Advances in Intelligent Systems and Computing 447

Jong-Hwan Kim
Fakhri Karray
Jun Jo
Peter Sincak
Hyun Myung *Editors*

Robot Intelligence Technology and Applications 4

Results from the 4th International Conference on Robot Intelligence Technology and Applications

Springer

Advances in Intelligent Systems and Computing

Volume 447

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.

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

Jong-Hwan Kim · Fakhri Karray
Jun Jo · Peter Sincak · Hyun Myung
Editors

Robot Intelligence Technology and Applications 4

Results from the 4th International Conference on Robot Intelligence Technology and Applications

Springer

Editors
Jong-Hwan Kim
Korea Advanced Institute of Science and Technology (KAIST)
Daejeon
Korea, Republic of (South Korea)

Fakhri Karray
Department of Electrical and Computer Engineering (ECE)
University of Waterloo
Waterloo, ON
Canada

Jun Jo
Griffith University
Gold Coast, QLD
Australia

Peter Sincak
Center for Intelligent Technologies
Technical University of Kosice
Kosice
Slovakia

Hyun Myung
Korea Advanced Institute of Science and Technology (KAIST)
Daejeon
Korea, Republic of (South Korea)

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-31291-0 ISBN 978-3-319-31293-4 (eBook)
DOI 10.1007/978-3-319-31293-4

Library of Congress Control Number: 2016939939

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 Switzerland

Preface

Recently designed and built "Intelligent" robots are becoming more and more powerful in terms of human-like abilities in terms of perception, cognition, and decision-making. While humans have far superior intellectual abilities as compared to machines and robots, the advantages of "intelligent" robots remain in their excellent way for connecting and communicating with other machines, robots, or devices. These facts result in the so-called "networked robots" which are connected to different external sensor arrays, cameras, or even other robots. The robots can extend their information source and overcome intelligence of humans. So in fact we live in an era when humans with highly developed intellect are being compared with multisource supported robots with ability to get pieces of information from cyberspace or any kind of local and remote sensors in real-time manner.

This is the fourth edition that aims at serving the researchers and practitioners in related fields with a timely dissemination of the recent progress on robot intelligence technology and its applications, based on a collection of papers presented at the 4th International Conference on Robot Intelligence Technology and Applications (RiTA), held in Bucheon, Korea, December 14–16, 2015. For better readability, this edition has the total of 49 articles grouped into three parts: Part I: Ambient, Behavioral, Cognitive, Collective, and Social Robot Intelligence, Part II: Computational Intelligence and Intelligent Design for Advanced Robotics, Part III: Applications of Robot Intelligence Technology, where individual chapters, edited by Fakhri Karray, Jun Jo, Peter Sincak, Hyun Myung along with Jong-Hwan Kim, begin with a brief introduction.

Part I: Ambient, Behavioral, Cognitive, Collective, and Social Robot Intelligence

RiTA conference is one of the few conferences which addresses **R**obot **I**ntelligence **T**heory and **A**pplications. These perspectives of robots intelligence technology including ambient, behavioral, cognitive, collective, genetic, and social intelligences are very important in creating an alternative to single self-learning robot. In the first sentence of this part we have written the word Intelligence in quotes so honestly believe that the notion "robot intelligence" is going to evolve and the artificial intelligence in many of their various forms is a key factor for creating robots as machines with different levels of intelligence related to given task from humans or companionship with humans.

In this first part, we find 24 chapters dealing with various partial contributions to ambience, cognition, behavioral modeling, social robots and collective intelligence. The chapters can be categorized into four sections;

A. Robot Navigations, Localization, Path Planning, and Related Problems in Various Environments

1. Behavior and Path Planning for the Coalition of Cognitive Robots in Smart Relocation Tasks
2. Trajectory Generation Using RNN with Context Information for Mobile Robots
3. Fast and Smooth Replanning for Navigation in Partially Unknown Terrain: The Hybrid Fuzzy-D*lite Algorithm
4. Accurate Localization in Urban Environments Using Fault Detection of GPS and Multi-sensor Fusion
5. Simultaneous Localization and Mapping with a Dynamic Switching Mechanism (SLAM-DSM)
6. ROSLAM—A Faster Algorithm for Simultaneous Localization and Mapping (SLAM)
7. Adaptive Computation Algorithm for Simultaneous Localization and Mapping (SLAM)
8. Interactive Markerless Augmented Reality System Based on Visual SLAM Algorithm

B. Robot Movement and Related Problems of Control

1. Control Strategy Design for Throw-in Challenge in a Humanoid Robot Soccer Game
2. Study on a Two-Staged Control of a Lower-Limb Exoskeleton Performing Standing-Up Motion from a Chair
3. Adaptive Control for Directional Drilling Systems with Delay and Parameter Uncertainty
4. Design and FPGA Implementation of a Fuzzy-PI Controller for Omnidirectional Robot System

5. Android Based Mobile Robotic Platform Performance Testing for Real-Time Navigation
6. Implementation and Control of a Bistable Two wheeled Inverted Pendulum-Type Mobile Robot
7. Autonomous Control of a Drone in the Context of Situated Robotics
8. Fast *MAV* Control by Control/Status OO-Messages on Shared-Memory Middleware.

C. Robot Behaviors, Agent System for Complex Tasks

1. Framework and Modeling of a Multi-robot Simulator for Hospital Logistics
2. Emotion in Robot Decision Making
3. Multi-robot Task Allocation Using Clustering Method

D. Human Robot Interaction and Related Topics

1. Modeling of Violin Playing Robot Arm with MATLAB/SIMULINK
2. Gaze Control of Humanoid Robot for Learning from Demonstration
3. A Novel Design of a Full Length Prosthetic Robotic Arm for the Disabled
4. Preliminary Study in a Novel Robotic-Assisted Femoral Shaft Fracture Reduction System
5. Pet Care Robot for Playing with Canines

Generally we can consider these chapters as contributions to the selected issues of Robot Intelligence from Theory and Applications. These are extremely important since communities in Artificial Intelligence and Robotics tend to be isolated and we do need to prevent the re-invention of artificial intelligence in robotics community to help science, research and technology. On the other hand the practical application of artificial intelligence in Robotics can give a valuable feedback to artificial intelligence researchers to rethink the theory since applications are providing an important experience and data for the theory. Both communities need to be in a closer contact and benefit from each other while sharing the obtained knowledge for better future of mankind.

Part II: Computational Intelligence and Intelligent Design for Advanced Robotics

This part consists of two segments of different topics which cover broad spectrum of topics related to robot intelligence: Computational Intelligence and Intelligent robot design.

The *Computational Intelligence* is a methodology involving computing that exhibits an ability to learn or to deal with new situations. It usually comprises of soft computing techniques such as evolutionary computation, neural networks, and fuzzy systems.

By representing a problem with a chromosome and genes, and describing fitness of this chromosome in the form of objective function, evolutionary computation effectively solves a problem using meta-heuristics inspired by genetics. The problems hard to be solved due to their inherent complexity, or the problems that do not have mathematical model necessary for classical optimization methods or hard computing techniques, can be candidates for the application of computational intelligence approach. The robots that use these problem solving capabilities can be regarded as having computational intelligence. The following chapters illustrate very well these approaches.

A. Computational Intelligence

1. Making a More Reliable Classifier via Random Crop Pooling
2. Learning with Learning Robots: A Weight-Lifting Project
3. Ensemble of Vector and Binary Descriptor for Loop Closure Detection
4. Learning with Small Autonomous Robots

The rest of the chapters in this part are dedicated to the *Design of Intelligent Robots* using intelligent perception and sensing capabilities. These robots can be also designed to be used as educational purposes as can be seen in the following chapters:

B. Design of Intelligent Robots

1. Design and Implementation of Double Passing Strategy for Humanoid Robot Soccer Game
2. Development of Motion Management System for the Robot Soccer Using Multiple Humanoid robots
3. Small-Size Robot Platform as Test and Validation Tool for the Development of Mechatronic Systems
4. Soft Robotics Technology and a Soft Table for Industrial Applications
5. Golf Playing DARwIn-OP: A Theoretical Approach
6. An Advanced Spider-Like Rocker-Bogie Suspension System for Mars Exploration Rovers
7. System of 3-D Printed Components for the Rapid Prototyping of Legged Robots

Part III: Applications of Robot Intelligence Technology

There are so many aspects of life that robots may be applied to make everyday life safer and more convenient. Robots should be intelligent enough to go beyond simple preprogrammed reactions to environmental stimuli. Intelligent robots should be able to solve complicated problems, self-learn and perform actions that were not explicitly programmed, and thus be able to participate in complex interactions with humans or other robots. This part presents 14 chapters that will introduce some of the many possible applications of robots with intelligent technologies. The intelligent technologies utilized in the chapters are as follows:

A. Computer Algorithms and Computational Intelligence techniques

1. A Rigid and Soft Combined Robot that Is Designed to Be Used in Confined Spaces
2. Design and Experimental Research of Pneumatic Soft Humanoid Robot Hand

B. Computer Vision and Image Processing

1. Concept of Distributed Processing System of Image Flow
2. A Likelihood-Based Data Fusion Model for the Integration of Multiple Sensor Data: A Case Study with Vision and Lidar Sensors
3. A Vision-Based 6-DOF Displacement Measurement Method for Assembling PC Bridge Structures Using a Planar Marker
4. A Color Constancy Algorithm Using Photodetector Characteristics of a Camera for Indoor Scenes
5. Real-Time Ball Detection and Following Based on a Hybrid Vision System with Application to Robot Soccer Field
6. Toward Autonomous UAV Landing Based on Infrared Beacons and Particle Filtering
7. Local Obstacle Avoidance Using Obstacle-Dependent Gaussian Potential Field for Robot Soccer
8. Methods of Visual Navigation of the UAV Flying Over the Nonplanar District

C. Various Sensor Technologies/Devices other than cameras

1. Porting Experiment of Robotic Machining Application Using ORiN SDK and Design of 3D Printer-Like Interface
2. Rotation Vector Sensor-Based Remote Control of a Mobile Robot via Google Glass

D. Wireless Technologies

1. A Comparative Study of Wi-Fi and Bluetooth for Signal Strength-Based Localisation

E. Cloud Computing, Simulation, Mechanical Structure and Solar Energy

1. Cloud-Based Robots and Intelligent Space Teleoperation Tools

We do hope that readers find the Fourth Edition of Robot Intelligence Technology and Applications, RiTA 4, stimulating, enjoyable and helpful for their research endeavors.

Jong-Hwan Kim
Fakhri Karray
Jun Jo
Peter Sincak
Hyun Myung

Contents

Part II Computational Intelligence and Intelligent Design for Advanced Robotics

Part III Applications of Robot Intelligence Technology

Part I
Ambient, Behavioral, Cognitive, Collective, and Social Robot Intelligence

Behavior and Path Planning for the Coalition of Cognitive Robots in Smart Relocation Tasks

Aleksandr I. Panov and Konstantin Yakovlev

Abstract In this paper we outline the approach of solving special type of navigation tasks for robotic systems, when a coalition of robots (agents) acts in the 2D environment, which can be modified by the actions, and share the same goal location. The latter is originally unreachable for some members of the coalition, but the common task still can be accomplished as the agents can assist each other (e.g., by modifying the environment). We call such tasks smart relocation tasks (as they cannot be solved by pure path planning methods) and study spatial and behavior interaction of robots while solving them. We use cognitive approach and introduce semiotic knowledge representation—sign world model which underlines behavioral planning methodology. Planning is viewed as a recursive search process in the hierarchical state-space induced by sings with path planning signs residing on the lowest level. Reaching this level triggers path planning which is accomplished by state-of-the-art grid-based planners focused on producing smooth paths (e.g., LIAN) and thus indirectly guarantying feasibility of that paths against agent's dynamic constraints.

Keywords Behavior planning · Task planning · Coalition · Path planning · Sign world model · Semiotic model · Knowledge representation · LIAN

1 Introduction

In pursuit of higher autonomy degree of modern robotics systems researchers combine various methods and algorithms of artificial intelligence, cognitive science, control theory into so-called Intelligent Control Systems (ICS) [1, 2]. These systems are the collections of software modules automating robot behavior and are conventionally organized in a hierarchical fashion. Usually three levels of control—strategic, tactical, and reactive (or named in another way but still bearing the same sense)—

A.I. Panov (✉) · K. Yakovlev
Federal Research Center "Computer Science and Control" of RAS,
pr. 60-letiya Octyabrya 9, 117312 Moscow, Russia
e-mail: pan@isa.ru

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_1

are distinguished [3]. In this work, we address the planning problem and examine planning methods on both strategic and tactical levels and their interaction. On the strategic level, we assume that there exist a description of situations and goals, and the search space induced by such descriptions is processed to produce a valid plan. Typically in AI planning [4] first-order logic is used to model the world as well as specialized first-order logic languages—PDDL and its derivatives—are used to formalize robot's actions [5, 6]. Planning relying on these models and languages is known as task planning. In our work a new formalism—sign world model—is introduced which is based on cognitive theories, takes into account results of recent cognitive and neurophysiologic research and thus makes robot's behavior more human-like, robust and versatile. We refer to planning based on sign world model as to behavior planning. As for the tactical level of control system, it deals mainly with navigation tasks so planning is considered in spatial (geometrical) sense and is aimed at finding a path (feasible trajectory) for the robot. Both planning activities—behavior planning and path planning—despite the common term involved in their names utilize different models and algorithms and commonly are studied independently. In the present work, we study them as a part of coherent framework. One should say that there exists a limited number of the approaches of task (not behavior) and path planning integration, see [7, 8] for example. These approaches mainly examine some aspects of task and path planning integration when there is a single robot interacting with the environment. In our work, we study the behavior of the coalition of robots and how integration of planning activities on both strategic and tactical layer can affect such behavior. We examine navigation tasks in 2D world which can be transformed by the robots' actions. More precisely, we investigate the case when unsolvable for some member of the coalition problem can be solved if other robots alter their plans and assist each other. We call such tasks—smart relocation tasks (as they cannot be accomplished by path planning only methods).

The latter of the paper is organized as follows—in Sect. 2 related works regarding task and path planning are discussed. Section 3 contains the description of the smart relocation task we are interested in. Novel world modeling formalism, which utilizes cognitive approach, is introduced in Sect. 4 and in Sect. 5 planning method based on this formalism is described. Suggested path planning approach is discussed in Sect. 6. Model example in studied in Sect. 7.

2 Related Works

2.1 World Modeling and Behavior Planning

Behavior (task) planning is the main objective of control systems based on cognitive architectures. Well-known SOAR [9, 10] system is considered as the industry standard in this area. In SOAR, as well is in majority of other cognitive control systems, agent's memory is separated into the long-term memory, the short-term memory and the memory of estimates. Objects, situations and goals are represented in the short-term memory in the form of attribute descriptions. The long-term memory contains transitions (operators) between short-term memory states and is represented by AI rules [11].

Agent's planning procedure in SOAR consists of a sequence of decisions, where the aim of each decision is to select and apply an operator in service of the agent's goals. The simple decision circle contains five steps: encode perceptual input, activate rules to elaborate agent's state (propose and evaluate operators) in parallel, select an operator, activate rules in parallel that apply the operator and then process output directives and retrievals from long-term memory.

Similar knowledge representation and planning method are implemented in other cognitive control systems. In the Icarus project [12, 13] the division of the long-term memory into conceptual and action memories was introduced. Planning procedure of Icarus relies on recursive action decomposition up to low-level actions, called skills. Skill sequence is executed when start situation is satisfied in short-term memory. In Clarion [14, 15] some rules of action choosing are based on neural networks. Thus knowledge representation in Clarion contains not only explicit (attributive) component but also an implicit one. The learning process on the set of predefined precedents is the distinctive feature of this system.

Modern algorithms of behavior (task) planning use so-called STRIPS description of planning domain [16]. One of the main directions in task planning is the development of special graph structures encoding both state descriptions and state transitions for further search. The first algorithm using graph representation was Graphplan [17]. Graphplan search procedure is executed on the special layered compact planning graph and returns a shortest-possible partial-order plan or state that indicates the absence of the valid plan.

Further research in this area was concentrated on development of specialized search algorithms for these graph structures. For example in the Fast Forward (FF) [18] and the Fast Downward (FD) planning systems [19] heuristic search is used. These planners are aimed at solving general deterministic planning problems encoded in the propositional fragment of PDDL description [6] and search the state space in the forward direction. FF, FD, and other widespread PDDL-based planners use the propositional representation with special implicit constraints being considered in some cases. For example, FD planner computes its causal graph heuristic function taking these implicit constraints into account as well as using hierarchical decompositions of planning tasks.

Another remarkable heuristic planning system is LAMA [20]. It uses pseudo-heuristic derived from landmarks—propositional formulas that must be true in every solution of a planning task. The LAMA system builds plan using finite domain rather than binary state variables as in the FF planner.

One should note that the propositional language for task description is not relevant to many real problems. Thus extensions of the language and development of hybrid planning domains is appealing research area. For example, UPMorphi universal planner [21] is capable of reasoning with mixed discrete and continuous domains and fully respect the semantics of PDDL+ [22]. UPMorhi performs universal planning on some initial discretization and checks the correctness of the result. If the validation fails, discretization is refined and algorithm is reinvoked.

All of the above-mentioned and other existing planners are not suitable for the cooperative behavior (task) planning. Special knowledge representation such as MA-PDDL [23] should be used in this case. These representations and planners based on them should solve symbol grounding problem [24] and support goal-setting and role distribution procedures. Such requirements can be met by the sign representation, which is based on the models of cognitive functions [25] and neurophysiological studies of the cognition process [26, 27]. We will use this approach to realize communication protocol for cooperative planning and providing a link between the symbolic models and sensor (low level) data.

2.2 Spatial Modeling and Path Planning

Traditionally in artificial intelligence and robotics path planning is viewed as a graph search process. Agent's knowledge about the environment is encoded into the graph model and the search for a path on that graph is performed. Typically, graph's vertices correspond to the locations an agent can occupy and edges—to the trajectories it can traverse (for example—straight sections or curves of predefined lengths and curvatures). Weighting function, which assigns weights to the edges, is commonly used to quantitatively express any characteristics of the corresponding trajectories (length, energy cost, risk of traversing, etc.). So to plan a path one needs to a) construct a graph model out of the environment description available to an agent b) find a (shortest) path on that graph.

The most widespread graph models used as the spatial world model of an agent are Visibility Graphs [28], Voronoi Diagrams [29], Navigation Meshes [30], Regular Grids [31] etc. Each of them needs its own algorithm to be executed to transform raw information about the environment to the model. In case environment is compound of the free space and the polygonal obstacles (the most widespread case), two graph models are typically used—visibility graphs and regular grids. Constructing visibility graph is computationally burdensome and each time goal position changes additional calculations should be performed to add corresponding edges to VG [32]. Algorithm of grid construction is much more simple—its complexity is a constant in respect of number of obstacles' vertices and edges, and no additional calculations should be made when goal or start position alters. So, grids can be referred to as simple yet informative graph models, and in most cases it is the grids that are used for path planning. Another reason grids are so widespread is that new knowledge on the environment gained via sensor information processing can be easily integrated into them [33] without the necessity to re-invoke graph construction algorithm, which significantly saves computational resources.

After the graph is constructed, the search for a path is performed (typically, the shortest path is targeted). There exist a handful of algorithms for that: Dijkstra [34], A* [35]—which is the heuristic modification of Dijkstra, and many of their derivatives: R* [36], Theta* [37], JPS [38], D* Lite [39] etc. Some of these algorithms are specifically tailored to grid path planning (like JPS or Theta*) some work on any graphs (D* Lite, R*) with A* and Dijkstra being the most universal ones (and the most computationally ineffective while solving practical tasks as well).

If we are talking not about an abstract agent, which can move in any directions, with any speed and acceleration, and stop instantly, we need to take into account agent dynamic constraints while searching for a path. Common way to consider these constraints is to incorporate them somehow into the graph model or, which is nearly the same, into the search space—see [40] for example. The main problem here is that the search space becomes orders of magnitude times larger, especially when an agent exhibits rather complex dynamics (for example—multirotor UAV). Another problem here is that admissible, monotone, well-informed heuristics utilized to guide the search can be easily introduced only for the spatial-only search spaces, which is not the case anymore. Summing up that mentioned above, one can claim that it may be beneficial to stay within spatial-only search spaces but search for such paths that indirectly guarantee feasibility against the agent's dynamic constraints, e.g., smooth paths not containing sharp turns. One of the recently introduced approaches in this area, is planning for angle-constrained paths [41]. We believe that this approach is very promising and suggest using LIAN algorithm [42] for agent's path planning. To the best of our knowledge it is the only angle-constrained path planning algorithm which is sound and complete (in respect to its input parameters).

When talking about coalitions of agents and multi-agent grid path planning the most well-studied problem is resolving spatial conflicts for groups of agents with primitive dynamics, e.g., agents that can move from an arbitrary grid cell to any of its eight adjacent neighbors and stop (and later on start) moving instantaneously. There exist both sound and complete but very computationally expensive methods of solving this task [43] and fast but incomplete algorithms [44, 45]. Much less attention is paid to the problem of agents spatial interaction when planning for a path—a problem which will be addressed in our work in more details.

2.3 Summary

Currently existing cognitive control systems and PDDL-based planners do not consider some important features of the planning problem in case coalition of interacting agents is involved. Dynamic formation of goals and goal sharing in the context of changing environment impose special restrictions on the knowledge representation to be used by planning systems. Necessity to divide beliefs of a single agent into communicable and personal parts presents another restriction. It is also worth noting that within existing task planning frameworks little attention is paid to coordination of path planning process and behavior knowledge about the environment. Regarding path planning itself one can state that grid-based path planning is the most widespread methodology as grids are simple yet informative spatial models and a handful of methods tailored to grid path finding exist. Unfortunately grid-based paths do not take into account agent's dynamic constraints while incorporating dynamic laws encodings into the search process severely degrades overall performance (due to the enormous extension of the search space). So it can be beneficial to stay within spatial only search spaces but search for a specific, geometrically constrained class of paths and thus indirectly guaranty path's feasibility. Further on we will present a coherent task-path planning framework which addresses all the mentioned concerns and bottlenecks.

3 Considered Case

Further on we will use the term agent as well as robot (robotic system) following the conventions of AI literature.

