Technische Universität München, Boltzmannstraße 3,
85748 Garching bei München, Germany
e-mail: roitberg@in.tum.de

1 Introduction

The capability to segment the body parts or, more generally, to estimate a skeleton of a person in an unsupervised way is fundamental for many applications: from health-care to ambient assisted living, from surveillance to action-recognition and people re-identification. The introduction of affordable RGB-D cameras as the Microsoft Kinect, has given a boost to the research in this area and many marker-less skeleton estimation algorithms were born, as Shotton's human pose recognition [1] and Buys' body pose detection [2]. However, all these systems perform better when the subject is seen frontally with respect to the depth camera, mainly because most of the training examples they were trained with referred to this pose. In this work, we want to overcome this problem when multiple cameras are available, thus generating a virtual depth image of the subject warped in frontal view after having fused the depth information coming from the cameras. Moreover, we propose an improvement to Buys' body pose detector [2], here used for skeleton estimation, by adding a preliminary people detection phase for background removal and performing an alpha-beta tracking on the final skeleton joints. The system has been tested on sequences of two freely moving persons imaged by a network composed of two first-generation Microsoft Kinect sensors. Summarizing, the contribution of this work is two-fold:

- We introduce a novel multi-view method to estimate the skeleton of a person based on the fusion of the 3D information coming from all the sensors in the network and a subsequent warping to a frontal pose;
- We improve the body pose detector in [2] by removing background points from the input depth image with a people detection phase and by adding a joint tracking filter to the output of the detector.

The remainder of the paper is organized as follows: Sect. 2, reviews the state-of-the-art of both single-camera and multi-camera skeleton tracking algorithms, while Sect. 3 gives an overview of our system. In Sect. 4, we describe the multi-view data fusion part of our system, while in Sect. 5 we describe the skeleton estimation algorithm we used and how we improved it. Finally, Sect. 6 details the experiments we performed and the results we achieved and in Sect. 7 conclusions are drawn.

2 Related Work

The skeleton of a person gives important cues on what the person is doing (action-recognition) [3, 4], who is the person viewed (people re-identification) [5–7], what are her intentions (surveillance) [8] and how are her health conditions (health-care) [9]. Furthermore, the wide literature on people tracking [10–12] demonstrates its usefulness for both security applications and human-robot interaction. One of the most important works on skeleton tracking is the one by Shotton et al. [1], which trains a random forest to recognize the body parts of a person with a huge train-

ing dataset composed of real and synthetic depth images of people. The classifier, licensed by Microsoft for entertainment applications, achieves good performance and works in real-time. The system is released within the Microsoft Kinect SDK and only works with Windows-based computers. Another work released as open-source within the *Point Cloud Library* [13] is the work of Buys et al. [2], which uses an approach similar to Shotton's. In our work, we use this latter body-part detector, that we also improved by adding a people detection pre-processing phase and an alpha-beta tracking algorithm.

Intelligent surveillance systems rely more and more on camera network cooperation. Indeed, more cameras are able to cover more space and from multiple views, obtaining better 3D shapes of the subject and decreasing the probability of occlusions. Recent works relies on camera collaboration in network to enhance skeletal estimation. In [14], the skeleton obtained by single RGB images is fused with the skeleton estimated from a 3D model composed with the visual hull technique. The visual hull is used for refining the pose obtained from the single images. In [15], a skeleton is computed for every camera from a single image and then these estimated are projected to 3D and intersected in space. The work by Gao et al. [16] addresses this problem by registering a 3D model to the scanned point cloud obtained by two Kinects. This work is very accurate but unfeasible for real-time purposes given the 6 seconds needed to process each frame. The work of Yeung et al. [17] proposes a solution to the same problem with two Kinects that can be used in real-time. In particular, they uses two orthogonal Kinects and fuse the skeletons obtained from the Microsoft SDK with a constrained optimized framework.

In this work we exploit the multi-view information at the depth level, leaving the skeleton estimation as the last part of the pipeline. In this way, we are able to obtain better skeletons also when the single ones are potentially noisy or when they have some not tracked joints. Moreover, we minimize the skeleton estimation error by warping the fused data to a frontal view, given that the skeleton estimation is best performed from frontally viewed persons.

3 System Overview

Figure 1 provides an overview of our system. In this work, a network composed of two first-generation Microsoft Kinect is considered, but the extension to a higher number of cameras is straightforward. At each new frame, the Kinects compute the 3D point cloud of the scene and the people detector segments only the points belonging to the persons in the scene. Afterwards, we transform the point clouds to a common reference frame given that the network is calibrated and we fuse the point cloud data after performing a fine registration with the Iterative Closest Point (ICP) algorithm [18]. The multi-view cloud obtained is then rotated and reprojected to a virtual image plane so as to generate a depth map of the persons seen from a frontal view. Then, body parts detection is performed on this virtual depth map and the joint position is computed from the body segmentation and tracked with an alpha-beta tracking

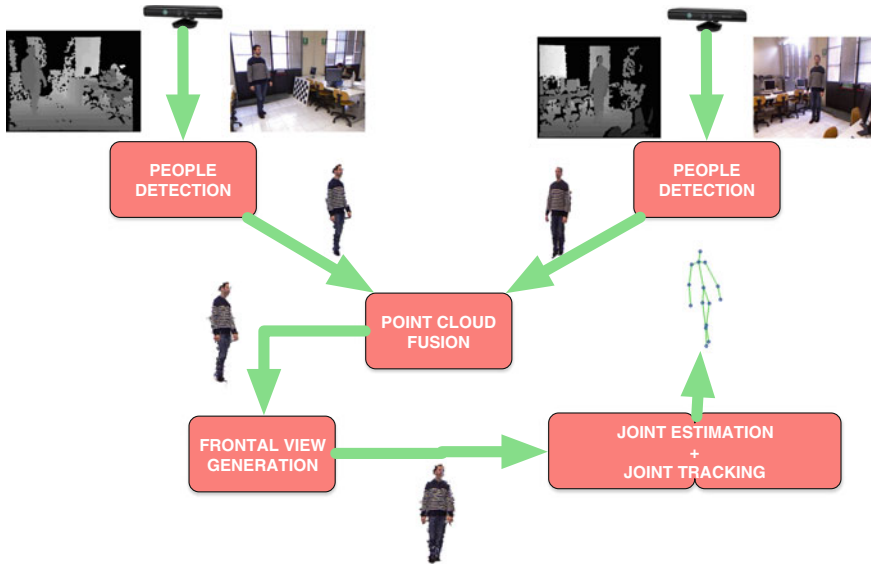


Fig. 1 Overview of the proposed system

filter. The obtained skeleton can then be reprojected to either of the original images. In Sects. 4 and 5, we will better review each step of the proposed method.

4 Multi-view Data Fusion

State-of-the-art body part detectors [1, 2] perform poorly in presence of occlusions. This case often occurs when a person is side-viewed by a camera, and having more cameras in the scene does not ensure that one of them sees the person completely. For this reason, our system exploits the perception network to perform data fusion and frontal view generation in order to provide to the body part detector a more complete depth image of the person in the scene, thus improving the final performance.

4.1 People Detection for Background Removal

The body part detector in [2] poorly estimates the lower body parts of a person when the ground plane is visible under the feet of the person or when the person is too close to the background. To overcome this problem, we added a people detection phase as a preprocessing for background removal. In this way, we build a new depth image where all the background points are set to a big depth value (e.g. 10 m), so that

the Random Forest in [2] can easily discard them from belonging to the foreground person. As for the people detector, we exploit the RGB-D people detection in [10, 11], that is publicly available in the Point Cloud Library and allows to robustly detect people and provide the point cloud points belonging to them. Then, these 3D points are reprojected to 2D to create a masked image which can be used instead of the entire depth image, improving the output of the original body part detector.

4.2 Point Cloud Fusion

The information coming from multiple cameras is here exploited at the depth level, fusing the point clouds by means of the Iterative Closest Point algorithm. In particular, considering the network of two cameras C_0, C_1 we used for the experiments, we first obtain the segmented point clouds P_0, P_1 by means of the people detector and then, given the extrinsic parameters of the network, we refer these point clouds to a common *world* reference frame. After this transformation, the resulting point clouds P_1^w and P_0^w are finely registered by means of an ICP algorithm in order to account for depth estimation errors intrinsic of the sensors [19] or possible inaccuracies in the extrinsic calibration of the network. In formulas, we obtain the point clouds:

$$P_0^w = \mathfrak{T}_0^w(P_0) \quad (1)$$

$$P_1^w = \mathfrak{T}_1^w(P_1) \quad (2)$$

$$P_{total}^w = P_0^w \oplus \mathfrak{T}_{ICP}(P_1^w) \quad (3)$$

where, \mathfrak{T}_i^j represents the transformation from the i reference frame to the j reference frame and \mathfrak{T}_{ICP} is the transformation obtained by performing ICP with the point cloud P_0^w as the target cloud and the P_1^w as the source cloud. In Fig. 2, an example of this process is shown. In order to lower the time to compute \mathfrak{T}_{ICP} , we calculate this transformation by using two downsampled versions of P_0^w and P_1^w and by limiting to 30 the number of iterations.