We consider the following task. The coalition of agents $A = \{A_1, \dots, A_N\}$ act in the static environment (workspace) which is the rectangular area of 2D Euclidean space U: $x_{min} \le x \le x_{max}, y_{min} \le y \le y_{max}$. U is comprised out of the free space U_{free} and the obstacles $U_{obs} = \{obs_1, \dots, obs_M\}$. Each obstacle is a polygon defined by the set of its vertices' coordinates $obs_i = \{p_{i1}, p_{i2}, \dots, p_{ij}, \dots, p_{iK_i}\}$, $p_{ij} = (x_{ij}, y_{ij}) \in U$. Obstacles are additionally characterized by types: $type(obs_i) = ot_j$, $ot_j \in OT$, $OT = \{ot_1, \dots, ot_Z\}$. All agents have similar sizes and can be represented as the circles of radius r in U (see Fig. 1).

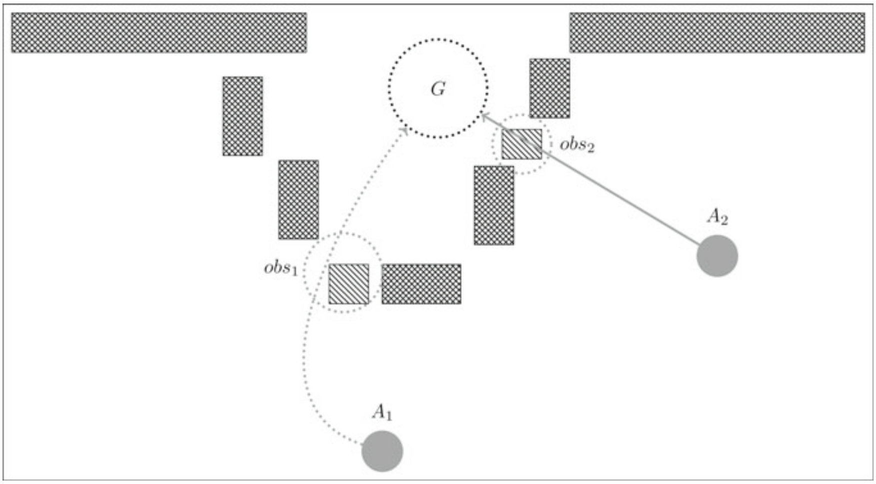


Fig. 1 Considered case of the coalition relocation task: A_1, A_2—members of the coalition, G—the goal area, obs_1, obs_2—obstacles of the ot_1 type (inclined *lines blocks*) destroyable by agent A_2

We suppose that the agent's movement dynamics is encoded as a set of differential equations:

$$\frac{dx}{dt} = f(x, u), \tag{1}$$

where $f(x, u)$—vector function, $x \in R^n$—vector of the phase coordinates, $u \in R^r$—control vector, t—time. Following [46] we assume that given dynamic constraints can be transformed to geometry constraints, e.g., we assume that a feasible trajectory for an agent is the angle-constrained path in U which is a sequence of line segments such that the angle of alteration between two consecutive segments does not exceed predefined threshold α_m.

Agent's knowledge base contains high-level representations of locations and distances as well as the mechanisms of mapping these representations to the workspace. Set of agent's actions are organized in a hierarchical structure and three types of actions exist: transition actions, transforming actions, (which are limited to destroying obstacles of different types) and messaging actions. We consider the case when each agent has its own planning focus containing current beliefs about external objects and processes. Details of the knowledge representation will be described further in Sect. 4.

Single agent's task is reaching the predefined goal area which is the same for all other agents. This common task description for an agent includes explicit constraint that all the agents should reach the goal area (not the only one). We investigate scenarios (as depicted on Fig. 1) when some agents cannot reach the goal area separately, without the assistance from the other members of the coalition. As seen on Fig. 1 agent A_1 cannot reach the goal as it is blocked by the obstacle obs_1 which cannot be destroyed by A_1, while A_2 can alter its plan, reach obs_1 first and destroy this obstacle assisting A_1 in accomplishing the task). We call such tasks—smart relocation tasks.

4 Knowledge Representation

Agent's knowledge base contains descriptions of objects, processes and properties of the external environment and information about other members of the coalition. To formalize the knowledge base we use the semiotic approach wherein all the above entities are mediated by sings. Each sign is composed of a name and three components—image, significance, personal meaning—which are used to implement different functional steps of the planning process. Signs come with the special structured set of links to other sings and to data from inner and external sensors of the agent. We will name these links as features (see Fig. 2).

The first component of a sign is image. Image is the set of structured sets of features specific to the mediated entity. Each structured set of features (shown as square in Fig. 2) corresponds to particular group of characteristic properties of the entity mediated by sign and differs from the other by its structure as well as by the features themselves. At the same time image also implements the process of recognizing the

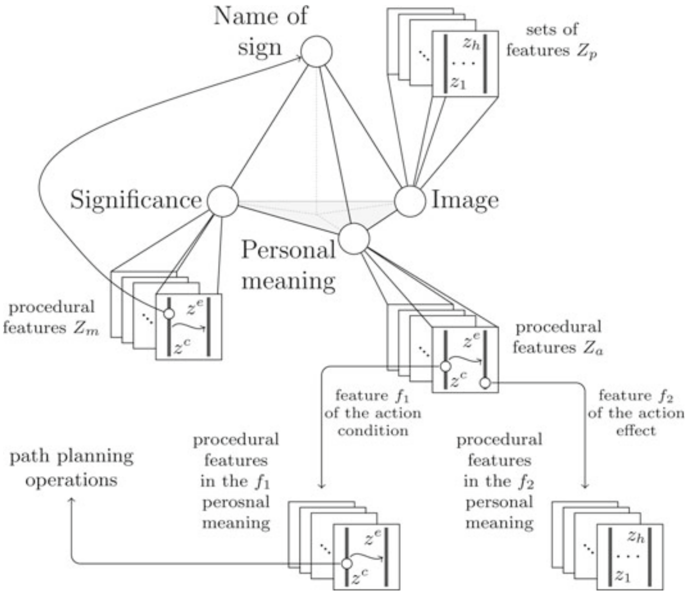

Name of sign
sets of features Z_p
z_h
z_1
Significance
Image
Personal meaning
procedural features Z_m
z^e
z^c
procedural features Z_a
feature f_1 of the action condition
feature f_2 of the action effect
path planning operations
procedural features in the f_1 perosnal meaning
procedural features in the f_2 personal meaning

Fig. 2 Structure of the sign knowledge representation

entity based on the input data. Each set of features encapsulated in image contains those features that are grouped together in the input data stream—shown as columns inside the square in Fig. 2. Each column contains features that together form certain part of the mediated entity description. Considering that features are links to other signs, the hierarchy of signs is formed on the set of images. Lowest level of the hierarchy consists of input data from sensors or information received from path planning operators.

The second component of a sign is significance. Significance is the set of procedural features (or causal relations) and it is used to describe characteristic actions in which mediated entity is engaged. A causal relation consists of the set of conditional features or conditions (encountered before the execution of the action) and the set of resultant features or effects (encountered after the execution of the action). Thus procedural features are the models of AI rules [11]. Additionally, at least one feature (condition or effect) is a link to the sign possessing the significance itself. Considering that features are links to other signs, another hierarchy of signs is formed on the set of significances of mediated actions. Lowest level of the hierarchy contains elementary skills. Significance components of common signs are the same for all agents in the coalition.

Finally the third component of a sign is personal meaning and it also (like significance) describes actions involving the mediated entity. There exist a link between a causal relation of the personal meaning and a causal relation of the significance defined by the function Ξ. Unlike significance, personal meaning contains special type of features—personal features—in its causal relations. These features mediate inner properties of the agent and replace elementary skills of the corresponding significance. Personal meaning implements the process of applying agent's actions. Lowest level of the personal meanings hierarchy is comprised by path planning operators.

The structure of the sign corresponds to psychological models of high cognitive functions [47, 48] and allows to separate generalized representation of actions that are known to all members of the coalition and specific implementation of such actions, which takes into account inner properties of the agent.

The hierarchy of signs (based on images, significances and personal meanings) serves as a tool for the input signal (low-level features) recognition and for the corresponding sign actualization. An algorithm of sign recognition is simple comparison of input features with corresponding set of features predicted by upper level signal on each level of the hierarchy [25]. In this way the recognition process is bottom-up spreading of the activation in the hierarchy of features right down to levels where there is correspondence between features and signs. This algorithm models functioning of the human cortex sensor regions [49, 50]. The set of activated (actualized) signs at the moment represents the agent's believe about the current environment state. Since the hierarchy of signs encodes the set of agent's actions via the procedural features all transitions between states during behavior planning are executed as top-down or bottom-up activation processes in the hierarchy. Low-level procedural features include path planning operators and reaching this level while the activation (planning) process triggers path planning procedures.

5 Behavior Planning Algorithm

On sign level behavior planning is realized in the situation space by *PMA* algorithm proposed in [51]. The situation is defined as the set of signs structured in the same way as the image components (see above), e.g., signs are split into groups which describe different parts of the situation. Use of sign representation allows to combine beliefs about relationships and beliefs about objects and consider all properties, processes and objects in a situation as signs. Transitions between situations are implemented by casual relations contained in procedural features of significances or personal meanings in dependence the planning step. The initial situation is defined as the current observed situation, i.e., the current set of actualized signs. The goal situation is agent's belief about the result of the solving current problem, i.e., it is the set of goal signs.

In relocation tasks low-level features are implemented by path planning algorithms and are considered to be personal features included in agent's personal meanings. Thus the hierarchy of procedural features in fact is the action hierarchy and low-level actions are performed by the subsystem of path planning which lies beyond sign representation.

The algorithm of behavior planning is an iterative process (*PMA*-procedure) consisting of the following steps:

- search of relevant significances (the *M*-step),
- choose a personal meaning from the set of personal meanings corresponding to the found significances (the *A*-step),
- send a message to other members of the coalition (part of the *S*-step),
- perform the action corresponding to the chosen personal meaning (part of the *S*-step),
- construct the new current situation using the set of features from the condition of performed action (the *P*-step).

Input of the *PMA*-procedure is the pair of two situations: start and final situations. On the first iteration of the algorithm the start situation is the current situation (observed by the agent) and the final situation is the goal situation. Then the *M*-step is done, e.g., the search on the significances of signs forming the final situation is performed. Effect parts of each significances are considered and each effect is compared to the set of signs of final situation (such comparison is valid as each effect is a set of features linked to corresponding signs). After the comparison is done such significances are selected which effect features matches maximum number of final signs.

The *A*-step is execution of Ξ procedure which associates the set of procedural features of the selected significances with the procedural features of personal meanings. Then the selection of one procedural feature of personal meaning is done. Transition to the *P*-step occurs. The *P*-step of the algorithm is combining of all features included in conditions of personal meaning features. After *P*-step the new iteration of *PMA*-procedure is executed with the same start situation and the new formed final situation.

If new situation is the subset of start situation (the first argument of *PMA*-procedure) then in some cases *S*-step occurs depending on cognitive qualities (parameters of planning process) of the agent. *S*-step results in defining the new goal situation as a result of the action (rule) application. This application adds to current situation description features from the image component corresponding to procedural features of personal meanings defined on the *A*-step. The activation process spreads top-down in the procedural features hierarchy (see Sect. 4). In some cases activation process reaches the level where path planning operators are included in personal features, so path planning is started. If path planner returns success *PMA*-procedure also ends with *success*—behavior plan including valid path of relocation to the goal area is constructed. If path planner returns *failure* along with the coordinates of blocking obstacle (see below) then the new features (corresponding to the identified obstacle) are added to the description of the current situation and the *PMA*-procedure is repeated.

All changes arising in the current observed situation (for example, emerging unaccounted obstacle detected by path planning process or the new task received from other member of the coalition) triggers the re-execution of *PMA*-procedure with new start or final situations.

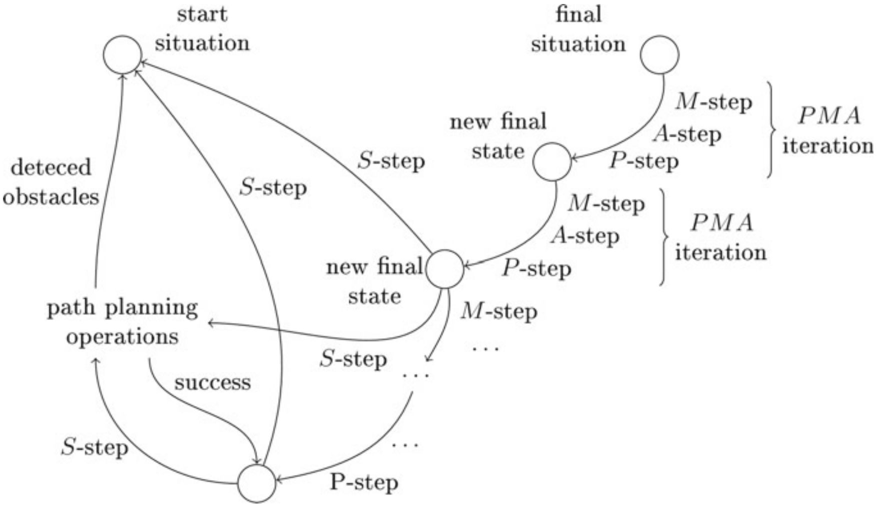


Fig. 3 Schema of behavior planning *PMA*-procedure

With new sign representation of agent's knowledge about environment and its own qualities we can describe and implement meta-cognitive regulation functions of the agent behavior. These functions are realized by rule (mental action) application mediated by personal meanings and significances during the selection of *PMA*-procedure parameters. This regulation process executes some rules that change personal features of the agent implemented parameters of planning and recognition processes.

All members of the coalition have signs that mediate both objects of external environment and other members of the coalition. The significances of these signs include agent's knowledge about actions available for corresponding agents. The personal meanings of these signs include actions by sending them a communication messages. The constructed plan of behavior can contain personal meanings of signs corresponding to other members of the coalition. In this case a message with description of the significance obtained by the inverse procedure Ξ is sent to the corresponding member. This message plays the role of the new task for this agent and triggers its *PMA*-procedure re-execution. Thus the common plan of the task resolving includes all sub-plans and all goal-setting messages of all members of the coalition (Fig. 3).

6 Path Planning Algorithm

We suggest using grids as spatial representations for path planning as they are both informative and easy-to-search graph models of the agents' environment (as described in Sect. 3). Grid is constructed by overlaying regular square mesh over the workspace U in such way that each grid element c, e.g., a cell corresponds to a unique square area in U sized $res \times res$, where res—is an input parameter. If this area overlaps with any obstacle, cell is marked untraversable (traversable—otherwise). We adopt center-based grid notation (in opposition to corner-based) meaning that a path should start (and end) at the center of some grid cell $c(x, y)$ (and thus it is supposed that any agent is tied to the center of some grid cell initially). We also adopt the idea of any-angle path finding [37] and consider the path as the sequence of traversable but not obligatory adjacent cells $\pi = \{c_1, c_2, \dots, c_p\}$, such that an agent can move from one cell to next one in the sequence following the straight line connecting the centers of those cells. Function $los(c_i, c_j) \rightarrow \{true, false\}$ is given to check this condition. If the size of the cell is big enough to accommodate an agent (e.g., $res \geq 2r$) one can use well-known (and fast) Bresenham algorithm [52] to check line-of-sight constraint. This algorithm identifies grid cells forming discrete representation of straight line, so after that, one needs to check if they all are traversable. Occasionally it can happen so that Bresenham algorithm identifies cells that are all traversable, although actual straight line intersect an untraversable cell (and thus, possibly, an obstacle). To avoid this we suggest double-outlining the obstacles in the following way: after the grid is constructed mark all the adjacent cells for each untraversable and then put all the marked cells untraversable—see Fig. 4.

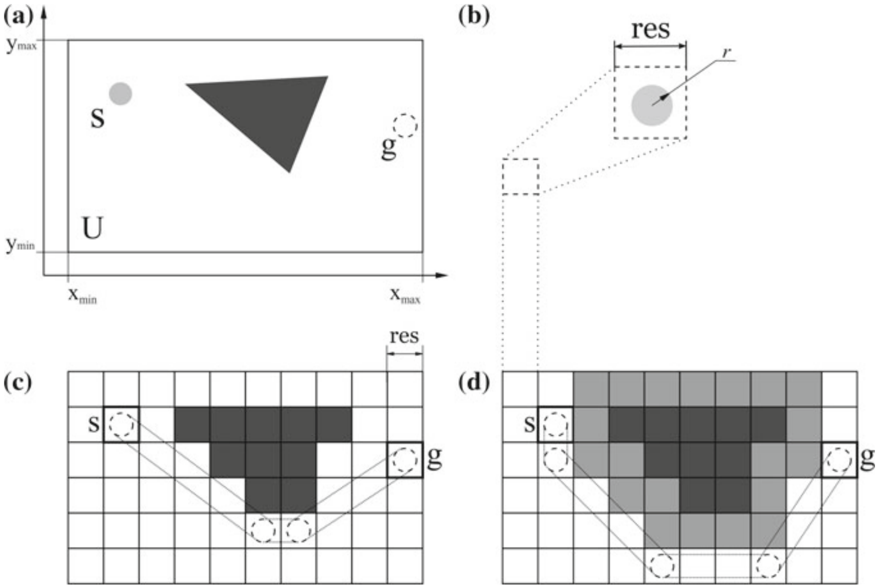


Fig. 4 Grid representation of the workspace. **a** Initial workspace. **b** Square pattern used for discretization. **c** Path on a grid colliding with the untraversable regions. **d** Double-outlining of the obstacles and creating additional untraversable cells prevent from generating paths colliding with true obstacles

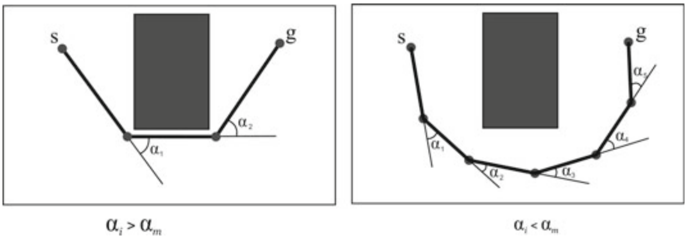


Fig. 5 LIAN algorithm details

We use two algorithms for path finding: Basic Theta* [37] and LIAN [42]. First algorithm searches for any angled path on given grid, second searches for angle-constrained path, e.g., for such a path $\pi = \{c_1, c_2, \ldots, c_p\}$ that an angle of alteration between any consecutive sections $\langle c_{i-1}, c_i \rangle, \langle c_i, c_{i+1} \rangle$ is less than the predefined threshold α_m (see Fig. 5). Searching for the angle-constrained path is much more burdensome but such a path indirectly guarantees its feasibility, e.g., agent's ability to follow the trajectory in U defined by that path without violating the dynamic constraints (as described in Sect. 2).

So we search for any angle path first and if it is found start searching for the angle-constrained path. If it is found, we report success to the upper level, e.g., behavior planning module and wait for the next goal to be given. Failure to find a path means that angle constraint is too strict (in respect to the current environment model, e.g. grid, and start-goal locations) so the new subgoal is to be given by the behavior planning module or the new angle constraint.

An interesting case occurs when any angle path planning returns failure, which means that there is no path to given goal location not due to the angle constraints, but due to the obstacles configuration, e.g., some obstacle is blocking the path (otherwise it would have been found as Theta* is sound and complete). In this case it is useless to ask for a new subgoal as a resultant path will not be found anyway. An obstacle blocking the path should be identified and its coordinates should be transmitted to the behavior planning module. To the best of authors' knowledge currently there are now works on the methods of identifying blocking obstacles, so this is an appealing research area to be investigated further.

Previously, when describing the path planning process we supposed that the goal cell is given, although, in the case sign-world model is used for behavior planning, the fuzzy goal area is under concern. This area is characterized by the point $cp(x, y)$ and radius r_g—if the agent reaches any point of the circle with radius r_g and center in $cp(x, y)$ path planning is considered to be successfully accomplished. This can be taken into account in the following way—execution of the LIAN algorithm should be stopped when any cell of the circle is under consideration. We also perform the consistency of the goal area check before path planning in the following manner: if given center point belongs to the grid's untraversable cell—choose one of its traversable adjacent cells as cp. If all the neighbors are untraversable—examine their neighbors and so on up to the moment traversable cell of the goal area will be identified. If all cells forming the goal area are untraversable—report behavior planner and wait for the new goal area.

7 Behavior and Path Planning: Case Study

Below we demonstrate implementation of the planning method in solving the smart relocation task as described in Sect. 3. We consider a simple case when two agents A_1 and A_2 form the coalition and share common goal area. Fragment of the agents knowledge base (sign model) is depicted in Table 1. Arrows used in description of image component split different columns of features (as described in Sect. 5).

We consider the planning process of the agent A_1 whose goal situation is described with signs “I—agent 1”, “agent 2”, “place X_1”. As said above all of these signs must be activated in knowledge base (sign word model) of agent A_1 which means that both “I agent 1” and “agent 2” should be in “place X_1”. Note that the description of A_2 goal situation is similar (“I—agent 2”, “agent 1”, “place Y_1”). Start situation for agent A_1 is depicted on Fig. 6 and is described by the signs “I—agent 1”, “place X_4” → “agent 2”, “place X_5” (here → is used to split different parts of situation description).

PMA-procedure executes its first iteration with the start situation and goal situation as input arguments. The M-step searches for the maximal effect-covered procedural feature from the set of available significances (“move 1”, “send mes-sage”).

Table 1 The fragment of the sign world model of agents

Specific signs of the A_1 agent	s_1: n=“obstacle 1” p={“place X_6”, “type ot_1”} m={“destroy 1”} $a=\emptyset$	s_2: n=“obstacle 2” p={“place X_2”, “type ot_1”} m={“destroy 1”} $a=\emptyset$	s_3: n=“far” p={mechanical sensor parameters} m={“move 1”} a={“I move 1”}	s_4: n=“agent 2” p={mass, coordinates etc.} m={“send message”} a={“I send message”}
	s_5: n=“I move 1” p={“I move 3”, “I move 3”, “place X_1”} $m=\emptyset$ $a=\emptyset$	s_6: n=“I move 3” p={“I”, “here” → “empty”, “place X_3”} m={“move 3”} a={path planning realization}	s_7: n=“I — agent 1” p={mass, coordinates etc.} $m=\emptyset$ $a=\emptyset$	s_{10}: n=“place X_1” p={“far”, “ahead”, “right”} $m=\emptyset$ $a=\emptyset$
	s_{10}: n=“move 1” p={“I move 3” → “I move 3” → “place X_1”} $m=\emptyset$ $a=\emptyset$	s_{10}: n=“destroy 1” p={“obstacle 1” → “empty”} $m=\emptyset$ $a=\emptyset$	s_{10}: n=“move 3” p={“here” → “empty”, “place X_3”} $m=\emptyset$ $a=\emptyset$	
Specific signs of the A_2 agent	s_1: n=“obstacle 1” p={“place Y_2”, “type ot_1”} m={“destroy 1”} $a=\emptyset$	s_2: n=“obstacle 2” p={“place Y_3”, “type ot_1”} m={“destroy 1”} a={“I destroy 1”}	s_3: n=“far” p={mechanical sensor parameters} m={“move 1”} a={“I move 1”}	s_8: n=“agent 1” p={mass, coordinates etc.} m={“send message”} a={“I send message”}
	s_5: n=“move 1” p={“I”, “here” → “empty”, “place Y_1”} m={“move 1”} a={path planning realization}	s_9: n=“I — agent 2” p={mass, coordinates etc.} $m=\emptyset$ $a=\emptyset$	s_{10}: n=“place Y_1” p={“far”, “ahead”} $m=\emptyset$ $a=\emptyset$	s_{11}: n=“destroy 1” p={“obstacle” → “empty”} $m=\emptyset$ $a=\emptyset$

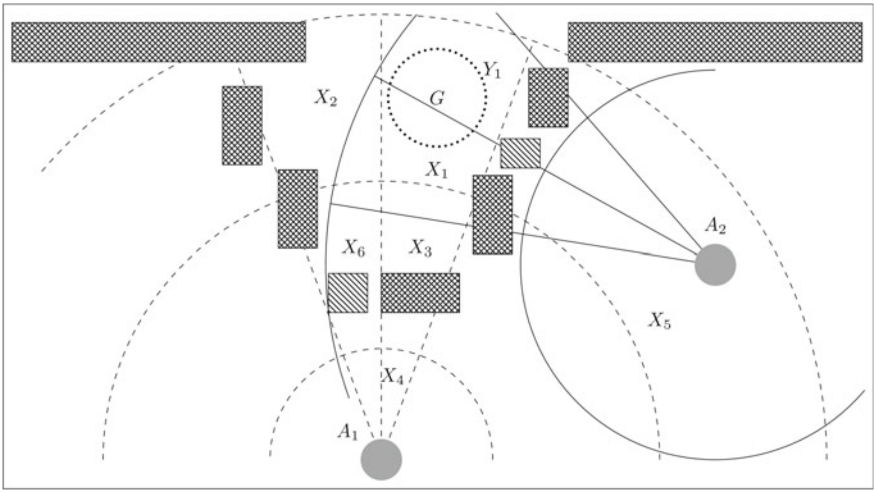


Fig. 6 Representations of space in sign level. X_i—names of places associated with signs for the agent A_1, Y_1—a name of the place associated with the sign for the agent A_2

From the Table 1 we can find that it is significance m = “move 1” (transition ahead and right). Then the A-step is executed and in our case the procedure gives the personal meaning that includes feature corresponding “I move 1”.

Suppose agent A_1 has such a cognitive quality (introspection level) used as input parameter for behavior planner that leads to S-step of PMA procedure. On the S-step activation spreads top-down in procedural features hierarchy and the activation trace (see Table 1) is “I move 1” → “I move 3” → “path planning pro-cedures”). So finally path planning process is invoked. On given map path cannot be found as “obs 1” is blocking the way, so its coordinates are returned back to behavior planner. These coordinates are low-level image features so the im-age-base hierarchy is traced bottom-up until the sign mediating the obstacle will be reached (activated), e.g., “obstacle 1” sign. This sign is added now to the description of start situation: “I—agent 1”, “place X_4” → “agent 2”, “place X_5” → “obstacle 1” and the description of final situation: “I—agent 1”, “agent 2”, “place X_1” → “obstacle 1”, “empty”. The PMA-procedure re-executes on the newly defined start situation. Again at the M-step the search for the maximal effect-covered procedural feature from the set of available significances (“move 1”, “send message”, “destroy 1”) is done and procedural feature “destroying 1” is selected. The A-step fails to fine suitable personal meaning (as agent A_1 has no destroying action). Thus communication action will be selected. So A_1 sends message to the agent A_2 with coordinates of obs1 and description of the action (“destroy 1”), needed to be performed in order for the agent A_1 to achieve the goal (“agent 1”,“place X_1”). The presence of the agent A_1 in the goal area (“agent 1”,”place X_1”) is required for A_2 to reach its goal (“I—agent 2”, “agent 1”, “place Y_1”) so signs “obstacle 1” and “destroy 1” (contained in transferred message) are added to A_2 goal which becomes: “I—agent 2”, “agent 1”, “place Y_1” → “obstacle 1”, "empty". As A_2 has personal meaning "I destroy 1" A-step of the PMA-procedure for A_2 will be accomplished with success and the obstacle will be destroyed. So both of the agents can now construct the plan of goal area achievement and thus accomplish smart relocation task.

8 Conclusion

In this paper we have presented an approach to build a two-layered planner for the robotic system which is a part of a coalition solving common task. We suggest following semiotic approach to develop cognitive top-level planner in order to make the system more versatile, robust and human-like (in contrary to the existing logic based approaches—PDDL languages etc.). We have introduced new knowledge representation formalism—sign world model—which aligns well with the results of recent cognitive and neurophysiologic research. Path planning operators are the integral part of this hierarchical model so task and path planning processes are tied together as a parts of coherent framework. Behavior planning is a recursive search process in the hierarchical state-space induced by sign representation ending up with path planner triggering: when behavior planning reaches the lowest level of sign world model state-of-the-art grid-based planners are executed. We suggest focusing on searching for the special type of smooth paths—angle-constrained paths—in order to indirectly satisfy agent's dynamic constraints and produce feasibly trajectories. We also discuss involving sound and complete any angle path planner in the loop in order to be able to better distinguish between different failure outcomes of the path planning process.

We believe that proposed approach leads not only to extensive flexibility of the planning system which is now capable of solving collaborative navigation tasks which are not solvable by the individual members implementing traditional path planning algorithms, but also will significantly aid solving various human–robot interaction problems. One of worth mentioning tasks which can be positively impacted is natural language formulation of collaborative task to the group (coalition) of robots. Regarding path planning, the task of the blocking areas identification (in case of planner failure) is an appealing direction of further research.

Acknowledgments The reported study was supported by RFBR, research projects No. 14-07-31194 and No. 15-37-20893.

References

1. Albus, J.S.: 4D/RCS: a reference model architecture for intelligent unmanned ground vehicles. AeroSense 2002. International Society for Optics and Photonics (2002)
2. Yoo, J.-K., Kim, J.-H.: Gaze control-based navigation architecture with a situation-specific preference approach for humanoid robots. IEEE Trans. Mechatron. (2015)

3. Emelyanov, S., et al.: Multilayer cognitive architecture for UAV control. Cogn. Syst. Res. **34** (2015)
4. Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory and Practice. Elsevier (2004)
5. Ghallab, M., et al.: PDDL-the planning domain definition language (1998)
6. Fox, M., Long, D.: PDDL2. 1: an extension to PDDL for expressing temporal planning domains. J. Artif. Intell. Res. (JAIR) **20**, 61–124 (2003)
7. Karlsson, L., et al.: Combining task and path planning for a humanoid two-arm robotic system. In: TAMPRA'12: Proceedings of the 2012 ICAPS Workshop on Combining Task and Motion Planning for Real-World Applications, pp. 13–20 (2012)
8. Abdo, N., Kretzschmar, H., Stachniss, C.: From low-level trajectory demonstrations to symbolic actions for planning. In: TAMPRA'12: Proceedings of the 2012 ICAPS Workshop on Combining Task and Motion Planning for Real-World Applications, pp. 29–36
9. Laird, J.: Extending the soar cognitive architecture. In: Proceedings of the First AGI Conference, pp. 224–235 (2008)
10. Laird, J.E.: The Soar Cognitive Architecture. MIT Press, Cambridge (2012)
11. Nilsson, N.J.: Artificial Intelligence: A New Synthesis. Morgan Kaufmann, San Francisco (1998)
12. Langley, P.: Learning to sense selectively in physical domains. In: Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, USA, pp. 217–226 (1997)
13. Langley, P.: Cognitive architectures and general intelligent systems. AI Mag. **27**, 33–44 (2006)
14. Sun, R., Bookman, L.: Computational Architectures Integrating Neural and Symbolic Processes, p. 496. Kluwer Academic Publishers, Boston (1994)
15. Sun, R.: The CLARION Cognitive Architecture: Extending Cognitive Modeling to Social Simulation, p. 434. Cambridge University Press, New York (2006)
16. Fikes, R.E., Nilsson, N.J.: STRIPS: a new approach to the application of theorem proving to problem solving. Artif. Intell. **2**(3–4), 189–208 (1971). http://doi.org/10.1016/0004-3702(71)90010-5
17. Blum, A.L., Frust, M.L.: Fast planning through planning graph analysis. Artif. Intell. **90**(1–2), 281–300 (1997). doi:10.1016/S0004-3702(96)00047-1
18. Hoffmann, J., Nebel, B.: The FF planning system: fast plan generation through heuristic search. J. Artif. Intell. Res. **14**, 253–302 (2001)
19. Helmert, M.: The fast downward planning system. J. Artif. Intell. Res. **26**, 191–246 (2006). doi:10.1613/jair.1705
20. Richter, S., Westphal, M.: The LAMA planner: guiding cost-based anytime planning with landmarks. J. Artif. Intell. Res. **39**, 127–177 (2010). doi:10.1613/jair.2972
21. Della Penna, G., Magazzeni, D., Mercorio, F.: A universal planning system for hybrid domains. Appl. Intell. **36**(4), 932–959 (2012). doi:10.1007/s10489-011-0306-z
22. Fox, M., Long, D.: Modelling mixed discrete-continuous domains for planning. J. Artif. Intell. Res. **27**, 235–297 (2006)
23. Kovacs, D.L.: A multi-agent extension of PDDL3.1. In: Proceedings of the 3rd Workshop on the International Planning Competition (IPC), ICAPS-2012, Atibaia, Brazil, pp. 19–27, 25–29 June 2012
24. Harnad, S.: Symbol Grounding Problem. Physica **42**, 335–346 (1990). doi:10.4249/scholarpedia.2373
25. Osipov, G.S., Panov, A.I., Chudova, N.V.: Behavior control as a function of consciousness. I. World model and goal setting. J. Comput. Syst. Sci. Int. **53**(4), 517–529 (2014). doi:10.1134/S1064230714040121
26. Ivanitsky, A.M.: Information synthesis in key parts of the cerebral cortex as the basis of subjective experience. Neurosci. Behav. Physiol. **27**(4), 414–426 (1997)
27. Edelman, G.M.: Neural Darwin-ism: The Theory Of Neuronal Group Selection. Basic Books, New York (1987)
28. Lozano-Prez, T., Wesley, M.A.: An algorithm for planning collision-free paths among polyhedral obstacles. Commun. ACM **22**(10), 560–570 (1979)

29. Bhattacharya, P., Gavrilova, M.L.: Roadmap-based path planning-using the Voronoi diagram for a clearance-based shortest path. Robot. Autom. Mag. IEEE **15**(2), 58–66 (2008)
30. Kallmann, M.: Navigation queries from triangular meshes. In: Boulic, R., Chrysanthou, Y., Komura, T. (eds.) MIG 2010. LNCS, vol. 6459, pp. 230–241. Springer, Heidelberg (2010)
31. Yap, P.: Grid-based path-finding. In: Co-hen, R., Spencer, B. (eds.) Canadian AI 2002. LNCS (LNAI), vol. 2338, pp. 44–55. Springer, Heidelberg (2002)
32. Wooden, D.T.: Graph-based path planning for mobile robots. PhD thesis, Georgia Institute of Technology (2006)
33. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. Computer **22**(6), 46–57 (1989)
34. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik **1**(1), 269–271 (1959)
35. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. **4**(2), 100–107 (1968)
36. Likhachev, M., Stentz, A.: R* search. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence. AAAI Press, Menlo Park (2008)
37. Nash, A., Daniel, K., Koenig, S., Felner, A. 2007. Theta*: any-angle path planning on grids. In: Proceedings of the National Conference on Artificial Intelligence, vol. 22, No. 2, pp. 1177. AAAI Press, Menlo Park, Calif
38. Harabor, D., Grastien, A.: Online graph pruning for pathfinding on grid maps. In: AAAI-11 (2013)
39. Koenig, S., Likhachev, M.: D*Lite. In: Proceedings of the National Conference on Artificial Intelligence AAAI (2000)
40. Kuwata, Y., Karaman, S., Teo, J., Frazzoli, E., How, J.P., Fiore, G.: Real-time motion planning with applications to autonomous urban driving. IEEE Trans. Control Syst. Technol. **17**(5), 1105–1118 (2009)
41. Kim, H., Kim, D., Shin, J.U., Kim, H., Myung, H.: Angular rate-constrained path planning algorithm for unmanned surface vehicles. Ocean Eng. **84**, 37–44 (2014)
42. Yakovlev, K., Baskin, E., Hramoin, I.: Grid-based angle-constrained path planning. In: Proceeding of the 38th German Conference on Artificial Intelligence (KI-2015) (2015)
43. Standley, T.: Finding optimal solutions to cooperative pathfinding problems. In: AAAI, pp. 173–178 (2010)
44. Wang, K.-H.C., Botea, A.: Fast and memory-efficient multi-agent pathfinding. In: Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS), pp. 380–387 (2008)
45. Silver, D.: AI Program. Wisdom **3**, 99–111 (2006)
46. Yakovlev, K., Makarov, D., Baskin, E.: Automatic path planning for an unmanned drone with constrained flight dynamics. Sci. Tech. Inf. Process. **5** (2015)
47. Leontyev, A.N.: The Development of Mind. Erythros Press and Media, Kettering (2009)
48. Vygotsky, L.S.: Thought and Language. MIT Press (1986)
49. Mountcastle, V.B.: Perceptual Neuroscience. The Cerebral Cortex. Harvard University Press, Cambridge (1998)
50. George, D., Hawkins, J.: PLoS Comput. Biol. **5**(10), 1–26 (2009)
51. Osipov, G.S., Panov, A.I., Chudova, N.V.: Behavior control as a function of consciousness. II. Synthesis of a behavior plan. J. Comput. Syst. Sci. Int. **54**(5) (2015)
52. Bresenham, J.E.: Algorithm for computer control of a digital plotter. IBM Syst. J. **4**(1), 25–30 (1965)

Trajectory Generation Using RNN with Context Information for Mobile Robots

You-Min Lee and Jong-Hwan Kim

Abstract Intelligent behaviors generally mean actions showing their objectives and proper sequences. For robot, to complete a given task properly, an intelligent computational model is necessary. Recurrent Neural Network (RNN) is one of the plausible computational models because the RNN can learn from previous experiences and memorize those experiences represented by inner state within the RNN. There are other computational models like hidden Markov model (HMM) and Support Vector Machine, but they are absent of continuity and inner state. In this paper, we tested several intelligent capabilities of the RNN, especially for memorization and generalization even under kidnapped situations, by simulating mobile robot in the experiments.

1 Introduction

In the robotics and artificial intelligent society, how to implement powerful computational model like human brain has been a big issue. The artificial neural network (ANN) has been studied at least 50 years for the purpose of making intelligent system comparable to brain [1–3]. Recently, among those ANNs related models, deep learning has shown amazing performance in the visual and voice recognition problems [4, 5]. Those recent successes of the ANN seem to give promising futures for realization of a real intelligent system. For the behavioral level, however, a conventional feed forward neural network (FFNN) model is not suitable for generation of intelligent actions because the FFNN has no context node memorizing previous action sequences. The RNN is a neural network model with feedback connections. Because of the feedback connections, the RNN can memorize actions

Y.-M. Lee (✉) · J.-H. Kim
School of Electrical Engineering, KAIST, 291 Daehak-ro, Yuseong-gu,
Daejeon 305-701, Republic of Korea
e-mail: ymlee@rit.kaist.ac.kr

J.-H. Kim
e-mail: johkim@rit.kaist.ac.kr

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_2

it experienced, which can lead to generation of intelligent actions. Moreover, the RNN has its own advantageous in generalization capability compared to other behavioral computational models.

In this paper, computational ability of the RNN is investigated. The purpose of the task is to reach the several goal targets and way points with a proper sequence and trajectory and the robot used in this paper is a differential wheeled mobile robot. In the task, there are two goal targets, one is a red color point and the other is a green color point. In the training phase, the mobile robot moves to the red point first and after closer to the red point to some allowable error, turns and moves to the green point. To succeed this task, context information should be memorized by the RNN. The positions of robot starting and target points were generated randomly from 0 to 1 on the normalized x-y plane. In the test phase, using obtained training data, the relation between current goal positions and desired robot movement is learned. All the process related to experiments is computer simulation programed by the MATLAB. In the following sections, RNN simulation results and some analysis are presented.

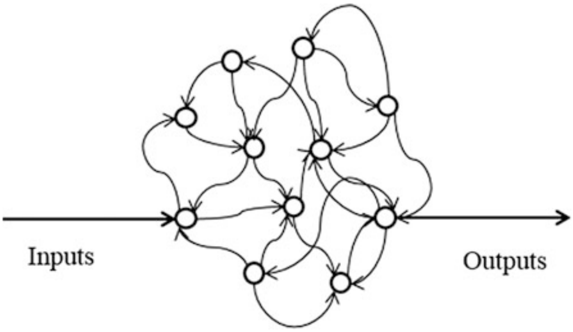
2 Artificial Neural Networks

2.1 The Recurrent Neural Network

The RNN is an artificial intelligence computational model. In contrast to the FFNN, the RNN has feedback connections by which temporal memory can be stored. In the robotics, this RNN model has been used for generating variety of robot trajectories such as mobile robot and robot manipulators [6, 7] (Fig. 1).

In general, the RNN can be categorized into two models, one is a continuous time RNN (CTRNN) and the other is a discrete time RNN (DTRNN). In this paper, the DTRNN is used for our robot simulation experiments.

Fig. 1 Recurrent neural network



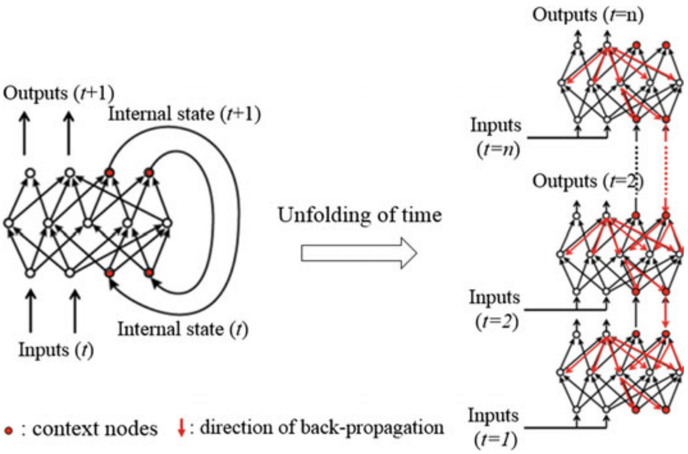


Fig. 2 Cascaded recurrent neural network and back propagation through time

2.2 Training Algorithm

There are various algorithms for training the RNN such as BPTT (Back Propagation-Through Time), EKF (Extended Kalman Filtering) algorithm and optimization training [8, 9]. In contrast to the FFNN, its training algorithm in most case is conventional BP (Back Propagation) algorithm; there is no clear winner in training the RNN. In this paper, the BPTT algorithm is used for training the RNN. The BP algorithm was invented for the purpose of training the conventional FFNN and the learning rule behind the BP algorithm utilizes a gradient descent method. The BP algorithm cannot be applied directly to the RNN because the RNN has feedback loops. To apply the BP algorithm to the RNN, it is needed to cascade the RNN through its training time. Figure 2 shows cascaded RNN architecture.

Since all connections inside the RNN is cascaded through time, there exist no closing loops and the conventional BP algorithm can be applied.

2.3 Mobile Robot Kinematics

In this section, mobile robot kinematics for moving to a target point is described. The mobile robot used in this paper is differential wheeled robot. For given velocity ν and angular velocity ω, next position and heading angle of the mobile robot can be calculated by equation below:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1}$$

where, θ is the heading angle value of the mobile robot with respect to the x axis.

2.4 Feed Forward Neural Network for Mobile Robot

Before conducting main experiments with multiple way points by RNN, the FFNN is tested for a mobile robot to get to a single goal point. The neural network has two input nodes and two output nodes. Two input nodes consist of the current angle value between the robot and a goal and the distance value from the robot to a goal position. Two output nodes consist of the velocity and angular velocity values for the next movement. Also the network has two hidden layer each with 40 nodes. The training data was collected through simulation in which the mobile robot reached the goal position according to the following equation:

$$v = k_1 d \quad \omega = k_2 \theta \tag{2}$$

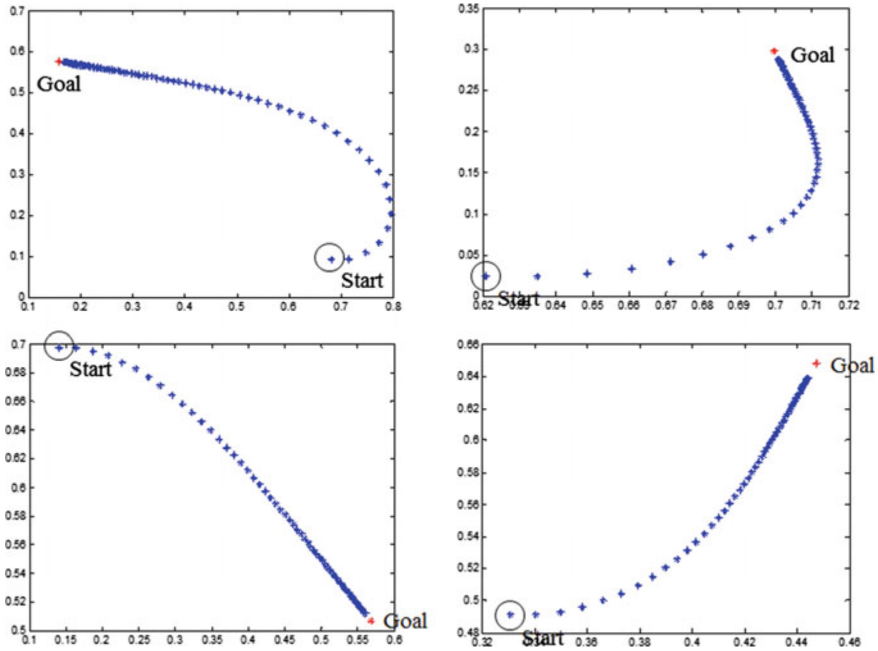


Fig. 3 Mobile robot trajectories when there is only one goal positions

where constant k_i were 0.05 and 0.2, respectively. Robot starting and goal positions were generated randomly on the x-y plane from 0 to 1. The simulation results are plotted in Fig. 3.