4.3 Frontal View Generation

The best skeleton estimation comes from frontal-viewed persons. For this reason, we want to warp the total point cloud P_{total}^w obtained at Sect. 4.2 to be frontal with respect to the camera we chose as a reference, here C_0 . In order to obtain this result, as shown in Fig. 3, we project the points of P_{total}^w to the ground, that is the xOy plane of the *world* reference frame, thus obtaining a 2D shape that usually resembles an ellipsoid O of points. We then calculate the principal components [20] of O in order to find a vector \hat{v} with the same direction of the major axis of O , that is then used to rototranslate the original P_{total}^w with M :

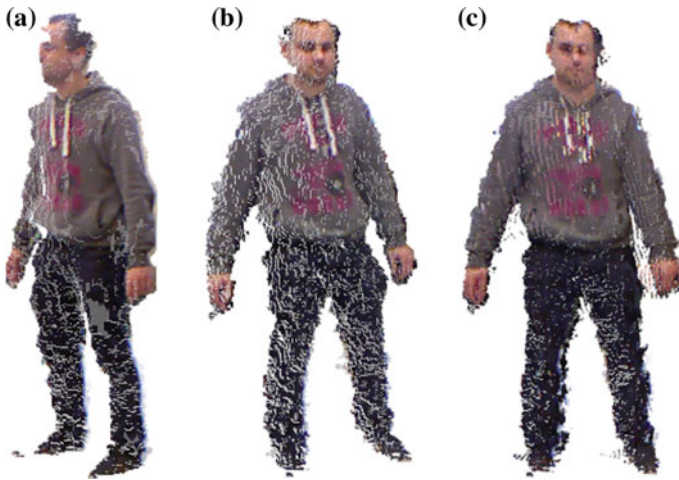


Fig. 2 An example of the depth data fusion process. In **a** the point cloud obtained from the C_1 camera, in **b** the point cloud obtained from the C_0 camera and in **c** the final fused cloud

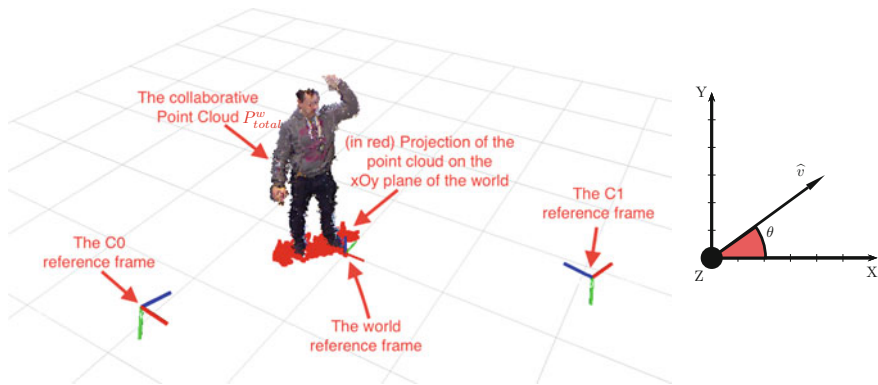


Fig. 3 The frontal view generation phase. On the *left* there are the reference system of our Kinects and the common world reference system, the collaborative cloud obtained with ICP after the people detection phase and the same cloud projected on the xy plane of the world reference frame (visible in *red*). On the *right* is shown a *top-view* of the world reference frame, the vector representing the principal component \hat{v} and the angle θ which is used for the frontal view warping. Best viewed in color

$$M = \begin{bmatrix} R & T \\ 0^{3 \times 1} & 1 \end{bmatrix} \tag{4}$$

where R is the rotation matrix which rotates a cloud of θ around the *world* z -axis and T the translation to bring the final point cloud to be centered on the *world* reference

frame. In order to compute R , we need to compute θ , which is the angle between \hat{v} and $u_x = (1, 0, 0)$. In formulas, we have:

$$\theta = \arccos\left(\frac{\hat{v} \cdot u_x}{|\hat{v}| |u_x|}\right) \quad (5)$$

$$R = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

$$T = -\begin{pmatrix} k_x \\ k_y \\ 0 \end{pmatrix}, \quad K = (k_x, k_y, k_z) = \frac{\sum_{i=0}^{|P_{total}^w|} P_{total}^w(i)}{|P_{total}^w|} \quad (7)$$

where K is the centroid of the total point cloud before the rototranslation. We can now obtain the desired frontal-view point cloud as:

$$P_{fv}^w = \{p = (x_p, y_p, z_p)^T \mid \exists q \in P_{total}^w, p = Mq\} \quad (8)$$

5 Body Skeleton Estimation

In this work, we use the algorithm in [2] to perform body part detection that is open source and available in the Point Cloud Library. This detector takes as input a depth image, that is then classified by a Random Forest. For this reason, the multi-view and frontal point cloud P_{fv}^w obtained in Sect. 4.3 has to be projected to 2D in order to create a virtual depth-image $D_{virtual}$ that could be processed by the body part detector.