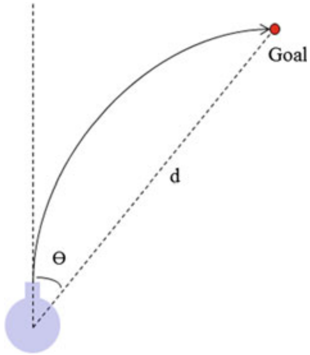
As shown in Fig. 3, the mobile robot could reach the goal position with proper sequences. Definitely, however, this simple task does not require the robot to be aware of the context information. The FFNN's output values solely depend on the current input values. Therefore, when there are more than two goals, the FFNN gets confused whether or not it has already reached the goal with higher priority.

3 Implementation

3.1 Architecture

To have the mobile robot to reach several target positions in a proper sequence, the RNN is required. In this paper, the RNN has four input nodes and two output nodes. Four input nodes consist of the current heading angle values respectively between the mobile robot and the goals 1 and 2 and the distance values respectively from the mobile robot to the goals 1 and 2. Two output nodes consist of the velocity and angular velocity values for the next movement. The network has one hidden layer with 50 nodes. Also the network has 10 context node with feedback connections. Activation function used in the network node is a sigmoid function by which only one directional rotation is permitted to the mobile robot because the sigmoid function can generate a positive value only. In the training phase, initially all weight values exist of the RNN were set to random based on the Gaussian distribution. And the learning rate for training the network was fixed to 0.1 during the entire learning procedure (Figs. 4 and 5).

Fig. 4 Angle and distance between the robot and goal position



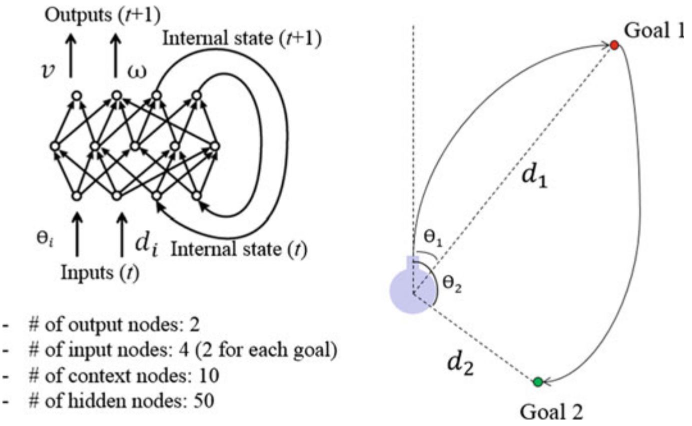


Fig. 5 The recurrent neural network architecture for the mobile robot

4 Experiments

After training the RNN, in the test phase, starting and target point positions were generated with the same method used in collecting the training data. Total training time was about 5 h by the MATLAB programming simulator. The test results are plotted in Fig. 6.

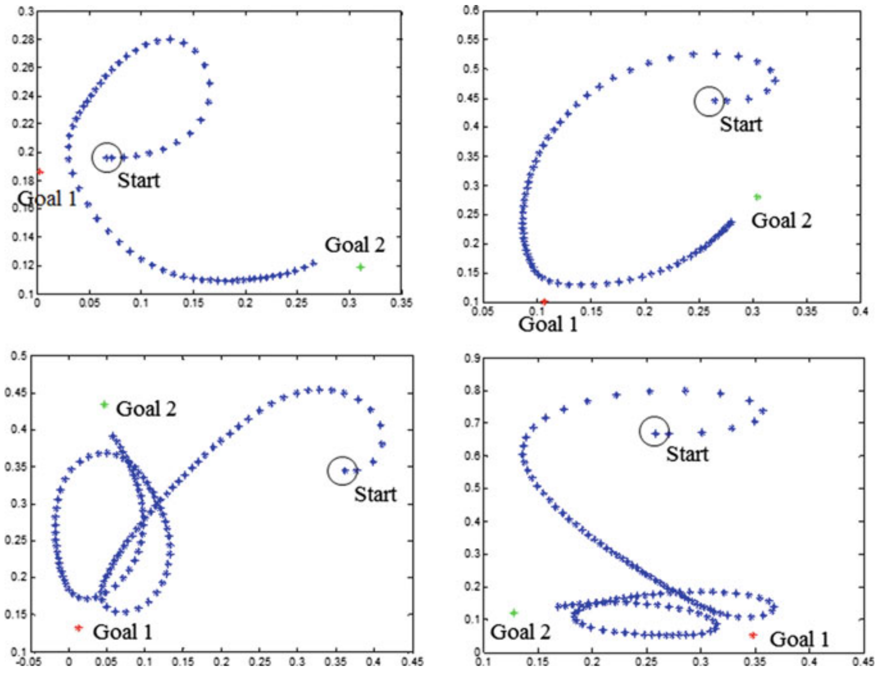


Fig. 6 Mobile robot trajectory when there are two goal targets

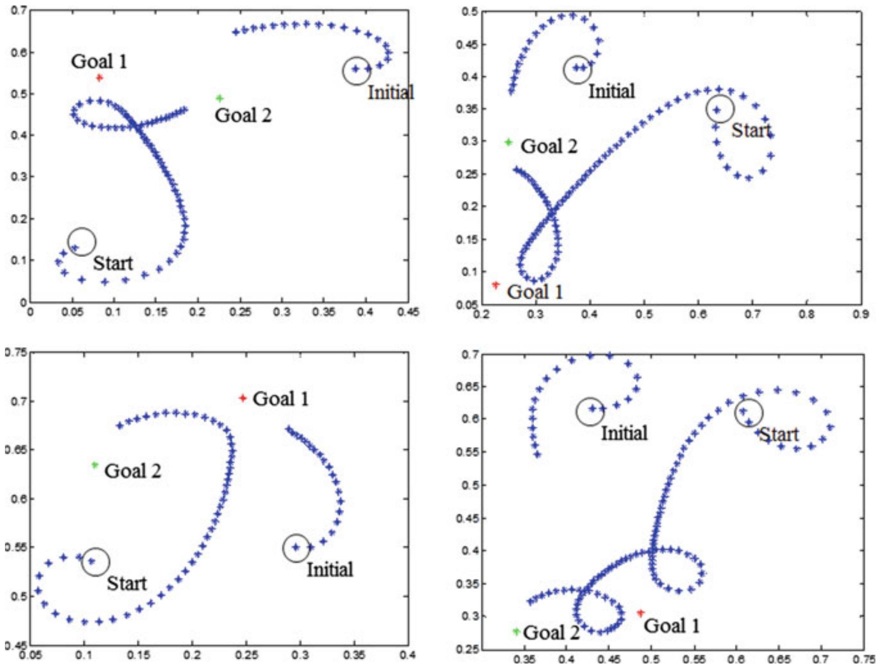


Fig. 7 Mobile robot trajectory when external disturbance exist

In the experimental results, the mobile robot always moved to goal points with proper sequences. In the figure, ‘Start’ is the starting position of the mobile robot and ‘Goal 1’ and ‘Goal 2’ are positions of goals 1 and 2, respectively. In Fig. 6, some trajectories showing strange morphology like circling and that is because in this simulation, mobile robot was permitted to rotate only one direction. Since the robot and goal positions are generated randomly in the training and test phases, we can conclude that the RNN properly generates not only given training trajectories but also unlearned trajectories, therefore guarantees generalization capability of the RNN. To prove robustness of the RNN, in an additional experiment, external disturbance was given to kidnap the robot to other places. After 20 steps moving from the initial stating position, the mobile robot’s position was moved abruptly to randomly selected position on the x-y plane. The results is plotted in Fig. 7.

In the figure, ‘Initial’ is the initial position of the mobile robot and ‘Start’ is the starting position of the mobile robot after kidnapping. As shown in the figure, even though the mobile robot was kidnapped, it could generate a proper trajectory. Finally, the change of the context node activation value is plotted in Fig. 8 according to the mobile robot’s trajectory. We found that only one context node out of 10 context nodes was changed through time.

In this figure, we can find some regularity. If the robot gets closer to the goal 1, the context node value gets increased. And after turning to goal 2, the context node

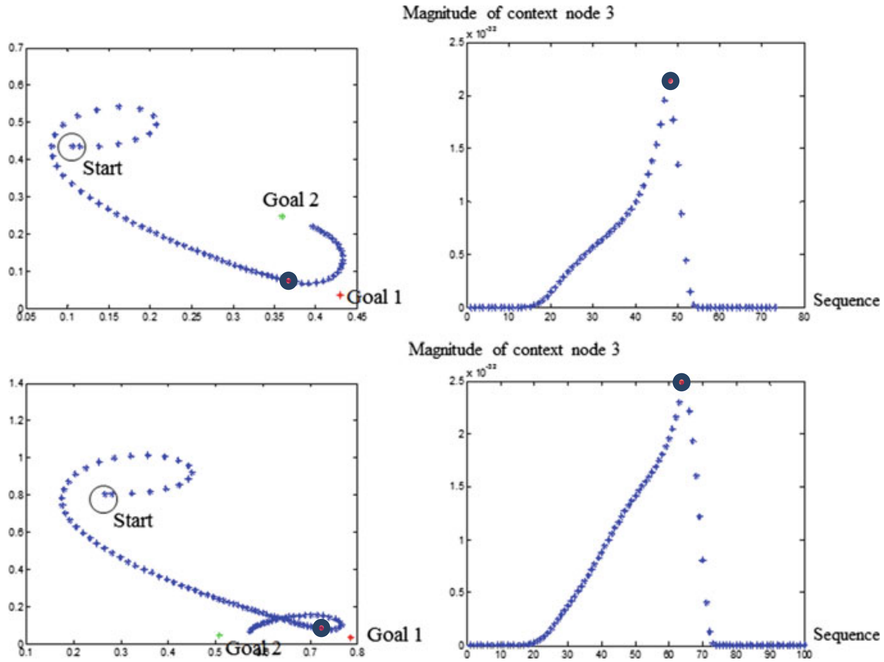


Fig. 8 Change of context node 3

value gets decreased. The turning point is indicated on Fig. 8 using big circle containing red point on the center of it. It is not enough to conclude from Fig. 8 that the context layer has a memory in which a clear switching point exists. However, it looks like that the mobile robot has strong tendency to reach the red point first.

5 Conclusion

In this paper, we tested various capability of the RNN through mobile robot simulation. From the fact that the mobile robot could generate proper sequences even with the external disturbance, we confirmed that the RNN was robust and reliable behavioral computational model. Such kind of characteristics of the RNN may be extended to more complex task such as object manipulation by using humanoid robot arms and walking pattern generator for humanoid robots.

Acknowledgment This work was supported by a grant from the National Research Foundation of Korea (NRF) funded by the Korea government (MSIP) (No. NRF-2014R1A2A1A10051551).

References

1. Hopfield, J.: Neural networks and physical systems with emergent collective computational abilities. In: Proceedings of the National Academy of Sciences, vol. 79, USA (1982)
2. Kohonen, T.: The self-organizing map. Proc. IEEE **78**(9) (1990)
3. Rojas, R.: Unsupervised learning and clustering algorithms. In: Neural Networks. Springer, Berlin, pp. 99–121 (1996)
4. Bengio, Y.: Learning deep architectures for AI. Found. Trends® Mach. Learn. **2(**1) 1–127 (2009)
5. Hinton, G.E.: Deep belief networks. Scholarpedia **4(**5) 5947 (2009)
6. Yokoya, R., et al.: Experience-based imitation using rnnpb. Adv. Robot. **21(**12) 1351–1367 (2007)
7. Sugita, Y., Tani, J.: Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. Adapt. Behav. **13(**1) 33–52 (2005)
8. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Cogn. Model. (1988)
9. Hinton, G.: A practical guide to training restricted Boltzmann machines. Momentum **9(**1) 926 (2010)

Fast and Smooth Replanning for Navigation in Partially Unknown Terrain: The Hybrid Fuzzy-D*lite Algorithm

N.H. Reyes, A.L.C. Barczak, T. Susnjak and A. Jordan

Abstract This paper presents a hybrid Fuzzy-D*lite algorithm for smoothly navigating robots in an unknown terrain, in real-time. D*lite is a clever optimal, incremental and heuristic search algorithm that is known to be capable of achieving a speed up of one to two orders of magnitude over repeated A* searches. Given a target destination and an incomplete map, it is able to generate a sequence of way-points for a robot, on the fly, performing course corrections whenever necessary, at a reduced computational time and memory footprint due to its incremental search capability. On the other hand, a cascade of fuzzy systems designed to take advantage of symmetry in the problem domain implements target pursuit and stationary spinning behaviours, for a two-wheeled robot. These reactionary systems calculate the exact steering angle and speed adjustments, enabling the robot to navigate smoothly and fast. We demonstrate how these complementary algorithms can be fused together to achieve smooth and fast continuous re-planning actions in a partially unknown terrain.

1 Introduction

Imagine sending a robot in an unchartered territory, on a mission to find a fixed goal position. Equipped with a rough map of the exploratory space and on-board sensors for detecting obstacles within its immediate vicinity, it proceeds with goal-directed path-planning by first making some assumptions about the unknown areas in the map; that is, that they are all traversable. For every step of the way, it consults the map to find an approximate heading angle, and using its on-board sensors, it verifies the safety of the immediate terrain. If it learns that parts of the map are wrong, or if it learns that the current path it is following leads to a dead end, the main question is, how can it perform course corrections efficiently using its current knowledge of the world? This calls for an efficient solution that can build knowledge as it traverses

N.H. Reyes (✉) · A.L.C. Barczak · T. Susnjak · A. Jordan
Massey University, Albany, New Zealand
e-mail: n.h.reyes@massey.ac.nz

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_3

the world, and make use of previous path-planning results to improve its future re-planning actions. Traditional search algorithms that guarantee optimality, such as uniform cost search and the A* algorithm [1] will not do well for this class of problems as they lack the capability of performing incremental search [2–5]. If the robot employs A*, every time the robot learns that there is a correction to be made in the map, A* will have to re-calculate for the shortest path from scratch. In general, given a huge gridworld (discretized) representation of the world, traditional path-planning algorithms will require a tremendous amount of work during continuous planning in a dynamically changing world, or if there are too many unknown regions and corrections in the map.

Navigating a robot also requires a real-time algorithm to compute for the exact steering angle and speed adjustment to achieve a smooth and fast trajectory towards the goal. As the robot pursues a target, it may require taking a quick sharp turn, while moving forward to avoid obstacles. Additionally, it may require making a complete U-turn at a stationary point, at the sight of a dead end. Moreover, changing the steering angle of the robot requires actually controlling its wheel velocities, or executing a sequence of leg maneuvers to move the robot in the right direction, while maintaining stability.

In this work, we combined an optimal, incremental and informed search algorithm, called D*lite, together with a cascade of fast reactionary fuzzy logic systems for navigating a 2-wheeled robot in a partially unknown terrain. Our design takes advantage of the presence of symmetry in the problem domain to cut down significantly on the number of fuzzy control rules; thereby, also cutting down the computational costs. The paper is structured as follows: first, a brief literature review discusses the challenges of re-planning in an unknown terrain, within the scope of this work. Thereafter, the subsequent sections describe the details of the proposed hybrid method and the experiments performed to verify the efficiency of the combined approaches. A discussion of the results is then provided, followed by the conclusions.

2 Related Work

Efficient solutions to continuous planning or re-planning problems that can be found in the literature [2, 3, 5, 6] commonly operate on graphs, and are generalizable to solving both routing and symbolic planning problems. As a necessary precursor to utilizing these algorithms for robot path-planning tasks, the exploratory environment needs to be transformed first into its discrete gridworld representation. Cell connectivity in the graph is dictated upon by the set of allowable robot maneuvers. For instance, a robot may be limited to taking only four possible actions (i.e. move north, south, east or west), hence requiring a 4-connected gridworld, or the robot may be agile enough to move along diagonals, requiring an 8-connected gridworld representation of the exploratory domain. The pioneering works of Stentz on Focussed Dynamic A* [3] gave rise to much attention in the field, as he was able to combine techniques from algorithm theory with artificial intelligence. A heuristic function is used to guide an incremental search algorithm that is able to efficiently re-use information from previous search results. The efficacy of the algorithm was demonstrated in solving autonomously unstructured outdoor navigation, with or without an a priori map, by integrating it into Mars Rover prototypes and tactical mobile robot prototypes for urban reconnaissance. Experiments show that for environments requiring a million cells gridworld representation, D* can run up to 200 times faster than A* [7]. D* offers an efficient, complete and optimal solution; however, the algorithm is difficult to extend and maintain. Its core requires nested if statements with complex conditions. Furthermore, the algorithm involves complex tie-breaking criterion when comparing priorities during planning.

Koenig et al. took the initiative of developing a simplified algorithm that implements the same navigation strategy of D*. As a result, they developed the Life-long Planning A* (LPA*) algorithm that repeatedly determines shortest paths between a start and goal position, as the edge costs of a graph change [5]. Subsequently, they extended LPA* to work on goal-directed navigation in unknown terrain, and called their algorithm D*lite [2]. D*lite proved to be more efficient than D* [2]. It is easier to extend, maintain and analyse, and is suitable also to mapping an unknown terrain [8]. One extension to D*lite solves the moving target problem during re-planning [9]. Another extension uses linear interpolation techniques to calculate accurate path cost estimates for arbitrary positions within each grid cell—Field D*, an algorithm that is able to improve on the quality of the paths generated, producing a range of continuous headings. Remarkably, Field D* was used by NASA’s Jet Propulsion Laboratory on the Mars rovers “Spirit” and “Opportunity” on Mars [10].

3 Motivations

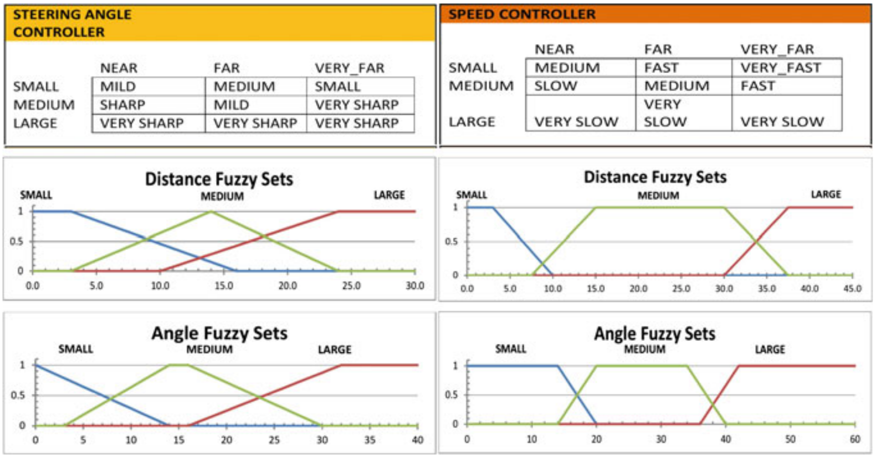
Contrary to existing research that aims to improve the quality of the path generated by grid-based path-planners using interpolations and multi-resolution path-planning [10], this work capitalizes on the strength of fuzzy logic in solving highly non-linear problems, while generating fast and smooth control outputs. This research draws its inspiration from the success of previous works on hybrid path-planning techniques [11, 12]. In comparison to the work in [11]; here, we address the problem of navigation in an unknown terrain. Moreover, a fuzzy spinning motion generator that works with fuzzy target pursuit is introduced in this work, allowing the robot to take U-turns during pursuit. The works in [12] also combined D*lite with fuzzy logic, for a multi-objective mission flight planning. They combined D*lite and fuzzy logic together, but for different reasons and their integration architecture is totally different. They used a multi-objective fuzzy system as an internal component of D*lite, for customizing the calculation of the rhs-values. In this work, D*lite is used to guide the fuzzy controllers, and the outputs of the system directly control the motors of the robot (wheel velocities); thus enabling both steering angle adjustments and absolute speed control. Last but not least, our novel design strategy is able to split the input feature space into half, whenever symmetry can be found.

4 The Hybrid Fuzzy-D*lite Algorithm

For brevity of discussion, this section takes a top-down approach on presenting the details of the hybrid algorithm. Algorithm 1 utilizes Algorithm 2, while Algorithm 2 relies on two fuzzy systems defined in Fig. 1.

Algorithm 1 specifies the major components that comprise the hybrid path-planning algorithm and the synergy between them. We use the D*lite algorithm described in [6] as a global path-planner that guides a cascade of fuzzy control systems implementing fuzzy spin and target pursuit behaviours. For every fraction of a second, the fusion of the algorithms generate an exact steering angle adjustment and absolute robot speed, in accord with the environmental conditions. Furthermore, these values are transformed into left and right wheel velocities to navigate the robot, as it learns more about the environment.

First, given an incomplete map of the world, the robot's current position, its heading angle and a target destination, using D*lite and an initial optimistic assumption that all unknown cells are traversable, the algorithm proceeds by calculating the shortest path to the goal. This is done by executing steps 21–23 of the D*lite algorithm(second version) defined in [6]. It is guaranteed that D*lite will find the shortest path, if a path exists; otherwise, the robot stops if no finite path to the goal is found (g-value of Start cell is ∞). Our implementation of D*lite varies slightly with the one defined in [6], in that we do not check if an adjacent cell is a successor or predecessor in steps 7, 17, 20 and 26 of the second version of D*lite [6], but instead, we simply look at all neighbours of a cell. Despite these modifications, when tested, our D*lite experiment results completely matched the planning and re-planning simulations (D*lite, second version) shown in [6]. In this work, we have tested two heuristics: the Manhattan distance equivalent for 8-connected gridworlds



STEERING ANGLE CONTROLLER

	NEAR	FAR	VERY_FAR
SMALL	MILD	MEDIUM	SMALL
MEDIUM	SHARP	MILD	VERY SHARP
LARGE	VERY SHARP	VERY SHARP	VERY SHARP

SPEED CONTROLLER

	NEAR	FAR	VERY_FAR
SMALL	MEDIUM	FAST	VERY_FAST
MEDIUM	SLOW	MEDIUM	FAST
LARGE	VERY SLOW	VERY SLOW	VERY SLOW

Fig. 1 Details of the individual fuzzy logic control systems(FLS): fuzzy rules and fuzzy sets. *Left hand side* FLS for steering angle adjustment. *Right hand side* FLS for absolute speed computations

(which approximates the distance from the start cell to a cell with the maximum of the absolute differences of the x and y coordinates of both cells [6]), and the Euclidean distance. When using the Euclidean distance, we set the distance between diagonally adjacent cells to be $\sqrt{2}$ distance units; otherwise, we set the distance between two adjacent cells to 1. It is our finding that only the Euclidean distance gives us the shortest path to the goal, eliminating cases where the algorithm generates unnecessary crooked paths, as is the case when using the 8-connected Manhattan (since diagonally adjacent cells are treated of equal distance to horizontally and vertically adjacent cells = 1 distance unit).

Next, once D*lite finds the shortest path from Start to the Goal, the algorithm enters a loop (step 6–18) and engages the cascade of fuzzy logic systems. This loop runs until the goal is finally reached by the robot. For each step of the way, D*lite delivers the next waypoint (step 9), which then becomes the intermediary target (step 11–12) for the `FuzzySpin_and_TargetPursuit()` systems. The cascade of fuzzy systems return the exact left and right wheel velocities, but these values are refined further using a speedMultiplier calculated based on the current angleFromTarget (step 14–15) (`calculateSpeedMultiplier()` is implemented as a small look-up table with only three entries). Finally in step 16, the robot scans its immediate vicinity to check for any changes in the environment, or any cells that were previously unknown in its map. If changes are detected, efficient re-planning is performed. This is done using `scanGraphForChangedEdgeCostsAndUpdate()` which corresponds to steps 28–35 of [6].

Algorithm 2 details the combined fuzzy spin and target pursuit behaviours. The algorithm ties up with two fuzzy controllers defined in Fig. 1, whose architectural design takes advantage of the presence of symmetry in the feature space. Note that the fuzzy rule outputs do not specify exactly whether or not to steer the robot to the right or to the left. The output only specifies the magnitude of the steering action. The beauty of this design is that we are only solving for fuzzy sets covering half of the feature space. The results do not sacrifice accuracy, while significantly cutting down on the cost of computations. As can be seen, steps 14–21 (handling the case when the target on the right) is mirrored in steps 22–30 (target on left), except for the reversal of addition and subtraction operations. Similarly, steps 33–38 is mirrored in steps 39–45, but switching the left and right wheel velocities.

Given the angleFromTarget, distanceFromTarget and targetPosition (position of the target, relative to the heading angle of the robot), and the two fuzzy controllers (Fig. 1), the algorithm calculates for the absolute robot speed and steering angle (steps 1–9). These values are then translated into left and right wheel velocities in steps 10–46.

Steps 10–13 moves a robot straight if the target is directly in front of it. On the other hand, steps 10–31 accounts for target pursuit. The pursuit behaviour allows a robot to move towards a target, while making a turn and adjusting its speed. Lastly, steps 32–46 accounts for the fuzzy spin behaviour which overrides target pursuit whenever the angleFromTarget is found bigger than min_spin_angle. This is important, especially for making a U-turn whenever a dead end is encountered during its traversal.

5 Experiment Results and Discussion

A 2-wheeled robot simulation system using Box2D as the physics engine was developed in our laboratory to perform the experiments.

5.1 Calibration Map: Combined Fuzzy Spin and Target Pursuit

First, we calibrated the fuzzy controllers defined in Fig. 1, in connection with Algorithm 2 to ensure that the target pursuit behaviour is smoothly executed, and that the spinning behaviour is engaged every time the angleFromTarget exceeds min_spin_angle (20 degrees). The calibration results are reflected in Fig. 2, which shows several snap shots of the robot's movement, at different time sequences, for a wide range of positions and angles. As can be observed, robots with bigger angleFromTarget spin first, before moving toward the target. On the other hand, robots with smaller angleFromTarget move earlier toward the target.

```
Given: $M$, incomplete map of world
    $S$, Start position
    $G$, Goal position
    $R$, current robot position
    $robotHeadingAngle$
    $D * lite$ algorithm
    Combined fuzzy logic control systems: $FuzzySpin\_and\_TargetPursuit$
Output: left wheel velocity $WV_{left}$, right wheel velocity $WV_{right}$
$S = R$
$path = D^{*}lite \rightarrow computeShortestPath(S, G, M)$
if (no $path$ was found) then
    return
end if
while (goal has not yet been reached) do
    if ($path$ has been found) and ($calcDistanceFromTarget(R, G) > minTargetDistance$) then
        if (the distance to the next waypoint is less than $min\_distance\_waypoint$) and (goal position has not been reached yet) then
            $next\_waypoint = D * lite \rightarrow findNextWayPointTowardsGoal()$
        end if
        (angleFromTarget, targetPosition) = calcAngleFromTarget(robot heading angle, R, $next\_waypoint$)
        distanceFromTarget = calcDistanceFromTarget(R, $next\_waypoint$)
        $(WV_{left}, WV_{right}) = FuzzySpin\_and\_TargetPursuit$(angleFromTarget, distanceFromTarget, targetPosition)
        $speedMultiplier$ = calculateSpeedMultiplier(angleFromTarget)
        R = move($WV_{left} * speedMultiplier, WV_{right} * speedMultiplier$)
        $D^{*}lite \rightarrow scanGraphForChangedEdgeCostsAndUpdate()$ {scan the environment using on-board sensors, if any edge costs have changed, perform re-planning}
    end if
end while
```

Algorithm 1: Hybrid Fuzzy-D^{*}lite Algorithm

```
Given: angleFromTarget, distanceFromTarget, targetPosition, fuzzySpeed, fuzzyTurningAngle, tempVelocity
    {temporary variable}
    Fuzzy logic control system: $FLS_{pursuit\,Angle}$
    Fuzzy logic control system: $FLS_{pursuit\,Speed}$
Output: leftWheelVelocity, rightWheelVelocity
$FLS_{pursuit\,Speed} \rightarrow calculatePursuitSpeed(angleFromTarget, distanceFromTarget)$
$fuzzySpeed = FLS_{pursuit\,Speed} \rightarrow getCrispOutput()$
if $(angleFromTarget < minAngle)$ then
    $FLS_{pursuit\,Angle} \rightarrow setSteeringCoefficient(0.2)$ {coefficient for the rule outputs}
else
    $FLS_{pursuit\,Angle} \rightarrow setSteeringCoefficient(0.35)$ {coefficient for the rule outputs}
end if
$FLS_{pursuit\,Angle} \rightarrow calculateTurningAngle(angleFromTarget, distanceFromTarget)$
$fuzzyTurningAngle = FLS_{pursuit\,Angle} \rightarrow getCrispOutput()$
if $(angleFromTarget < very\_smallAngle)$ then
    $leftWheelVelocity = 1;$ {maximum velocity is 1}
    $rightWheelVelocity = 1;$
else
    if $(targetPosition == RIGHT\_SIDE)$ then
        $tempVelocity = fuzzySpeed + fuzzyTurningAngle${LEFT WHEEL}
        $tempVelocity = max(min(tempVelocity, 1.0), 0.0)$ {max. value is 1.0, min.value is 0.0}
        $leftWheelVelocity = tempVelocity$
        $tempVelocity = fuzzySpeed - fuzzyTurningAngle${RIGHT WHEEL}
        $tempVelocity = max(min(tempVelocity, 1.0), 0.0)$
        $rightWheelVelocity = tempVelocity$
    else
        if $(targetPosition == LEFT\_SIDE)$ then
            $tempVelocity = fuzzySpeed - fuzzyTurningAngle$ {LEFT WHEEL}
            $tempVelocity = max(min(tempVelocity, 1.0), 0.0)$
            $leftWheelVelocity = tempVelocity$
            $tempVelocity = fuzzySpeed + fuzzyTurningAngle${RIGHT WHEEL}
            $tempVelocity = max(min(tempVelocity, 1.0), 0.0)$
            $rightWheelVelocity = tempVelocity$
        end if
    end if
end if
if $(angleFromTarget > min\_spin\_angle)$ then
    if $(targetPosition == RIGHT\_SIDE)$ then
        $tempVelocity = (fuzzySpeed + fuzzyTurningAngle) * 0.3$
        $tempVelocity = max(min(tempVelocity, 0.35), 0.0)$ {max. value is 0.35, min.value is 0.0}
        $leftWheelVelocity = tempVelocity$
        $rightWheelVelocity = -1 * tempVelocity$
    else
        if $(targetPosition == LEFT\_SIDE)$ then
            $tempVelocity = (fuzzySpeed + fuzzyTurningAngle) * 0.3$
            $tempVelocity = max(min(tempVelocity, 0.35), 0.0)$
            $rightWheelVelocity = tempVelocity$
            $leftWheelVelocity = -1 * tempVelocity$
        end if
    end if
end if
```

Algorithm 2: Fuzzy Spin and Target Pursuit

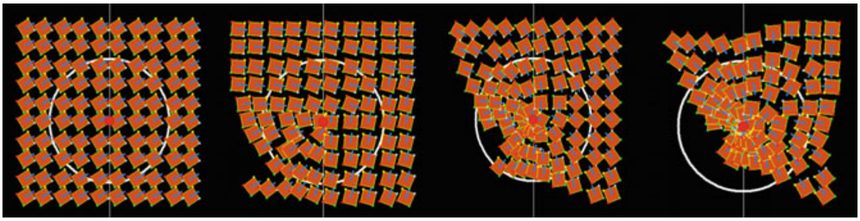


Fig. 2 Calibration maps for the combined target pursuit and spinning behaviours. The target is the *red ball*. Shown from *left* to *right*, are the sequence of robot movements taken at different time sequences

*5.2 Hybrid Fuzzy-D*lite in Action*

The scenarios depicted in Fig. 3 demonstrate how the hybrid algorithm performs planning and re-planning as the robot navigates an unknown terrain. The ***top-left*** of Fig. 3 shows the initial path-planning results, based on the free-space assumption about the unknown areas. The robot is completely oblivious of the presence of obstacles; therefore, the calculated shortest path (appearing as the connected red lines) included the unknown blocked cells. The ***top-right*** figure shows the cell connections representing the initial knowledge of the world, with assumptions. The unknown cells are yet to be discovered by the robot, and they are all assumed to be traversable. In conjunction, the details of the initial planning results, relevant to D*lite (h,g and rhs-values) can be viewed on the ***middle-left*** figure. Upon discovery that some of the unknown cells are actually blocked cells, re-planning is performed. A snapshot of one of the intermediary re-planning results can be viewed on the ***middle-right*** figure. At this stage, it can be seen that more green cells appear due to cell expansions. Finally, the two bottom figures depict the last stage of navigation. The complete smooth trajectory of the robot is shown at the ***bottom-left*** figure, and all the relevant details pertaining to D*lite is shown at the ***bottom-right***.

Using a set of 47 gridworlds of varying sizes, from 35 cells to 10000 cells, we measured the runtimes spent on D*lite calculations and the Fuzzy Logic controllers (Fig. 4). As can be viewed from Fig. 4, Fuzzy Logic's runtime is very fast, notching a maximum of only 921 μs, and can be considered approximately linear against the number of grid cells. On the other hand, D*lite's runtime notched a maximum of 9196 μs, and can be considered to be dependent on both the complexity of the grid topology and the presence of unknown cells. In terms of the proportion of time attributed to AI planning, the Fuzzy systems accounted for work between 0.64–11.30 %, while D*lite accounted for work between 88.70–99.36 %.

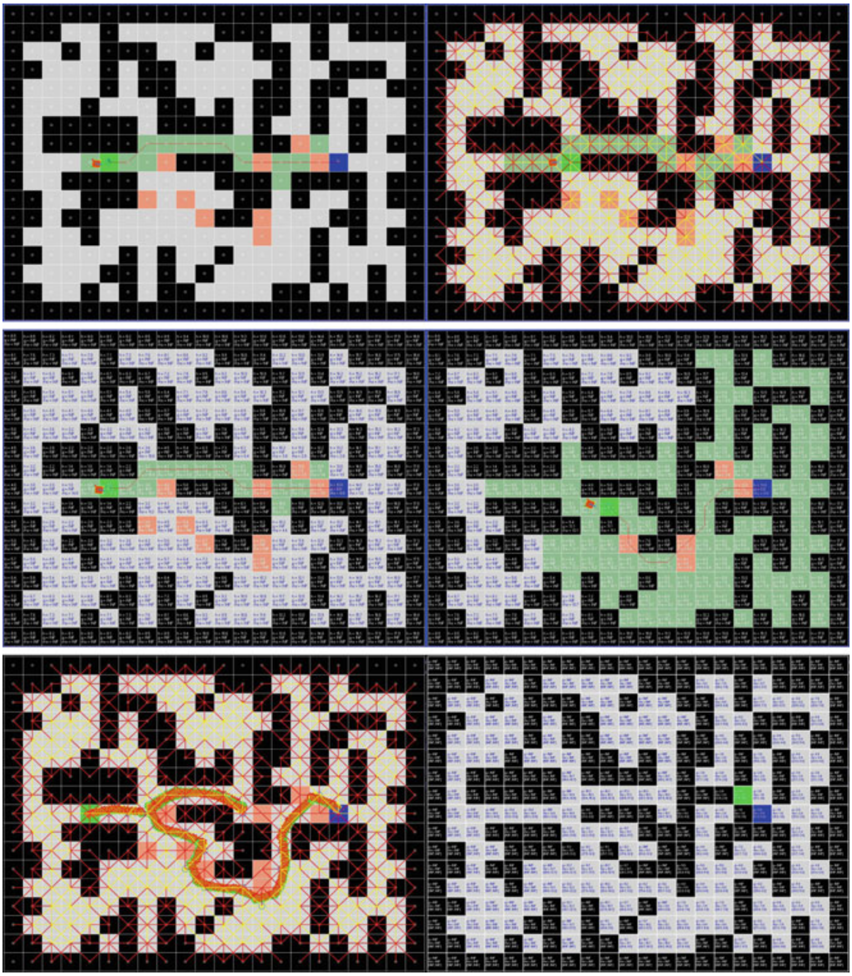


Fig. 3 The Hybrid Fuzzy-D*lite (using Euclidean distance heuristic) in action (planning and replanning). The gridworld in the experiment contains ***unknown areas*** in the map, marked by peach-coloured cells (these are actually blocked cells), ***known blocked areas*** as black cells, and ***known traversable areas*** as grey cells. *Green cells* indicate the vertices expanded during search, *red lines* indicate blocked connections, while *yellow lines* mean traversable connections. (S-start position, G-goal position)

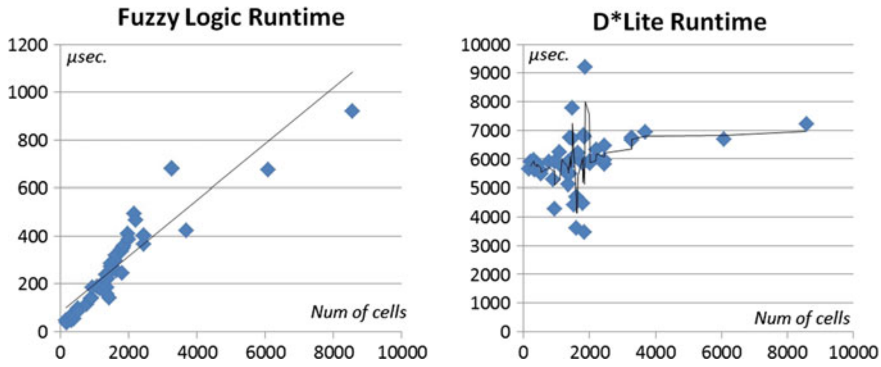


Fig. 4 Comparison of runtimes between D*Lite and the Fuzzy Logic Controller

6 Conclusion

We have proposed a novel hybrid Fuzzy-D*lite algorithm that combines two complementary algorithms for fast and smooth near-optimal goal-directed robot navigation in an unknown terrain, in real-time. The hybrid algorithm is capable of efficient replanning using the results of previous path-calculations, while the robot explores the environment using on-board sensors for more details about its traversability. Further, using a design architecture (for the cascade of fuzzy systems) that cuts down the feature space into half (whenever symmetry is present), the fusion of algorithms is capable of calculating the exact steering angle and absolute navigation speed for every fraction of a second, by controlling the left and right wheel velocities of a 2-wheeled robot.

References

1. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. **4**(2), 100–107 (1968)
2. Koenig, S., Likhachev, M.: Improved fast replanning for robot navigation in unknown terrain. In: Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA 2002, May 11–15, 2002, Washington, DC, USA, pp. 968–975 (2002)
3. Stentz, A.: The focussed d* algorithm for real-time replanning. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, Aug 20–25 1995, 2 vols., pp. 1652–1659 (1995)
4. Likhachev, M., Koenig, S.: Lifelong planning for mobile robots. In: Advances in Plan-Based Control of Robotic Agents, International Seminar, Dagstuhl Castle, Germany, Oct 21–26, 2001, Revised Papers, pp. 140–156 (2001)
5. Koenig, S., Likhachev, M., Furcy, D.: Lifelong planning A. Artif. Intell. **155**(1–2), 93–146 (2004)
6. Koenig, S., Likhachev, M.: Fast replanning for navigation in unknown terrain. IEEE Trans. Robot. **21**(3), 354–363 (2005)

7. Stentz, A., Hebert, M.: A complete navigation system for goal acquisition in unknown environments. Technical Report CMU-RI-TR-94-07, Carnegie Mellon (April 1994)
8. Likhachev, M., Koenig, S.: Incremental replanning for mapping. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, Sept 30–Oct 4, 2002, pp. 667–672 (2002)
9. Sun, X., Yeoh, W., Koenig, S.: Moving target d* lite. In: 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10–14, 2010, vol. 1–3, pp. 67–74 (2010)
10. Ferguson, D., Stentz, A.: Using interpolation to improve path planning: the field d* algorithm. J. Field Robot. **23**(2), 79–7101 (2006)
11. Reyes, N.H., Barczak, A.L., Susnjak, T., Sincák, P., Vašcák, J.: Real-time fuzzy logic-based hybrid robot path-planning strategies for a dynamic environment. In: Efficiency and Scalability Methods for Computational Intellect, pp. 115–141. IGI Global (2013)
12. Wu, P.P., Clothier, R., Campbell, D.A., Walker, R.: Fuzzy multi-objective mission flight planning in unmanned aerial systems. In: IEEE Symposium on Computational Intelligence in Multicriteria Decision Making, MCDM 2007, Honolulu, Hawaii, USA, April 1–5, pp. 2–9 (2007)

Accurate Localization in Urban Environments Using Fault Detection of GPS and Multi-sensor Fusion

Taekjun Oh, Myung Jin Chung and Hyun Myung

Abstract In order to make robots perform tasks autonomously, it is necessary for robots to know the surrounding environments. Therefore, a world modeling should be made in advance or concurrently. It is important to know an accurate position for the accurate world modeling. The aim of this paper is an accurate localization method for the world modeling under the situation where the portion of signals from global positioning system (GPS) satellites is blocked in urban environments. In this paper, we propose a detection method for non-line-of-sight satellites and a localization method using the GPS, the inertial measurement unit (IMU), the wheel encoder, and the laser range finder (LRF). To decide whether the signal from the satellite is blocked by the building, the local map that is made from the local sensors and an LRF is exploited. Then the GPS reliability is established adaptively in a non-line-of-sight situation. Through an extended Kalman filter (EKF) with the GPS reliability the final robot pose is estimated. To evaluate the performance of the proposed methods, the accuracy of the proposed method is analyzed using ground truth from Google maps. Experimental results demonstrate that the proposed method is suitable for the urban environments.

Keywords Multi-sensor · Localization · World modeling · Non-line-of-sight

T. Oh (✉) · H. Myung
Department of Civil and Environmental Engineering, KAIST, Daejeon 34141, Korea
e-mail: buljaga@kaist.ac.kr

H. Myung
e-mail: hmyung@kaist.ac.kr

M.J. Chung
Department of Electrical Engineering, KAIST, Daejeon 34141, Korea
e-mail: mjchung@kaist.ac.kr

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_4

1 Introduction

Localization is an essential technique for the robot autonomy. If the robot does not know where it is, the robot cannot perform any task because of the lack of surrounding environment information. In order to solve the robot localization problem, various ways have been actively researched. Simultaneous localization and mapping (SLAM) [1–8] that makes the robot navigate in unknown environments has been investigated as one way for the robot localization.

Localization approaches using the prior map data [9–12] have also received much attention in recent years. For this approach, the map data that is made in advance, for example Google maps, 3D point cloud maps, and image maps, are required in order to determine the robot pose. Matching between the prior map data and current sensor information enables the robot to estimate the final pose. Although this approach relatively guarantees accurate robot pose estimation than other robot approaches, this approach should have the prior map data and may result in the localization failure due to the inaccurate map information. In efforts to increase the rate of the robot pose estimation accuracy, the global positioning system (GPS) is used for the robot localization. Obst et al. have proposed a localization method using the GPS and a 3D map [13]. The 3D map consists of a digital terrain model called Open Street Map and a digital elevation model. The virtual ray from the robot to a satellite is generated and makes the robot know whether the satellite signal is blocked by a building or not. By detecting non-line-of-sight satellites, the GPS reliability can be determined to help the robot estimate its pose. However, to make the 3D map, it is necessary to have the digital elevation model and Open Street Map, but the reliability of Open Street Map is not guaranteed because Open Street Map can be modified anonymously on the web. The non-line-of-sight satellites detection algorithm using an omnidirectional camera has been proposed by Meguro et al. [14]. Using the camera image, the non-line-of-sight satellite signal of GPS is filtered and the accuracy of robot pose estimation is increased. However, illumination changes cause errors for the robot localization using the camera.

In this paper, we propose an accurate localization approach based on a detection method for non-line-of-sight satellites and a pose estimation method using an extended Kalman filter (EKF) algorithm for the world modeling in urban environments. A local map enables the robot to discriminate the non-line-of-sight satellites. Through results from non-line-of-sight satellite detection, the reliability of a GPS is determined to estimate the final robot pose. The EKF corrects the GPS error by fusing the multi-sensor data.

The remainder of this paper is organized as follows. In Sect. 2, the overall algorithm for the accurate localization is described in detail. Section 3 introduces the experimental setup and verifies the performance of the proposed method. In Sect. 4, conclusion and future works are discussed.

2 Localization Using Non-line-of-Sight Satellite Detection

In this section, EKF-based pose estimation for the world modeling through the non-line-of-sight satellite detection algorithm is introduced. First, generating a local map using local sensors in urban environments is presented. Then, the approach is described how to detect the non-line-of-sight satellite using the local map and determine a reliability of the GPS. Finally, EKF-based pose estimation by the reliability of the GPS is presented.