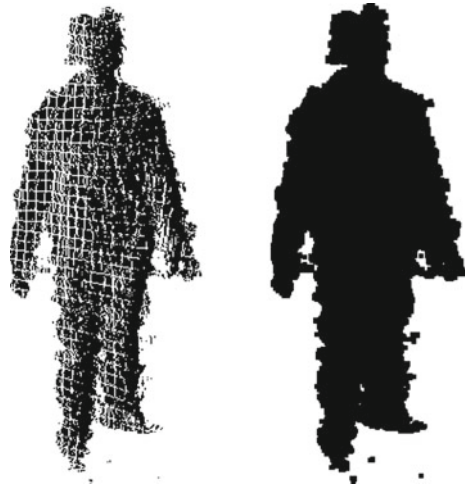
5.1 Virtual Depth Image Generation

In this work, the virtual depth image $D_{virtual}$ is estimated by projecting the points in P_{fv}^w to the image plane of the C_0 camera, that has been taken as a reference. However, this process often leaves some holes in the generated image. We thus implemented a hole-filling procedure that fills the holes with the nearest valid points until a threshold distance. In formulas:

$$D_{virtual} = \{d_{ij} = (i, j) \in \mathbb{N}^2 \mid i \in (0, 480), j \in (0, 640)\} \quad (9)$$

$$d_{ij} = \begin{cases} P_{fv}^0(i, j), & (i, j) \text{ provides a valid point in } P_{fv}^0 \\ P_{fv}^0(\bar{i}, \bar{j}), & (\bar{i}, \bar{j}) = \operatorname{argmin}\{\|(i, j) - (\hat{i}, \hat{j})\| < t \mid P_{fv}^0(\hat{i}, \hat{j}) \text{ is valid}\} \\ 10000, & \text{otherwise} \end{cases} \quad (10)$$

Fig. 4 On the *left*, the resultant point cloud re-projection to the image plane of the reference camera. On the *right*, the re-projection after the hole-filling procedure



In Fig. 4, a comparison of a sample image with and without the hole-filling procedure is shown. This hole-filled depth map is then provided as input of the body part detector.

5.2 Joint Estimation

The body part detector [2] assigns one of the 24 labels defined at training time to each pixel of the depth map and calculates the blobs of the coherent voxels with the same label.

From this preliminary segmentation, we then compute the positions of the skeleton joints in two steps. At first, we address the problem of the possibility of a single label being assigned to multiple coherent groups of voxels. This issue can be solved by combining the coherent voxel groups into a single blob or sorting them by their size with the largest one being selected for the joint calculation. Although the results of these simple methods were satisfying, the body part positions were imprecise in certain cases, especially as it comes to the smaller body parts such as hands and elbows. An improvement was achieved by building an optimal tree of the body parts, starting from the *Neck* as the root blob and further recursively estimating the child-blobs. This method is based on a pre-defined skeleton structure, which settles whether two body parts are connected as well as certain constraints regarding the expected size of the limbs.

In the second phase, the 3-D position of the joint is calculated from the selected blob. In most cases, the 3D centroid of the corresponding blob point cloud provides a good estimate for the joint position. An exception to this is the *Hip* blob, which also contains a large part of the torso. Besides, the *Shoulder* and *Elbow* joints are special cases which are described below.

Shoulders The shoulder position is calculated from the corresponding chest blob. Inside the blob point cloud, we estimate the voxel V_{y_max} with the maximum Y-value. We further build a sub-group of voxels belonging to the chest blob and having the distance to the V_{y_max} below a certain threshold (10 cm) and use the centroid of this sub-blob as the final position.

Elbows If the elbow blob was detected, we use the normal approach calculating the centroid of the blob. Otherwise, we estimate the point inside the arm blob, which has the longest distance from the previously estimated Shoulder joint.

Hips We define a certain threshold and build a sub-group of voxels belonging to the lower part of the hip blob. The centroid of this sub-group is used as the result position.

5.3 Joint Tracking over Time

The *Alpha-Beta filter* detailed in Algorithm 1 was implemented on top of the standard joint calculation to assure a consistent and continuous motion over time. This deterministic approach estimates the new position based on the predicted and measured position, with the weight of the measured position given by the parameter α , while β shows the weight of the velocity update.

Careful tuning of the α and β parameters is necessary to achieve the best results. Additionally, we have modified the update parameters for the hand joints, which usually have higher velocities than other body parts.

Algorithm 1 Applying the Alpha-Beta filter for joint tracking at a timestep t

- 1: **Input:** $Xm = [Xm_1, \dots, Xm_n]$ - measured values of m body joints; $Xp = [Xp_1, \dots, Xp_n]$ - previous values of m body joints; $V = [V_1, \dots, V_n]$ - velocities of the body joints
Output: $X = [X_1, \dots, X_n]$ - estimated joint positions; $V = [V_1, \dots, V_n]$ - updated velocities
 For each body joint k in $\{1, n\}$:
 - 2: Calculate the predicted position: $X_k = Xp_k + V_k * dt$
 - 3: Difference between measured and predicted: $R_k = Xm_k - X_k$
 - 4: New joint position value: $X_k = X_k + \alpha * R_k$
 - 5: New joint velocity value: $V_k = V_k + (\beta * R_k)/dt$
-

6 Experiments

We tested the different steps of the proposed approach with two series of RGB-D frames recorded from a network of two first-generation Microsoft Kinect. In these sequences, two different freely-moving persons were performing different movements. For measuring the accuracy, we considered the following skeleton estimation error:

$$\epsilon = \frac{\sum_{frames} \frac{\sum_{joints} \|pos_{estim} - pos_{actual}\|}{N_{joints}}}{N_{frames}} \quad (11)$$

where the ground truth for joint position pos_{actual} has been manually annotated. The system used for testing the methods proposed is an Ubuntu 14.04 machine with an Intel core i7-4770 CPU and a NVidia Geforce GTX 670 GPU. In Table 1, we reported a quantitative comparison of skeleton estimation with our methods and the original one in terms of (11). In this table, we report also a baseline multi-view approach at the skeleton level in which each fused skeleton \hat{S} is the average of the single-view skeletons S_0 and S_1 . While the original method [2] is independent from the background for the body pose estimation, the joint estimation algorithm is not and this cause the large ϵ obtained in our tests. Adding a people detection step, thus improve exponentially the performance gained by the joint estimator and our joint-tracking filter maintains the performance while smoothing the joints estimated. Our novel multi-view approach outperforms both the single-view skeleton estimation and the baseline multi-view method we used. The results achieved are from 20 to 33 % better than the single ones and up to 24 % better than the baseline skeleton-based multi-view approach. Furthermore, the computational burden needed for computing a skeleton is around 100 ms (60 ms for computing the virtual depth image plus 40 ms for the PCL skeleton computation) allowing the real-time usage of the proposed approach. In Fig. 5, we reported a qualitative comparison of skeleton estimation with these techniques.

Table 1 The performance achieved by the original method [2] and our method. PD stands for people detection and JT for joint tracking

	First person	Second person
[2]	97.82	139.50
Ours single-view with PD	34.35	35.01
Ours single-view with PD and JT	34.50	35.35
Baseline multi-view at the skeleton level	30.50	29.65
Proposed multi-view approach at the depth level	23.26	28.22



Fig. 5 Some sample frames of the dataset we used for testing the proposed approach. Each row represents a frame. The different columns represent: **a** [2] on the C_0 stream; **b** ours with PD and JT on the C_0 stream; **c** ours with PD and JT on the C_1 stream; **d** our multi-view approach re-projected on the C_0 camera

7 Conclusions

In this work, we addressed the problem of human skeleton estimation and tracking in camera networks. We proposed a novel system to fuse depth data coming from multiple cameras and to generate a frontal view of the person in order to improve the skeleton estimation that can be obtained with state-of-the-art algorithms operating on depth data. Furthermore, we improved single-camera skeletal tracking by

exploiting people detection for background removal and joint tracking for filtering joint trajectories. We tested the proposed system on hundreds of frames taken from two Kinect cameras, obtaining a great improvement with respect to state-of-the-art skeletal tracking applied to each camera. The proposed approach can be also applied to real-time scenarios given the low computational burden required.

References

1. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. *Commun. ACM* **56**(1), 116–124 (2013)
2. Buys, K., Cagniard, C., Baksheev, A., De Laet, T., De Schutter, J., Pantofaru, C.: An adaptable system for RGB-D based human body detection and pose estimation. *J. Vis. Commun. Image Represent.* **25**(1), 39–52 (2014)
3. Roitberg, A., Perzylo, A., Somani, N., Giuliani, M., Rickert, M., Knoll, A.: Human activity recognition in the context of industrial human-robot interaction. In: Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA), pp. 1–10. IEEE (2014)
4. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1290–1297. IEEE (2012)
5. Munaro, M., Ghidoni, S., Dizmen, D.T., Menegatti, E.: A feature-based approach to people re-identification using skeleton keypoints. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 5644–5651. IEEE (2014)
6. Munaro, M., Basso, A., Fossati, A., Van Gool, L., Menegatti, E.: 3D reconstruction of freely moving persons for re-identification with a depth sensor. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 4512–4519. IEEE (2014)
7. Munaro, M., Fossati, A., Basso, A., Menegatti, E., Van Gool, L.: One-shot person re-identification with a consumer depth camera. In: Person Re-Identification, pp. 161–181. Springer (2014)
8. Hsieh, J.-W., Hsu, Y.-T., Liao, H.-Y.M., Chen, C.-C.: Video-based human movement analysis and its application to surveillance systems. *IEEE Trans. Multimedia* **10**(3), 372–384 (2008)
9. Diraco, G., Leone, A., Siciliano, P.: An active vision system for fall detection and posture recognition in elderly healthcare. In: Design, Automation and Test in Europe Conference and Exhibition (DATE), 2010, pp. 1536–1541. IEEE (2010)
10. Munaro, M., Basso, F., Menegatti, E.: Tracking people within groups with RGB-D data. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2101–2107. IEEE (2012)
11. Munaro, M., Menegatti, E.: Fast RGB-D people tracking for service robots. *Auton. Robots* **37**(3), 227–242 (2014)
12. Munaro, M., Lewis, C., Chambers, D., Hvass, P., Menegatti, E.: RGB-D human detection and tracking for industrial environments. In: Intelligent Autonomous Systems 13, pp. 1655–1668. Springer (2016)
13. Rusu, R.B., Cousins, S.: 3D is here: Point cloud library (PCL). In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 1–4. IEEE (2011)
14. Kanaujia, A., Haering, N., Taylor, G., Bregler, C.: 3D human pose and shape estimation from multi-view imagery. In: 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 49–56. IEEE (2011)
15. Lora, M., Ghidoni, S., Munaro, M., Menegatti, E.: A geometric approach to multiple view-point human body pose estimation. In: 2015 European Conference on Mobile Robots (ECMR), pp. 1–6. IEEE (2015)