2.1 *Local Map*

It is possible to detect the non-line-of-sight satellite backward direction of the robot using the map that is generated through the robot moving. However, the robot cannot detect the non-line-of-sight satellite forward robot direction because of the lack of map information. Therefore, a local map helps the robot detect the non-line-of-sight satellite in this situation. The local map is generated by local sensors. Although local sensors accumulate the error, local sensors are relatively accurate for short distance robot moving. The process of generating the local map and an example scene of the local map are shown in Figs. 1 and 2, respectively. The

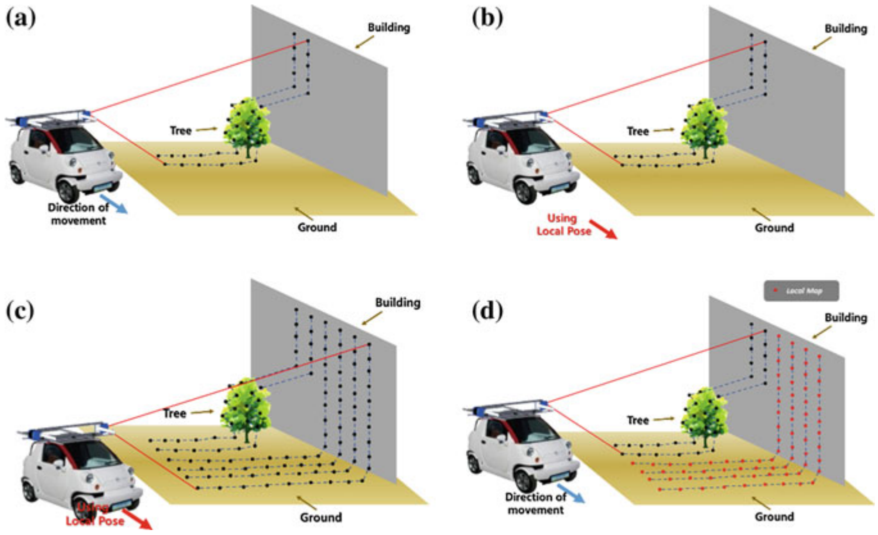


Fig. 1 The process of generating local map. **a** before generating local map; **b** calculation local pose using the local sensors; **c** generating local map using the local pose; **d** generation of local map. *Red dots* indicate the local map

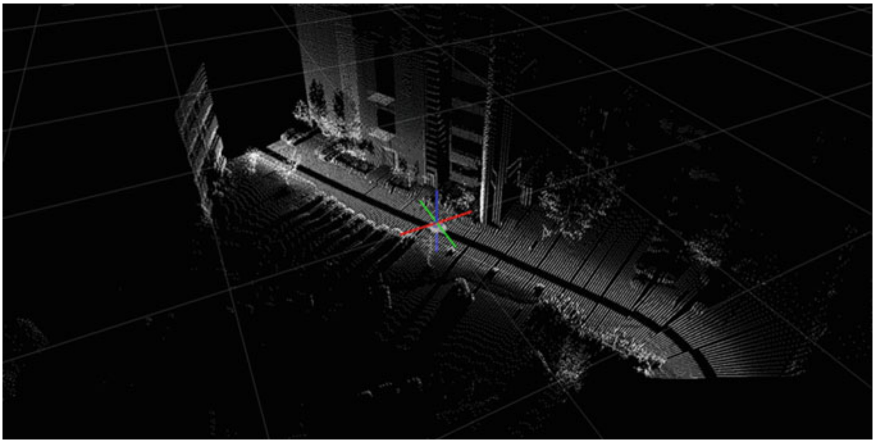


Fig. 2 Local map generated using local sensors

robot discriminates building information of forward direction of the robot using the local map.

2.2 Non-line-of-Sight Satellite Detection Algorithm

Although a building is located in forward robot direction, the robot can detect this building from the results of the local map. Therefore, robot can detect the non-line-of-sight satellite when the GPS data are acquired. Figure 4 illustrates the concept of the non-line-of-sight satellite. Point $P_0 = (x_0, y_0, z_0)$ means the robot position from the GPS data and point $P_1 = (x_1, y_1, z_1)$ is the point cloud data on the local map from the depth information r acquired by the laser range finder (LRF). Vector $\vec{A} = (a, b, c)$ is direction vector from the robot to the satellite from GPS data. A virtual ray from the robot to the satellite is defined by a line equation and point $P_2 = (x_2, y_2, z_2)$ is derived by a perpendicular foot from P_1 to the virtual ray. The shortest distance d from P_1 to P_2 is used to check whether the virtual ray from the robot to the satellite is blocked by building or not. A signal from the satellite is determined as non-line-of-sight signal, if shortest distance d is smaller than a threshold value d_{th} and a height value z_2 of point P_2 is smaller than the height value z_1 of point P_1. The results of the non-line-of-sight satellite detection algorithm are shown Fig. 3. The gray line is the signal that is not acquired from the GPS or that is not used for calculating of pose data, although that signal is acquired from the GPS. The red and green line is exploited for calculating of pose data. The red line means the non-line-of-sight satellite signal that is determined by our detection algorithm and the green line indicate the direct satellite signal.

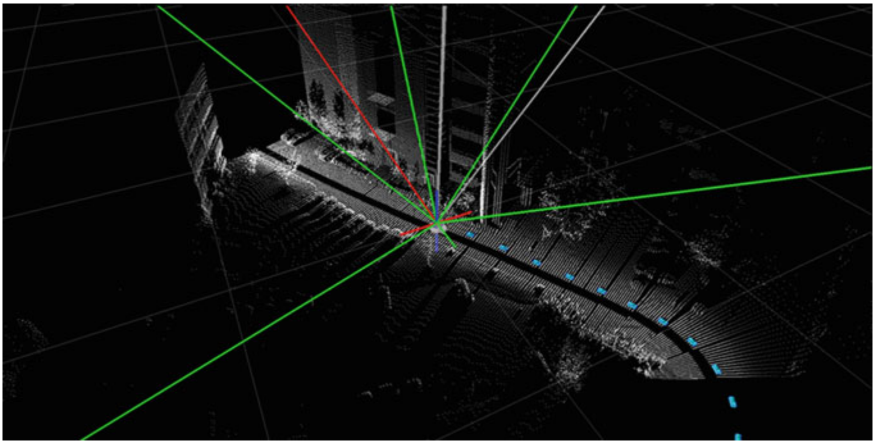


Fig. 3 The results of the non-line-of-sight satellite detection using the local map. The results are acquired from Fig. 2

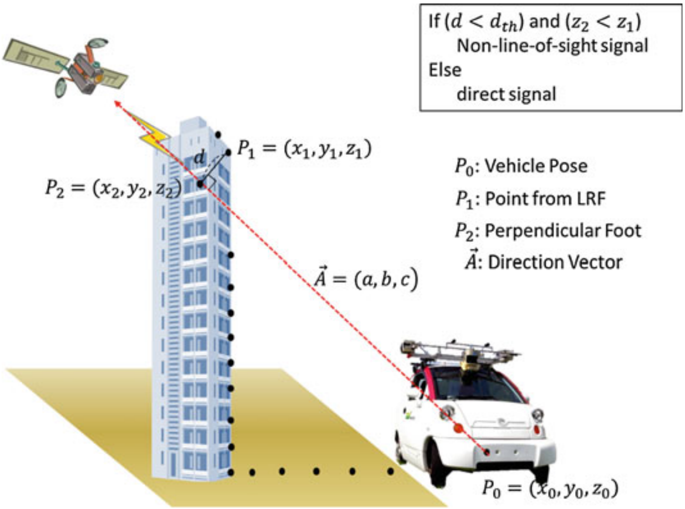


Fig. 4 The concept of the non-line-of-sight satellite detection using the local map

2.3 GPS Reliability

If the non-line-of-sight satellite is detected, the GPS data should not be used directly. It is not possible to trust the GPS data in the non-line-of-sight situation. It is necessary correcting algorithm for the GPS information. In this subsection, the generation of the GPS reliability is described.

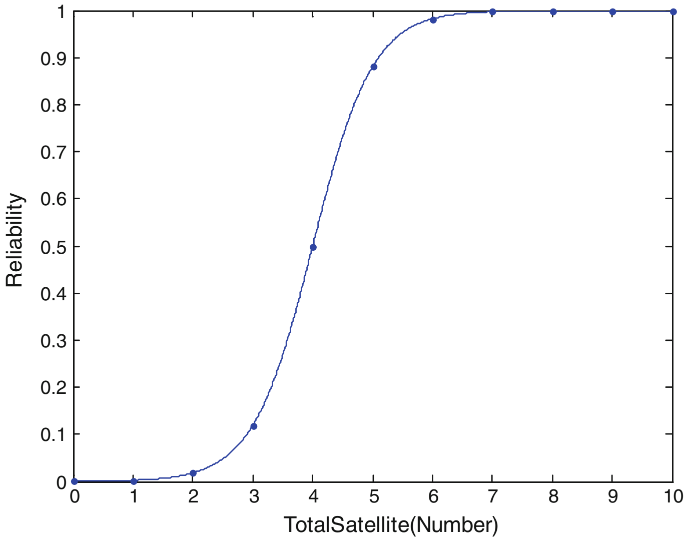


Fig. 5 GPS reliability $(K_a = 2,\ K_s = 4)$

The number of available satellite S^A that is acquired from the GPS data means the number of satellite for calculating the pose. The number of non-line-of-sight satellite S^N is obtained from the results of our detection algorithm in previous Sect. 2.2. The number of total reliable satellite S^T is defined as follows:

$$S^T = S^A - S^N. \tag{1}$$

The reliability of GPS is written as a modified sigmoid function as follows:

$$R^{\text{GPS}} = \frac{1}{1 + e^{-K_a (S^T - K_s)}} \tag{2}$$

where K_a and K_s are user parameters for the GPS reliability. The user parameter K_s is usually set 4, because to acquire pose data from the GPS, four satellites are required at a minimum. The other user parameter K_a is set bigger than 2. K_a changes the slope of sigmoid function. If value of K_a is smaller than 2, the reliability of GPS is not reflected rapidly. The GPS reliability when K_a and K_s are 2 and 4, respectively, is shown in Fig. 5.

2.4 EKF-Based Pose Estimation

In this subsection, the EKF-based pose estimation approach using the GPS reliability is presented. The system modeling of the EKF is defined as follows:

$$x_k = f(x_{k-1}) + w_{k-1}, \quad w_{k-1} \sim N(\vec{0}, Q) \tag{3}$$

$$z_k = h(x_{k-1}) + v_{k-1}, \quad v_{k-1} \sim N(\vec{0}, R) \tag{4}$$

where state vector $x = [X, Y, V, \theta, \omega]^T$ and measurement $z = [X^{\mathrm{GPS}}, Y^{\mathrm{GPS}}, V^{\mathrm{WO}}, \theta^{\mathrm{GPS}}, \omega^{\mathrm{IMU}}]^T$. The components of state vector X, Y, V, θ, and ω are latitudinal position, longitudinal position, velocity, rotation angle, and angular rate of robot, respectively. The component of measurement X^{GPS}, V^{WO}, and θ^{GPS} are data from the GPS, V^{WO} from wheel odometry (WO), and ω^{IMU} from inertial measurement unit (IMU). The process noise w_{k-1} and measurement noise v_{k-1} are denoted Gaussian noise that is zero mean and covariance Q and R, respectively. The process model f and the measurement model h is defined as follows:

$$f(x_k) = \begin{bmatrix} X_k + V_k \Delta t_k \cos(\theta_k) \\ Y_k + V_k \Delta t_k \sin(\theta_k) \\ V_k \\ \theta_k + \omega_k \Delta t_k \\ \omega_k \end{bmatrix} \tag{5}$$

$$h(x_k) = \begin{bmatrix} X_k \\ Y_k \\ V_k \\ \theta_k \\ \omega_k \end{bmatrix}. \tag{6}$$

Although the covariance R is determined by the sensor specification, in this paper adaptive technique using the weight through the GPS reliability is proposed as follows:

$$R = W \cdot \mathrm{diag}\left(\sigma^X, \sigma^Y, \sigma^V, \sigma^\theta, \sigma^\omega\right) \tag{7}$$

where σ^X, σ^Y, σ^V, σ^θ, and σ^ω indicate standard deviation of latitudinal position from GPS, longitudinal position from GPS, velocity from WO, rotation angle from GPS, and angular rate from IMU, respectively. The weight matrix W is defined as follows:

$$W = \mathrm{diag}\left(w^X, w^Y, 1, w^\theta, 1\right) \tag{8}$$

where the components of weight matrix is written using the GPS reliability in Eq. (2) as follows:

$$w^X = w^Y = w^\theta = 2 - R^{\mathrm{GPS}}. \tag{9}$$

3 Experiments

In this section, an experimental setup and the results of proposed method are described in detail. To confirm the performance of our method, the sensor system, the experimental environment, and the method for implementation are presented below. The experiment results of the proposed method were compared another method graphically and numerically to validate the effectiveness of our method.

3.1 Experimental Setup

The experimental environment and sensor system are illustrated in Fig. 6. The system is composed of an electrical vehicle, three LRF, two wheel encoder, a GPS and an IMU. In experiment, front LRF is not exploited. The experimental site is Korea advanced institute of science and technology (KAIST) in Daejeon, South Korea. The size of the experimental environment is about 250 m × 310 m, elapsed time is 265 s, and a moving distance for the experiment is 1029 m. In the experimental environment, there are some buildings to block the satellite signal.

3.2 Results

The graphical experimental results are shown in Figs. 7 and 8. In Fig. 7, the blue circle indicates the pose from GPS, the green solid line WO, the yellow solid line WO + IMU, the light blue dot GPS + WO + IMU, and the red solid line proposed method. The results of WO, WO + IMU, GPS + WO + IMU diverge from the road line. The result of GPS is discontinuous. This phenomenon is caused by the non-line-of-sight satellite GPS. Discontinuous world modeling from the GPS data is

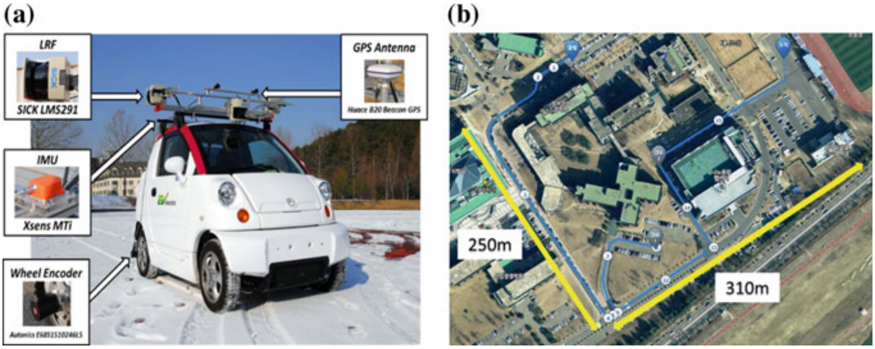


Fig. 6 Experimental setup. **a** Sensor system; **b** experimental environment

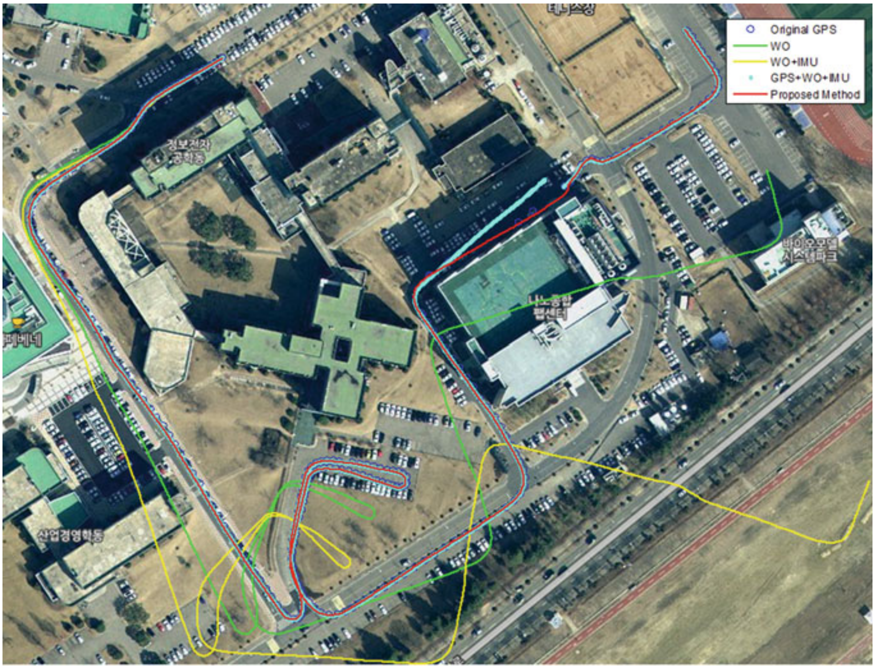


Fig. 7 Experiment results. The *blue circle* means the pose from GPS, the *green solid line* WO, the *yellow solid line* WO + IMU, the *light blue dot* GPS + WO + IMU, and the *red solid line* proposed method

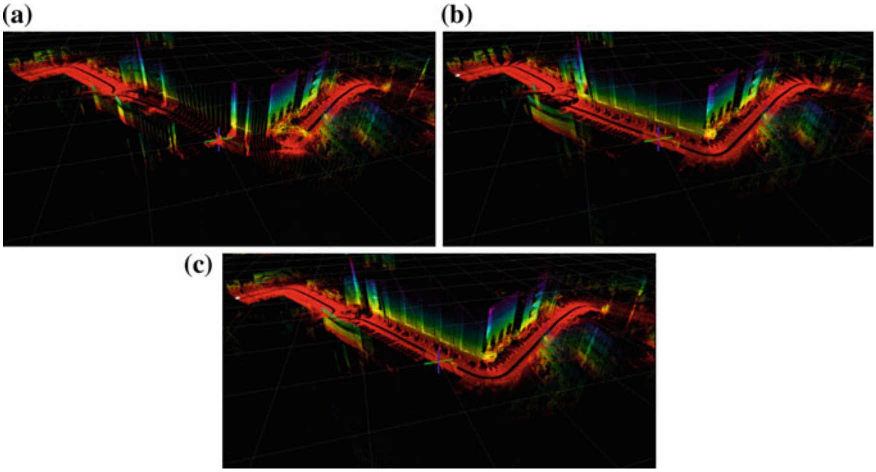


Fig. 8 World modeling results. **a** GPS; **b** GPS + WO + IMU; **c** proposed method

Table 1 RMSE results

	RMSE (m)
GPS	5.174
GPS + WO + IMU	3.580
Proposed method	2.748

depicted in Fig. 6a. The graphical result of our proposed method is well matched to the road line in Fig. 7 and shows the continuous world modeling in Fig. 8c. The numerical results of root mean square error (RMSE) are described in Table 1. The error our proposed method is smaller than the others.

4 Conclusions and Future Works

We proposed an approach for accurate localization for world modeling based on a detection method for non-line-of-sight satellites and a pose estimation method using the EKF algorithm in urban environments. A framework of building a local map, detecting the non-line-of-sight satellite, pose correction and building the accurate world modeling had been developed in this paper. We confirmed that the results of our proposed method have better performance graphically and numerically.

Height data from the GPS in not considered and overall algorithm is conducted offline in this paper. For future works, we will expand our method to a 3D and make the real-time algorithm by modifying the sensor system.

Acknowledgments This work was financially supported by the Korean Ministry of Land, Infrastructure and Transport (MOLIT) as 「U-City Master and Doctor Course Grant Program」

References

1. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: AAAI/IAAI, pp. 593–598 (2002)
2. Grisetti, G., Stachniss, C., Burgard, W.: Improving grid-based SLAM with Rao-blackwellized particle filters by adaptive proposals and selective resampling. In: IEEE International Conference on Robotics and Automation, pp. 2432–2437. IEEE Press, New York (2005)
3. Dellaert, F., Kaess, M.: Square root SAM: simultaneous localization and mapping via square root information smoothing. Int. J. Robot. Res. **25**(12), 1181–1203 (2006)
4. Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J.J., Dellaert, F.: iSAM2: incremental smoothing and mapping using the bayes tree. Int. J. Robot. Res. **31**(2), 217–236 (2011)
5. Kim, H., Oh, T., Lee, D., Choe, Y., Chung, M.J., Myung, H.: Mobile robot localization by matching 2D image features to 3D point cloud. In: Ubiquitous Robots and Ambient Intelligence (2013)
6. Kim, H., Lee, D., Oh, T., Lee, S.W., Choe, Y., Myung, H.: Feature-based 6-DoF camera localization using prior point cloud and images. In: Robot Intelligence Technology and Applications, pp. 3–11. Springer International Publishing (2014)

7. Lee, D., Jung, J., and Myung, H.: Pose graph-based RGB-D SLAM in low dynamic environments. In: IEEE International Conference on Robotics and Automation. IEEE Press, New York (2014)
8. Lee, D., Myung, H.: Solution to the SLAM problem in low dynamic environments using a pose graph and an RGB-D sensor. Sensors **14**(7), 12467–12496 (2014)
9. Baldwin, I., Newman, P.: Road vehicle localization with 2D push-broom LIDAR and 3D priors. In: IEEE International Conference on Robotics and Automation, pp. 2611–2617. IEEE Press, New York (2012)
10. Qin, B., chong, Z.J., Bandyopadhyay, T., Ang. Jr. M. H., Frazzoli, E., Rus, D.: Curb-intersection feature based monte carlo localization on Uban roads. In: IEEE International Conference on Robotics and Automation, pp. 2640–2646. IEEE Press, New York (2012)
11. Stewart, D., Newman, P.: LAPS-localization using appearance of prior structure: 6-DoF monocular camera localisation using prior pointclouds. In: IEEE International Conference on Robotics and Automation, pp. 2625–2632. IEEE Press, New York (2012)
12. Senlet, T., Elgammal, A.: Satellite image based precise robot localization on sidewalks. In: IEEE International Conference on Robotics and Automation, pp. 2647–2653. IEEE Press, New York (2012)
13. Obst, M., Bauer, S., Reisdorf, P., Wanielik, G.: Multipath detection with 3D digital maps for robust multi-constellation GNSS/INS vehicle localization in urban areas. In: IEEE Intelligent Vehicles Symposium, pp. 184–190. IEEE Press, New York (2012)
14. Meguro, J.-I., Murata, T., Takiguchi, J.-I., Amano, Y., Hashizume, T.: GPS multipath mitigation for urban area using omnidirectional infrared camera. IEEE Trans. Intell. Transport. Syst. **10.1**, 22–30 (2009)

Simultaneous Localization and Mapping with a Dynamic Switching Mechanism (SLAM-DSM)

Chun-Hsiao Yeh, Herng-Hua Chang, Chen-Chien Hsu and Wei-Yen Wang

Abstract In this paper, we propose a simultaneous localization and mapping (SLAM) algorithm incorporating a dynamic switching mechanism to switch between FastSLAM 1.0 and 2.0, based on a threshold of effective sample size (ESS). By taking advantages of FastSLAM 1.0 and 2.0 through the proposed dynamic switching mechanism, execution efficiency is significantly improved while maintaining an acceptable accuracy of estimations. To show the effectiveness of our proposed approach in comparison to FastSLAM 1.0 and 2.0, several simulations are demonstrated in this paper.