16. Gao, Z., Yu, Y., Zhou, Y., Du, S.: Leveraging two kinect sensors for accurate full-body motion capture. *Sensors* **15**(9), 24297–24317 (2015)
17. Yeung, K.-Y., Kwok, T.-H., Wang, C.C.: Improved skeleton tracking by duplex kinects: a practical approach for real-time applications. *J. Comput. Inf. Sci. Eng.* **13**(4), 041007 (2013)
18. Besl, P.J., McKay, N.D.: Method for registration of 3-D shapes. In: *Robotics-DL Tentative*, pp. 586–606. International Society for Optics and Photonics (1992)
19. Zennaro, S., Munaro, M., Milani, S., Zanuttigh, P., Bernardi, A., Ghidoni, S., Menegatti, E.: Performance evaluation of the 1st and 2nd generation Kinect for multimedia applications. In: *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE (2015)
20. Rao, C.R.: The use and interpretation of principal component analysis in applied research. *Sankhyā: Ind. J. Stat. Ser. A* 329–358 (1964)

Author Index

A

Al Assad, Omar, [157](#)
An, Qi, [31](#), [91](#)
Antonello, Morris, [757](#)
Asama, Hajime, [91](#), [1089](#)
Asamura, Tomohiro, [255](#)

B

Balajee, Kannan, [157](#)
Baloch, Ghulam, [157](#)
Behnke, Sven, [123](#), [855](#)
Benz, Nicolai, [177](#)
Bergen, Marcus, [177](#)
Berns, Karsten, [143](#), [883](#)
Bi, Sheng, [1129](#)
Bortoletto, Roberto, [329](#)
Buck, Sebastian, [375](#), [403](#), [911](#)

C

Cai, Bin, [771](#), [1061](#)
Cai, Hegao, [781](#)
Cao, Qixin, [447](#), [1143](#)
Carraro, Marco, [485](#), [1155](#)
Cenci, Annalisa, [473](#)
Cerruti, Giulio, [357](#)
Chen, Weidong, [229](#), [301](#), [573](#), [587](#), [1013](#)
Chen, Xin, [1129](#)
Chen, Yiping, [459](#)
Cheng, Ming, [459](#)
Cheruiyot, Gilbert, [1143](#)
Chiba, Ryosuke, [39](#)
Chu, Leeyee, [499](#)
Cui, Long, [1027](#)

D

D'Angelo, Antonio, [389](#)
Daowei, Jiang, [695](#)
Degl'Innocenti, Dante, [389](#)

Del Pobil, Angel P., [243](#), [839](#)
Dong, Min, [1129](#)
Dong, Sun, [1117](#)
Dorronzoro Zubiete, Enrique, [315](#)
Droeschel, David, [123](#), [855](#)
Duan, Yi, [343](#)
DU, Boyang, [601](#)
Duran, Angel J., [839](#)

F

Fan, Yang, [659](#)
Fan, Zhengjie, [285](#)
Fang, Zheng, [433](#)
Farinelli, Alessandro, [629](#)
Fernández-Vargas, Jacobo, [499](#)
Fischinger, David, [243](#)
Fleischmann, Patrick, [883](#)
Forman, Douglas, [157](#)
Frontoni, Emanuele, [473](#)
Fu, Xiangyu, [587](#)
Fu, Zhuang, [563](#)
Funato, Tetsuro, [31](#)
Furuta, Yuki, [811](#)

G

Gao, Feng, [725](#)
Gao, Lifu, [1049](#)
Gatto, Alejandro, [285](#)
Ge, Yao, [675](#)
Gefen, Yonatan, [157](#)
Gini, Maria, [629](#)
Giovannini, Bruno S., [213](#)
Göckel, Fabian, [143](#)
Guo, Liangbin, [1049](#)
Guo, Lindong, [1103](#)
Guo, Weibin, [993](#), [1037](#), [1049](#)
Guo, Wei-Zhong, [357](#)
Guoliang, Zhang, [601](#), [659](#)

H

Haga, Nobuhiko, 31
 Hagiwara, Yoshinobu, 3, 53
 Hamada, Masashi, 31
 Han, Odbal, 1037, 1061
 Hanten, Richard, 403, 911
 Haubner, Hans-Jörg, 177
 Holovashchenko, Viktor, 157
 Hoshikawa, Suguru, 269, 977
 Hosoda, Koh, 77, 343
 Hu, Tianjiang, 185
 Hu, Yan, 725
 Huang, Haibo, 1117
 Huang, Zhifeng, 549, 955
 Huskić, Goran, 375

I

Ikemoto, Shuhei, 77, 343
 Imai, Kazuya, 539
 Imamoglu, Nevrez, 315
 Inaba, Masayuki, 811
 Inagaki, Yuto, 811
 Inamura, Tetsunari, 3
 Inoue, Yohei, 869
 Ishihara, Masahiro, 977
 Ishii, Akira, 19
 Ishii, Soichi, 825

J

Jain, Arpit, 157
 Ji, Yonghoon, 1089
 Jing, Zeng, 601
 Jiang, Li, 965
 Jiang, Ping, 39, 955
 Jiang, Yinlai, 269, 977
 Jung, Jae-Hun, 687

K

Kanai-Pak, Masako, 549
 Kato, Ryu, 269
 Kita, Kahori, 315, 499, 525, 539
 Kitajima, Yasuko, 549
 Kondo, Toshiyuki, 19
 Kou, Jieting, 771
 Krombach, Nicola, 855
 Kumazawa, Nobuhiro, 69
 Kuwahara, Noriaki, 549

L

Lee, Dong-Hyuk, 687
 Lee, Jang-Myung, 687
 Lei, Yuan, 659
 Lenz, Christian, 123
 Li, Changle, 713

Li, Liang, 1103
 Li, Liyi, 713
 Li, Shuai, 157
 Li, Tianyi, 897
 Li, Xiangpeng, 1117
 Liciotti, Daniele, 473
 Lin, Chingszu, 549, 955
 Lin, Zhenkun, 1003
 Liu, Gangfeng, 713
 Liu, Hong, 965
 Liu, NaiLong, 1027
 Liu, Wei, 771, 1061
 Liu, Xiangxiao, 343
 Liu, Yi, 993, 1037, 1049, 1073
 Liu, Yixiang, 1003
 Liu, Yu, 965
 Liu, Yuan, 1049
 Liu, Yulong, 459
 Liu, Zhe, 573, 587
 Lu, Junguo, 573
 Luan, Chengzhi, 433

M

Maeda, Jukai, 549
 Maeda, Yusuke, 255, 825, 939
 Mancini, Adriano, 473
 Martinez-Martin, Ester, 243
 Mastrogiovanni, Fulvio, 357, 615
 Matsuda, Fumihiko, 69
 Menegatti, Emanuele, 485, 757, 925, 1155
 Mimura, Tomohiro, 3
 Miura, Jun, 869
 Mizuta, Kotaro, 737
 Mukaino, Masahiko, 69
 Munaro, Matteo, 485, 757, 925, 1155
 Murai, Yuta, 977

N

Nakagawa, Yoshinori, 825
 Nakahata, Keigo, 315
 Nakamura, Mitsuhiro, 549
 Nakamura, Tatsuhiko, 269, 977
 Nana, Pang, 1129
 Nardi, Daniele, 105
 Nattero, Cristiano, 615

O

Ogata, Taiki, 549, 955
 Ohtsuka, Kei, 69
 Oishi, Shuji, 869
 Okada, Hiroyuki, 737
 Okada, Kei, 811
 Oliveira, Hugo A., 213
 Oswald, Moritz, 883

Ota, Jun, [39](#), [549](#), [795](#), [955](#)
 Otte, Sebastian, [403](#)
 Owaki, Dai, [31](#)

P

Pagello, Enrico, [285](#), [329](#)
 Pang, Zhenan, [659](#)
 Park, Young-Sik, [687](#)
 Parker, James, [629](#)
 Patrick, Romano, [157](#)
 Pech, C. Robert, [911](#)
 Peng, Wei, [301](#)
 Pfister, Thomas, [883](#)
 Piovesan, Davide, [329](#)
 Pitteri, Giorgia, [925](#)
 Potena, Ciro, [105](#)
 Pretto, Alberto, [105](#)

Q

Qi, Chenkun, [725](#)
 Qiu, Qiang, [1143](#)