Keywords FastSlam · Particle filter · Extended Kalman filter

1 Introduction

Simultaneous localization and mapping (SLAM) plays an important role in the field of robot navigation in unknown environment, where the robot has to build a map from its surroundings while localizing itself simultaneously [1]. The algorithm builds the whole map of the environment and the robot's position using landmark measurements collected from robot information and the odometer. According to

C.-H. Yeh (✉) · H.-H. Chang
Department of Engineering Science and Ocean Engineering, National Taiwan University,
No. 1, Sec. 4, Roosevelt Rd, Taipei, Taiwan
e-mail: danielyehh@gmail.com; r04525061@ntu.edu.tw

H.-H. Chang
e-mail: herbertchang@ntu.edu.tw

C.-C. Hsu · W.-Y. Wang
Department of Electrical Engineering, National Taiwan Normal University,
162 He-Ping/East Rd., Sec. 1, Taipei, Taiwan
e-mail: jhsu@ntnu.edu.tw

W.-Y. Wang
e-mail: wywang@ntnu.edu.tw

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_5

previous study, SLAM algorithms are divided into two categories known as, EKF-SLAM [2] and FastSLAM [3]. EKF-SLAM is based on extended Kalman filter, which puts robot's odometer information and landmark measurements into EKF's covariance matrix, while repeating prediction and updating processes at the same time. Several attempts have been made to improve EKF-SLAM's performance. For instance, a map management method [4] was proposed to increase robot pose accuracy and landmark estimation by deleting suitable landmarks. Moreover, several soft-computing approaches [5, 6] were presented to make EKF-SLAM algorithms more robust. However, due to the computational complexity of EKF-SLAM algorithms, data association easily fails when sensor update takes too much time for computing a large number of landmarks.

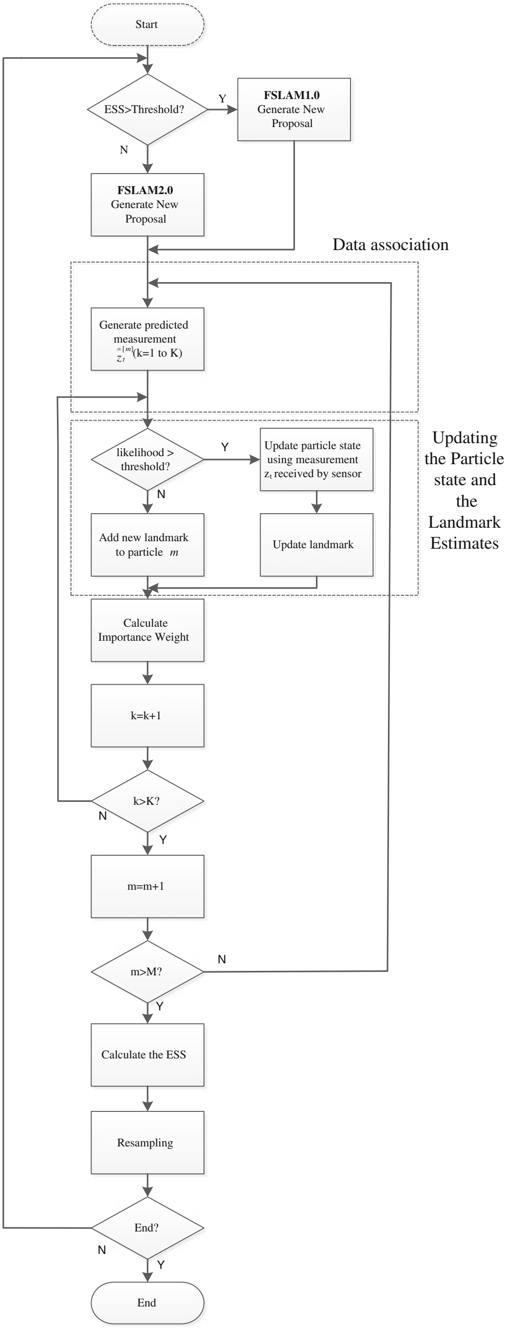
To solve the problem of computational complex in EKF-SLAM, several FastSLAM algorithms were presented based on Rao–Blackwellized particle filter (RBPF) [7]. Basically, FastSLAM solve part of localization by using particle filter (PF), while solving part of mapping by using EKF. Different from EKF-SLAM, FastSLAM calculates each landmark individually with EKF for, it eliminates complex calculation of excessively large matrix. From previous research, the initial FastSLAM version is named FastSLAM1.0, which estimates the robot position only by using particle filter (PF). Unfortunately, the estimation accuracy is far from satisfactory since only control inputs are used to predict the location of the robot. Thus, a modified version called FastSLAM2.0 was created to solve this problem. FastSLAM2.0 relied not only on motion estimate but also sensor measurement to improve the estimation accuracy. Though, FastSLAM2.0 performs better than FastSLAM1.0, its runtime efficiency is compromised, especially when a huge number of landmarks appear.

This paper proposes modified SLAM which improves the computational efficiency to solve the burden of wasting a lot of runtime to run FastSLAM2.0,

2 SLAM-DSM for Mobile Robots

In order to improve the computational efficiency, we design a switching mechanism between FastSLAM1.0 and 2.0 based on the effective sample size (ESS) as a threshold. If the ESS value is smaller than a preset threshold, the difference between each particle's importance weights is large, and we run FastSLAM2.0 at that iteration. However, if the ESS value is larger than the threshold, the difference between each particle's importance weights is small. Then we switch FastSLAM2.0 to FastSLAM1.0 to reduce computation at that iteration. Figure 1 shows the flowchart of the proposed SLAM (SLAM-DSM). Detailed descriptions are listed below

Fig. 1 Flowchart showing the SLAM-DSM



2.1 Generate a New Proposal

For particle $m = 1, \ldots, M$, a probabilistic guess about position of robot which agrees with the control input at recent time t is generated:

$$x_t^m \sim p\left(x_t | x_{t-1}^m, u_t\right) \tag{1}$$

Then, we use these particles and landmarks k in these particles to generate estimated measurements

$$\hat{z}_t^m = g\left(\hat{x}_t^m,\ \theta_{k,t-1}^m\right) \tag{2}$$

Calculate the landmarks by using the estimated measurements model of Jacobian matrix

$$H_{\theta_k} = \partial g\left(\hat{x}_t^m,\ \theta_{k,t-1}^m\right) / \partial \theta_{k,t-1}^m \tag{3}$$

The innovation covariance matrix can be calculated as

$$S_k = H_{\theta_k} \sum_{k,t-1}^{m} H_{\theta_k}^T + R_t \tag{4}$$

Since H_θ is the Jacobian of measurement model with landmark pose. The covariance and mean of the sampling distribution can now be obtained.

$$\begin{gathered}\sum_{x_t,t}^{m} = \left[H_{x_t}^T (S_k)^{-1} H_{x_t} + \left(\sum_{x_t,t}^{m}\right)^{-1}\right]^{-1} \\ u_{x_t,t}^m = \sum_{x_t,t}^{m} H_{x_t}^T (S_k)^{-1}\left(z_t - \hat{z}_t^m\right) + u_{x_t,t-1}^m\end{gathered} \tag{5}$$

Finally, the new sample is drawn in Gaussian approximation

$$x_t^m \sim N\left(u_{x_t,t}^m,\ \sum_{x_t,t}^{m}\right) \tag{6}$$

2.2 Data Association

After estimating the position of the robot, we use particle's state and landmarks information $k = 1, \ldots, N$ in particle m to generate estimated measurements

$$\hat{z}_t^m = g\left(\hat{x}_t^m,\ \theta_{k,t-1}^m\right) \tag{7}$$

The likelihood between actual measurements z_t and expected measurements $\hat{z}_t^m$ is obtained by using following equations:

$$p_k = \frac{1}{\sqrt{|2\pi S_k|}} \exp\left(-\frac{1}{2}(z_t - \hat{z}_{k,t})^T (S_k)^{-1} (z_t - \hat{z}_{k,t}) \right) \tag{8}$$

S_k is the innovation covariance matrix:

$$S_k = H_{\theta_k} \sum_{k,t-1}^{m} H_{\theta_k}^T + R_t \tag{9}$$

If any likelihood is larger than the threshold, update the landmark with the largest likelihood. However, we regard this landmark as a new one, and add it into the particle.

2.3 Updating the Particle State and the Landmark Estimates

Each landmark has its own mean and covariance in FastSLAM2.0, which is the same as FastSLAM1.0. Landmark estimates can be updated by using EKF.

$$\begin{aligned}
K &= \sum_{n_t,t-1}^{m} H_{\theta_{\hat{n}}}^T S_{\hat{n}}^{-1} \\
u_{\hat{n}_t,t}^m &= u_{\hat{n}_t,t-1}^m + K\left(z_t - \hat{z}_t^m\right) \\
\sum_{\hat{n},t}^{m} &= \left(I - KH_{\theta_{\hat{n}}}\right) \sum_{\hat{n},t-1}^{m}
\end{aligned} \tag{10}$$

If there is no likelihood higher than the threshold, we add this new landmark into the particle rather than updating its information

$$\begin{aligned}
N_t^m &= N_{t-1}^m + 1 \\
u_{\hat{n},t}^m &= g^{-1}\left(x_t^m, \hat{z}_{\hat{n},t-1}\right) \\
\sum_{\hat{n},t}^{m} &= \left(H_{\theta_{\hat{n}}}^T R_t^{-1} H_{\theta_{\hat{n}}}\right)^{-1}
\end{aligned} \tag{11}$$

2.4 Calculating Importance Weight and Effective Sample Size

We use the following equations to calculate importance weight for each particle:

$$w_t^{[m]} = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{p(x^{t,m} | z^t, u^t, n^t)}{p(x^{t-1,m} | z^{t-1}, u^{t-1}, n^{t-1}) p\left(x_t^m | x^{t-1,m}, z^t, u^t, n^t\right)} \tag{12}$$

Then, we use importance weight to obtain the effective sample size (ESS).

$$\text{ESS}_t = \frac{M}{1 + cv_t^2} \tag{13}$$

In effective sample size equation, M is defined by the number of particles which is used in SLAM, while cv_t^2 is defined by the importance weight's coefficient of variation.

$$cv_t^2 = \frac{1}{M} \sum_{i=1}^{M} (Mw(i) - 1)^2 \tag{14}$$

Basically, our algorithm use effective sample size (EES) which is related to the value of difference between every importance weights. It is used to be a threshold at beginning of next iteration for deciding to run FastSLAM1.0 or FastSLAM2.0. The algorithm code is shown below.

```
begin

if ESS > = thershold do
FastSLAM1.0
else
FastSLAM2.0

end.
```

2.5 Resampling

A tournament selection mechanism is used for resampling after all particles being assigned importance weight and ready to enter next iteration.

3 Simulation Results

Basically, we compare FastSLAM1.0 and FastSLAM2.0 with our proposed idea to evaluate the performance. The software environment is to use Visual Studio 2012 with the OpenCV library, while the hardware environment is to use a laptop with Intel Core i5-3230 M, 2.6 GHz CPU, 8 GB RAM. In our experiments, we use 16, 50, 70 landmarks to set each environment on a map that have 500 * 500 pixels in dimension, and use 16 particles for SLAM algorithms. Each robot's move is represented by 10 pixels, and each rotation is 15°. Gaussian noise is introduced into the distance and rotation is obtained from odometer. The laser is called Virtual LRF which has maximum distance range of about 100 pixels and scanning range of 180°. It has 181 measurements for each individual scanning. In each run, robot needs to move 100 steps along the path. Therefore, we execute 20 runs in each separate environment.

Figure 2 shows the simulation results of various SLAM algorithms, where the path in pink is the robot's trajectory, and the path in gray is the estimated path of particle with the highest weight among those 16 particles. Particles of the proposed approach are shown in small green circles, and a robot is represented by yellow circle which is overlapped by the convergence of particles. Landmarks are marked by black dot surrounded by a brown circle, while the estimation of landmark location is represented by red circles. Virtual LRF scanning is shown by 181 blue lines. From the simulation in Fig. 2, we find out that the robot's localization and landmark estimation by SLAM-DSM are more accurate than FastSLAM1.0, and similar to FastSLAM2.0.

In the experiments, we set the particle number as 16, so the value of effective sample size (ESS) region is from 0 to 16. Figure 3 shows three charts set with 16, 50, 70 landmarks into three different environment. Also, x-axis is defined by run time, while y-axis is defined by localization error. We run SLAM and set ESS to be threshold from 3 to 16. For example, if we run SLAM in 70 landmarks situation and set ESS as 8, we can obtain the localization error to be 1.01 and run time is 143 s. So, we have this point on x-y coordinate system, which x value is 143, and y value is 1.01. Finally, we obtain many points from each ESS' threshold and analyze it.

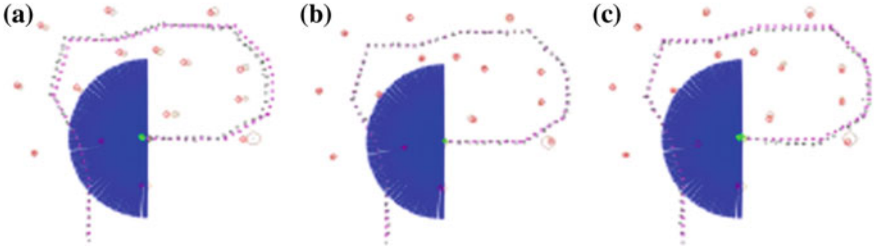
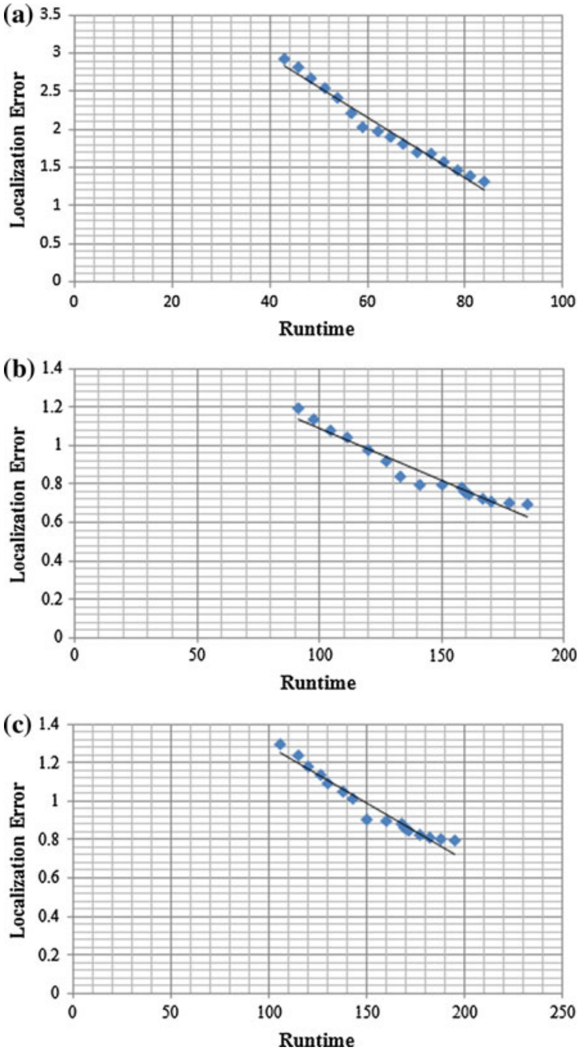


Fig. 2 Simulation results of different SLAM algorithms. **a** FastSLAM1.0. **b** SLAM-DSM. **c** FastSLAM2.0

Fig. 3 Simulation results of ESS from 3 to 16 in different environment. **a** 16 landmarks, **b** 50 landmarks, **c** 70 landmarks

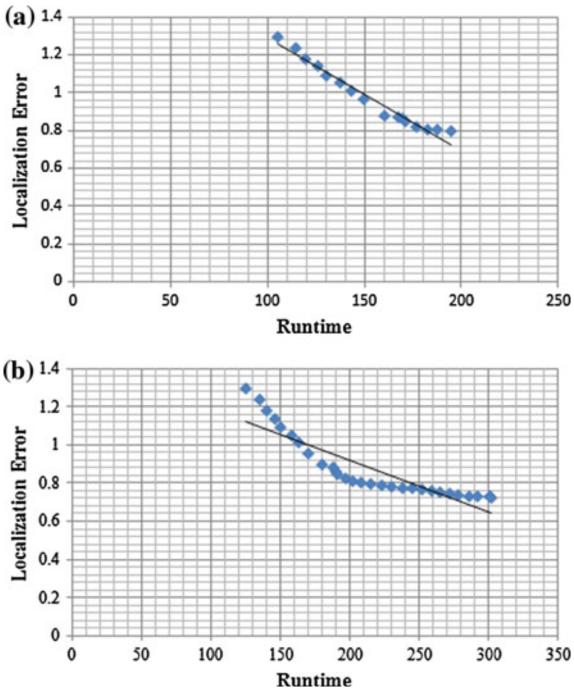


Then, we choose optimal ESS's threshold which is on the left side of the black line, to represent smaller value of localization error and run time. Apparently, we find that the value of optimal ESS is near half of its particle number in three different situations.

Moreover, we use different environment to do experiment to prove that our result is correct. As shown in the simulation of Fig. 4, we keep number of landmark same (70 landmarks), and change particle number from 16 to 20, 30. Finally, we can prove that the ideal ESS is close to half of its particle number.

Table 1 represents results of different SLAM algorithms by running each 20 times individually. The 'Localization error' stands for the mean error between the

Fig. 4 Simulation results of 20, 30 particle numbers under 70 landmarks. **a** particle number = 20, **b** particle number = 30



true landmark locations and those estimated by best particle after simulations is completed. To demonstrate computational efficiency, we also show the average runtime of various FastSLAM algorithms under different number of landmarks. In the experiment, we run the SLAM under 16 landmarks represented in Table 1, 50 landmarks in Table 2, 70 landmarks in Table 3. All three Tables have shown that

Table 1 Localization error and runtime efficiency of different SLAM algorithms under 16 landmarks (Choosing optimal ESS = 7 in SLAM-DSM)

SLAM Algorithm	Localization error (pixel)	Runtime (s)
FastSLAM 1.0	2.93	42.94
FastSLAM 2.0	1.41	86.90
SLAM-DSM	1.83	54.10

Table 2 Localization error and Runtime efficiency of different SLAM algorithms under 50 landmarks (Choosing optimal ESS = 8 in SLAM-DSM)

SLAM Algorithm	Localization error (pixel)	Runtime (s)
FastSLAM 1.0	1.20	98.38
FastSLAM 2.0	0.69	187.60
SLAM-DSM	0.79	133.50

Table 3 Localization error and Runtime efficiency of different SLAM algorithms under 70 landmarks (Choosing optimal ESS = 9 in SLAM-DSM)

SLAM Algorithm	Localization error (pixel)	Runtime (s)
FastSLAM 1.0	1.29	109.47
FastSLAM 2.0	0.79	198.38
SLAM-DSM	0.88	146.30

SLAM-DSM reduces a lot of runtime compared to FastSLAM 2.0, but localization error increases just a little. In summary, SLAM-DSM improves the computational efficiency of FastSLAM 2.0.

4 Conclusion

In this paper, we propose a dynamic switching mechanism to improve the runtime efficiency of FastSLAM2.0 and estimate accuracy of FastSLAM1.0. The main idea of this paper is to design a mechanism to switch FastSLAM1.0 and FastSLAM2.0. In the algorithm, we use the effective sample size (ESS) to be threshold for switching, which is related to the value of difference between all the importance weights. Through conducting the experiment in different situations, we can choose an ideal effective sample size (ESS) to be a threshold for reducing maximum runtime and localization error in those situations. As a result, our algorithm uses the advantage of FastSLAM1.0 and FastSLAM2.0 to demonstrate better runtime efficiency and estimation accuracy.

References

1. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1151–1156 (2003)
2. Smith, R.C., Cheeseman, P.: On the representation and estimation of spatial uncertainty. Int. J. Robot. **5**, 56–58 (1986)
3. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: Proceedings of AAAI National Conference on Artificial Intelligence, pp. 593–598 (2002)
4. Dissanayake, G., Williams, S.B., Durrant-Whyte, H., Bailey, T.: Map management for efficient simultaneous localization and mapping (SLAM). Auton. Robots **12**, 267–286 (2002)
5. Chatterjee, A.: Differential evolution tuned fuzzy supervisor adapted, extended Kalman filtering for SLAM problems in mobile robots. Robotica **27**, 411–423 (2009)
6. Chatterjee, A., Matsuno, F.: A neuro-fuzzy assisted extended Kalman based approach for simultaneous localization and mapping (SLAM) problems. IEEE Trans. Fuzzy Syst. **15**, 984–997 (2007)
7. Murphy, K.: Bayesian map learning in dynamic environments. Neural Inform. Proc. Syst. **12**, 1015–1021 (2000)

ROSLAM—A Faster Algorithm for Simultaneous Localization and Mapping (SLAM)

Teng-Wei Huang, Chen-Chien Hsu, Wei-Yen Wang and Jacky Baltes

Abstract Computationally efficient SLAM (CESLAM) has been proposed to solve simultaneous localization and mapping problem in real-time design. CESLAM first uses the landmark measurement with the maximum likelihood to update the particle states and then update their associated landmarks later. This improves the accuracy of localization and mapping by avoiding unnecessary comparisons. This paper describes a modified version of CESLAM called rapidly operations SLAM (ROSLAM) which improves the runtime even further. We present an empirical evaluation of ROSLAM in a simulated environment which shows that it speeds up previous well known algorithms by 100 %.

Keywords FastSLAM · CESLAM · Particle filter · Extended Kalman filter

1 Introduction

Simultaneous localization and mapping (SLAM) is a crucial technique for robot navigation in unknown environments. SLAM is the problem where an intelligent robot moves and estimates a map of its environment as well as its pose relative to

T.-W. Huang (✉) · C.-C. Hsu · W.-Y. Wang
Department of Electrical Engineering, National Taiwan Normal University,
162 He-Ping East Rd., Sec. 1, Taipei, Taiwan
e-mail: andy02172001@hotmail.com; 60275035H@ntnu.edu.tw

C.-C. Hsu
e-mail: jhsu@ntnu.edu.tw

W.-Y. Wang
e-mail: wywang@ntnu.edu.tw

J. Baltes
Department of Computer Science, University of Manitoba, Winnipeg, MB R3T 2N2, Canada
e-mail: jacky@cs.umanitoba.ca

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_6

the map simultaneously [1]. Using landmark measurements obtained from the robot's sensors and odometry, we can determine the robot's position within a map. After localization we add unknown landmarks to complete the map.