R

Rahimi, Mehdi, [511](#)
 Recchiuto, Carmine Tommaso, [199](#)
 Reilly, Thomas, [329](#)
 Ren, Anye, [725](#)
 Roitberg, Alina, [1155](#)
 Rosa, Paulo F.F., [213](#)
 Rosendo, Andre, [343](#)

S

Saito, Yoshito, [939](#)
 Saitoh, Eiichi, [69](#)
 Sakoda, Shintaro, [977](#)
 Sasaki, Takuya, [31](#)
 Sassa, Ryoma, [69](#)
 Schmidt, Daniel, [143](#)
 Schüller, Sebastian, [123](#)
 Schwarz, Max, [123](#)
 Sekine, Masashi, [315](#), [525](#)
 Sen, Shiraj, [157](#)
 Sgorbissa, Antonio, [199](#)
 Sharma, Pramod, [157](#)
 Shen, Lincheng, [417](#)
 Shen, Yantao, [511](#)
 Shiota, Kouki, [525](#)
 Shirafuji, Shouhei, [39](#), [795](#)
 Song, Caiwei, [713](#)
 Song, Huang, [965](#)
 Spohn, Leon, [177](#)
 Strand, Marcus, [177](#)
 Sun, Zhiming, [433](#)

T

Takagi, Takehiko, [977](#)
 Takakusaki, Kaoru, [39](#)
 Takano, Toshiaki, [53](#)
 Takayama, Shinichiro, [977](#)
 Tan, Huan, [157](#)
 Tanabe, Shigeo, [69](#)
 Tanahashi, Kazumasa, [269](#)
 Taniguchi, Akira, [53](#)
 Taniguchi, Tadahiro, [3](#), [53](#)
 Tapio, V. J. Tarvainen, [525](#)
 Terada, Yuri, [795](#)
 Theurer, Charles, [157](#)
 Tian, Shihe, [563](#)
 Tosello, Elisa, [285](#)
 Tsuchiyama, Kazuhiro, [69](#)

U

Ueda, Ryuichi, [737](#)
 Urino, Takuya, [77](#)

V

Vincze, Markus, [243](#)
 Vonscheidt, Denis, [177](#)

W

Wang, Bing, [1103](#)
 Wang, Cheng, [459](#)
 Wang, Chunxiang, [897](#), [1103](#)
 Wang, Daqing, [1049](#)
 Wang, Guangwei, [645](#)
 Wang, Hesheng, [229](#), [573](#), [587](#), [1013](#)
 Wang, Jingchuan, [301](#), [573](#), [587](#)
 Wang, Kundong, [675](#)
 Wang, Shuyuan, [185](#)
 Wang, Wenshan, [447](#), [1143](#)
 Wang, Xun, [417](#)
 Wang, Zengfu, [771](#), [993](#), [1037](#), [1049](#), [1061](#),
[1073](#)
 Wanpeng, Lv, [53](#)
 Wei, Qing, [1027](#)
 Wei, Zhixuan, [229](#)
 Wheeler, Frederick, [157](#)
 Wu, Baoyuan, [993](#), [1037](#), [1049](#)

X

XU, Jun, [601](#)
 Xu, Qingsong, [645](#)
 Xu, Xiaojun, [897](#)

Y

Yabuki, Yoshiko, [269](#), [977](#)
 Yamakawa, Hiroshi, [91](#), [737](#)

Yamashita, Atsushi, [91](#), [1089](#)
Yan, Jihong, [781](#)
Yan, Qingquan, [993](#), [1037](#)
Yang, Hao, [1117](#)
Yang, Ming, [897](#), [1103](#)
Yang, Zhen, [563](#)
Yano, Shiro, [19](#), [53](#)
Yokoi, Hiroshi, [269](#), [977](#)
Yong, Zhang, [601](#)
Yoon, Ha-Neul, [687](#)
Yozu, Arito, [31](#)
Yu, Wenwei, [315](#), [499](#), [525](#), [539](#)
Yuan, Liang, [695](#)
Yue, Zhao, [1129](#)

Z

Zang, Xizhe, [1003](#)

Zell, Andreas, [375](#), [403](#), [911](#)
Zhang, Daibing, [417](#)
Zhang, Jianwei, [417](#)
Zhang, Qiang, [993](#), [1037](#), [1049](#), [1073](#)
Zhang, Xinbin, [781](#)
Zhao, Jie, [713](#), [781](#), [1003](#)
Zhao, Liang, [1013](#)
Zhao, Xianchao, [725](#)
Zheng, Hui, [563](#)
Zheng, Zhong, [771](#), [993](#), [1061](#)
Zhou, Ling, [993](#), [1073](#)
Zhou, Xiang, [897](#)
Zhu, Xiaoxiao, [447](#)
Zhu, Yanhe, [1003](#)
Zingaretti, Primo, [473](#)