2 Related Work

Since SLAM is a fundamental problem in robot navigation, there has been a lot of research in this field. There are two major SLAM algorithms: EKF-SLAM [2] and FastSLAM [3]. Both will be described in the following subsection.

2.1 EKF-SLAM

The EKF-SLAM is based on extended Kalman filter (EKF) that uses odometry and landmark measurements to calculate the required information to update the estimates for the position of the robot and the landmarks using an EKF. In addition, this method estimates a complete map with the robot adjusting every step of its calculated position. Ideally, EKF will rapidly converge when the system is linear. However, in real-world applications, most systems are nonlinear which lead to slow or no convergence.

Several solutions were proposed to improve EKF-SLAM. Focusing on map management, Dissanayake et al. proposed a method to delete unreliable landmarks to enhance the accuracy of map generation and robot's position [4]. Many soft-computing methods [5–7] are also proposed in order to improve EKF-SLAM. However, when the quantity of landmarks becomes larger, it may place a heavy load on the operating system due to the quadratic size—with respect to the number of landmarks—of the covariance matrix [3].

2.2 FastSLAM

To improve the runtime of EKF-SLAM, many FastSLAM algorithms were presented based on Rao–Blackwellized particle filters (RBPF) [8]. The FastSLAM algorithm subdivides the SLAM problems into two phases: mapping and localization. In mapping, FastSLAM uses EKFs to estimate; while for localization, the method uses particle filter (PF). The heavy burden from possible large matrix can be dismissed, as FastSLAM calculates every landmark for each EKF only.

There are two versions of FastSLAM, FastSLAM 1.0 [3] and FastSLAM 2.0 [1]. FastSLAM 1.0 only uses the control signal to localize the robot, while FastSLAM 2.0 uses the sensor measurements as well to improve localization and mapping result. FastSLAM 2.0 is more accurate than FastSLAM 1.0. As a trade-off, FastSLAM 2.0 runs slower than FastSLAM 1.0, especially with a large number of landmarks.

2.3 CESLAM

Therefore, CESLAM [9] was proposed to overcome all of the above problems. CESLAM combines the strong points of both FastSLAM 1.0 and FastSLAM 2.0. To predict a probabilistic position of the robot, CESLAM only uses the odometry, which reduces the runtime. Then, CESLAM will update its position using the maximum likelihood position of the sensor measurements and previous information, so that accuracy can be improved and runtime decreased.

However, there are some shortcomings in CESLAM. This paper presents a rapidly operating SLAM (ROSLAM) for mobile robot that can improve the runtime of CESLAM.

3 Rapidly Operation SLAM for Mobile Robots

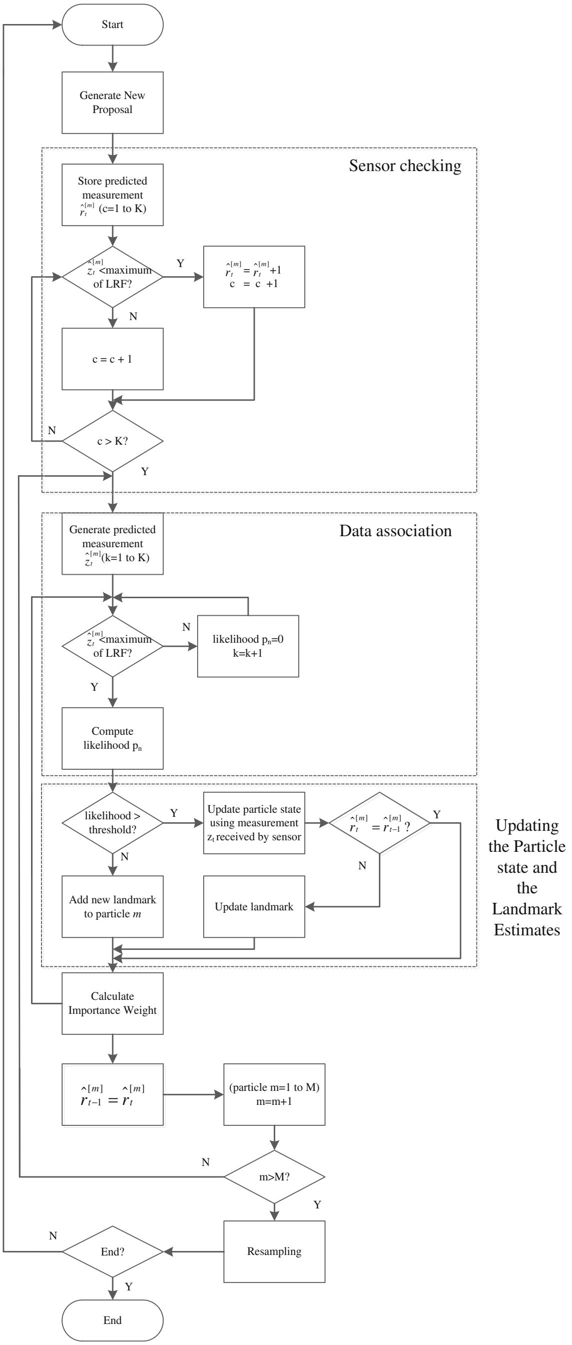
In order to improve the runtime, we propose ROSLAM which will use the number of landmarks to compare with the previous one and decide whether to skip updating the landmarks or not. Furthermore, in the data association phase, if the estimated measurements exceed the actual range of the laser range finder, ROSLAM will skip the step of calculating the likelihood. Then, the likelihood of these measurements will be set to zero like CESLAM. At the same time, we also check the likelihood of adding a new landmark or continue to the next step. The next step is updating the landmark estimates. Using the estimated measurements we update the state of particle first and then correct the landmarks. Figure 1 shows the flowchart of the proposed approach.

3.1 General New Proposal

For particle $m = 1, \ldots, M,$ considers the current measurements z^t and control inputs u^t to produce a possible state of robot

$$x_t^{[m]} \sim p\left(x_t | x_{t-1}^{[m]},\ z_t,\ u_t,\ n^t\right) \tag{1}$$

Fig. 1 Flowchart showing the ROSLAM


Start
Generate New Proposal
Sensor checking
Store predicted measurement $\hat{r}_t^{[m]}$ (c=1 to K)
$\hat{z}_t^{[m]}$ <maximum of LRF?
Y
$\hat{r}_t^{[m]} = \hat{r}_t^{[m]}+1$
c = c +1
N
c = c + 1
N
c > K?
Y
Data association
Generate predicted measurement $\hat{z}_t^{[m]}$(k=1 to K)
$\hat{z}_t^{[m]}$ <maximum of LRF?
N
likelihood p_n=0
k=k+1
Y
Compute likelihood p_n
likelihood > threshold?
Y
Update particle state using measurement z_t received by sensor
$\hat{r}_t^{[m]} = \hat{r}_{t-1}^{[m]}$?
Y
N
N
Add new landmark to particle *m*
Update landmark
Updating the Particle state and the Landmark Estimates
Calculate Importance Weight
$\hat{r}_{t-1}^{[m]} = \hat{r}_t^{[m]}$
(particle m=1 to M)
m=m+1
N
m>M?
Y
Resampling
N
End?
Y
End

3.2 Sensor Checking

By storing the number of landmark in current measurements

$$\hat{r}_t^{[m]} = r_t^{[m]} + 1 \tag{2}$$

3.3 Data Association

Using previous landmarks information $k = 1, \ldots, N$ and the state of particle in particle m to produce estimated measurements

$$\hat{z}_t^{[m]} = g\left(\hat{x}_t^{[m]},\ \theta_{k,\mathrm{t}-1}^{[m]}\right) \tag{3}$$

We set zero to the likelihood of the estimated measurement without further calculation, even when it is out of the LRF's range. Relatively, we compare the actual measurement Z_t and estimated measurement $\hat{z}_{k,t}$ with the likelihood via the following equation

$$p_k = \frac{1}{\sqrt{2\pi S_k}} \exp\left(-\frac{1}{2}\left(z_t - \hat{z}_{k,\mathrm{t}}\right)^T \left(S_k\right)^{-1}\left(z_t - z_{k,\mathrm{t}}\right)\right) \tag{4}$$

where S_k is the innovation covariance matrix

$$\begin{aligned} H_{\theta_k} &= \partial g\left(x_t^m, \theta_{k,t-1}^m\right) / \partial\theta_{k,t-1}^m \\ S_k &= H_{\theta_k} \sum_{k,t-1}^{[m]} H_{\theta_k}^T + R_t \end{aligned} \tag{5}$$

3.4 Updating the Landmark Estimates and Particle's State

If the likelihood is higher than the threshold, then the measurement already exists for the particle. Thus, we first complete updating the particle's state and then correct the landmark. Update the landmarks information using the estimated measurement that has the largest likelihood

$$\begin{aligned} H_{x_t} &= \frac{\partial g\left(x_t^m, \theta_{\hat{n},t-1}^m\right)}{\partial x_t^m} \\ \sum\limits_{x_t,t}^{[m]} &= [H_{x_t}^T (S_k)^{-1} H_{x_t} + (\sum\limits_{x_t,t}^{[m]})^{-1}]^{-1} \\ u_{x_t,t}^{[m]} &= u_{x_t,t-1}^{[m]} + \sum\limits_{x,t}^{[m]} H_{x_t}^T (S_k)^{-1} (z_t - \hat{z}_t^{[m]}) \end{aligned} \tag{6}$$

This procedure is the same as the CESLAM. After executing the above process, we can use the updated state of particle to produce new estimated measurement and revise the landmark estimated by EKF

$$\begin{aligned} K &= \sum\limits_{n_t,t-1}^{[m]} H_{\theta_{\hat{n}}}^T S_{\hat{n}}^{-1} \\ u_{\hat{n},t}^{[m]} &= u_{\hat{n},t-1}^{[m]} + K(z_t - \hat{z}_t^{[m]}) \\ \sum\limits_{\hat{n},t}^{[m]} &= (I - KH_{\theta_{\hat{n}}}) \sum\limits_{\hat{n},t-1}^{[m]} \end{aligned} \tag{7}$$

when there is no likelihood higher than the threshold add this landmark information into the particle instead of updating

$$\begin{aligned} N_t^{[m]} &= N_{t-1}^{[m]} + 1 \\ u_{\hat{n},t}^{[m]} &= g^{-1}(x_t^{[m]}, \hat{z}_{\hat{n},t}) \\ \sum\limits_{\hat{n},t}^{[m]} &= (H_{\theta_{\hat{n}}}^T R_t^{-1} H_{\theta_{\hat{n}}})^{-1} \end{aligned} \tag{8}$$

3.5 Calculating Importance Weight

The importance weight is assigned by the following:

$$\begin{aligned} w_t^m &= \frac{\text{target distribution}}{\text{proposal distribution}} \\ &= \frac{p\left(x^{t,[m]} | z^t, u^t, n^t\right)}{p(x^{t-1,[m]} | z^{t-1}, u^{t-1}, n^{t-1}) p(x^{[m]} | x^{t-1,[m]}, z^t, u^t, n^t)} \end{aligned} \tag{9}$$

3.6 Resampling

After every particle is assigned an importance weight, we calculate effective sample size (ESS) to decide whether to do the resampling or not [10]

$$\begin{aligned} \mathrm{ESS}_t &= \frac{M}{1 + cv_t^2} \\ cv_t^2 &= \frac{1}{M} \sum_{i=1}^{M} (Mw(i) - 1)^2 \end{aligned} \tag{10}$$

A tournament selection [11] mechanism is adopted for resampling while the ESS is lower than the threshold.

4 Simulation Results

For performance assessment, we compare FastSLAM 1.0 and FastSLAM 2.0 with the proposed approach. The experiments used a PC with an Intel Core i7-920 2.67 GHz CPU, 3 GB RAM, and the simulation platform was Visual Studio 2010 Express with the OpenCV library. The size of the map was 500 × 500 pixels. There were seventy landmarks on the map. The robot moved 5 pixels or rotated 15° in each step. We introduced Gaussian noise into the measurements obtained from the LRF and odometry. The maximum distance of the LRF is a hundred pixels and it has 180° scanning range, so it has 181 measurements for each scan. We calculated the mean robot pose error in localization result, and the robot moved along a prespecified path. Ten independent runs are executed in the simulation and each run requires 166 steps to complete the SLAM task.

Figures 2, 3 and 4, shows the simulation results of several SLAM algorithms, where the green point path is the localization results of the best particle, and black point path is the real trajectory of robot. The pink circles are the particles which overlap or are near the dark yellow circle. A dark yellow circle is the robot current position. The estimations on landmark position are the red hollow circles, and the black dots with the brown circle scattered on the map represent the landmark. The space within the blue painted sector is the range of simulation LRF, and the blue dots in this space represent the present measurements of the robot. As demonstrated in Figs. 2, 3 and 4, the accuracy of estimating landmarks and robot localization result in the proposed approach are better than the ones created by FastSLAM 1.0 and FastSLAM 2.0.

Fig. 2 Simulation results of FastSLAM 1.0

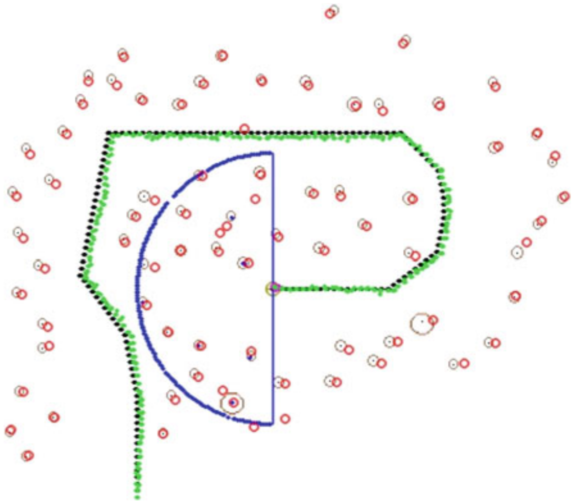
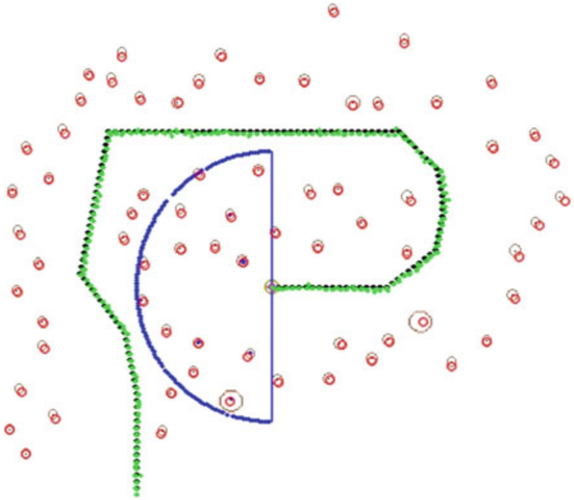


Fig. 3 Simulation results of FastSLAM 2.0



4.1 Runtime Efficiency and Localization Error

Table 1 shows the simulation results of 10 independent runs. The 'average localization error' means the error between the true and estimated robot's pose; 'Path RMSE' represents the RMSE between the true and estimated robot's pose. Table 2 shows the average runtime of various SLAM algorithms. When landmarks become larger, the advantage of the proposed approach can be clearly demonstrated.

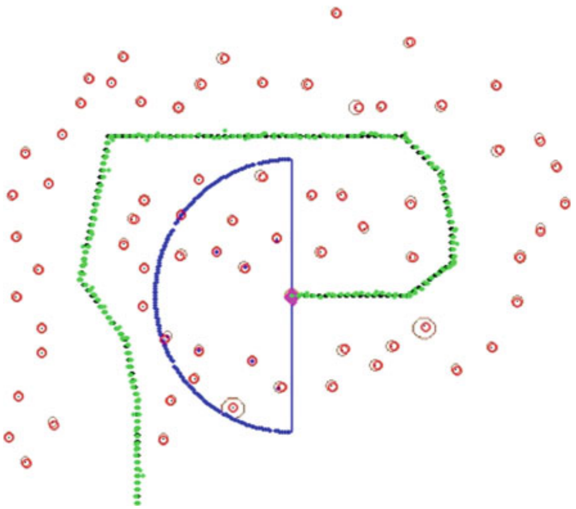


Fig. 4 Simulation results of ROSLAM

Table 1 Simulation localization result of several SLAM algorithms

SLAM algorithm	Average localization error (pixel)	Path RMSE (pixel)
FastSLAM1.0	4.17	2.563
FastSLAM2.0	2.37	1.398
ROSLAM	2.12	1.266

Table 2 Runtime comparison chart of several SLAM algorithms

SLAM algorithm	Runtime (s) (24 landmarks)	Runtime (s) (50 landmarks)	Runtime (s) (70 landmarks)
FastSLAM1.0	127.5	221.80	365.55
FastSLAM2.0	171.55	375.18	631.26
ROSLAM	46.95	54.52	69.17

5 Conclusion

In this paper, we modified the CESLAM to a new version that greatly improves its runtime. There ideas in this paper is to ignore the same landmarks while the robot stays in the same place so as to prevent obtaining additional measurements that add little information. As a result, wasted comparisons of measurements with all landmarks information in particles can be avoided to improve runtime performance in real-time applications.

Acknowledgments This research is partially supported by the “Aim for the Top University Project” and “Center of Learning Technology for Chinese” of National Taiwan Normal University (NTNU), sponsored by the Ministry of Education, Taiwan, R.O.C. and the “International Research-Intensive Center of Excellence Program” of NTNU and Ministry of Science and Technology, Taiwan, under Grants no. MOST 104-2911-I-003-301 and MOST 103-2221-E-003-027.

References

1. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1151–1156 (2003)
2. Smith, R.C., Cheeseman, P.: On the representation and estimation of spatial uncertainty. Int. J. Robot. **5**, 56–58 (1986)
3. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: Proceedings of AAAI National Conference on Artificial Intelligence, pp. 593–598 (2002)
4. Dissanayake, G., Williams, S.B., Durrant-Whyte, H., Bailey, T.: Map management for efficient simultaneous localization and mapping (SLAM). Auton. Robots **12**, 267–286 (2002)
5. Chatterjee, A.: Differential evolution tuned fuzzy supervisor adapted, extended Kalman filter-ing for SLAM problems in mobile robots. Robotica **27**, 411–423 (2009)
6. Chatterjee, A., Matsuno, F.: A neuro-fuzzy assisted extended Kalman based approach for simultaneous localization and mapping (SLAM) problems. IEEE Trans. Fuzzy Syst. **15**, 984–997 (2007)
7. Chatterjee, A., Matsuno, F.: A Geese PSO tuned fuzzy supervisor for EKF based solutions of simultaneous localization and mapping (SLAM) problems in mobile robots. Expert Syst. Appl. **37**, 5542–5548 (2010)
8. Murphy, K.: Bayesian map learning in dynamic environments. Neural Inform. Proc. Syst. **12**, 1015–1021 (2000)
9. Yang, C.-K., Hsu, C.-C., Wang,Y.-T.: Computationally efficient algorithm for simultane-ous localization and mapping (SLAM),” Proceedings of the IEEE International Conference Networking, Sensing and Control (ICNSC), pp. 328–332 (2013)
10. Liu et al. J.S.: A theoretical framework for sequential importance sampling and resampling. In: Doucet, A., de Freitas, N., Gordon, N.J. (eds.) Sequential Monte Carlo in Practice. Springer (2001)
11. Miller, B.L., Goldberg, D.E.: Genetic Algorithms, Tournament Selection, and the Effects of Noise. IlliGAL Report No. 95006, July 1995

Adaptive Computation Algorithm for Simultaneous Localization and Mapping (SLAM)

Da-Wei Kung, Chen-Chien Hsu, Wei-Yen Wang and Jacky Baltes

Abstract Computationally Efficient SLAM (CESLAM) was proposed to improve the accuracy and runtime efficiency of FastSLAM 1.0 and FastSLAM 2.0. This method adopts the landmark measurement with the maximum likelihood, where the particle state is updated before updating the landmark estimate. Also, CESLAM solves the problem of real-time performance. In this paper, a modified version of CESLAM, called adaptive computation SLAM (ACSLAM), as an adaptive SLAM enhances the localization and mapping accuracy along with better runtime performance. In an empirical evaluation in a rich environment, we show that ACSLAM runs about twice as fast as FastSLAM 2.0 and increases the accuracy of the location estimate by a factor of two.

Keywords Fastslam · CESLAM · Particle filter · Extended kalman filter

1 Introduction

Simultaneous localization and mapping (SLAM) is a crucial part of robot navigation in unknown environments. It is necessary, whenever a robot needs to locate itself and build a map of the environment simultaneously [1]. Using the landmark

D.-W. Kung (✉) · C.-C. Hsu · W.-Y. Wang
Department of Electrical Engineering, National Taiwan Normal University,
162 He-Ping East Rd., Sec. 1, Taipei, Taiwan
e-mail: davidkung727@gmail.com

C.-C. Hsu
e-mail: jhsu@ntnu.edu.tw

W.-Y. Wang
e-mail: wywang@ntnu.edu.tw

J. Baltes
Department of Computer Science, University of Manitoba, 77 Chancellor Ave,
R3T 2N2 Winnipeg, MB, Canada
e-mail: jacky@cs.umanitoba.ca

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_7

measurements and odometry of the robot, we can calculate the robot's position and build a map of the environment. There are two popular approaches to SLAM: EKF-SLAM [2] and FastSLAM [3]. The EKF-SLAM is based on extended Kalman filter (EKF), using landmark measurement and odometry information to calculate the EKF's covariance matrix, and then conducts the prediction and correction to finish the localization and mapping. But EKF-SLAM can only converge at a fast speed when the robot system is linear. Several attempts were proposed to address this problem for nonlinear systems. For example, Dissanayake et al. describe a strategy for deleting unsuitable landmarks to improve the accuracy of robot's position and landmark estimation [4]. There are some soft-computing approaches [5–7] that also enhance the EKF-SLAM algorithm. Due to computational complexity, increasing the number of landmarks results in squaring the size of the covariance matrix.

Section 2 presents an overview of related work. Section 3 presents a detailed description of the ACSLAM algorithm. Empirical results of several realistic environments are shown in Sect. 4. The paper concludes with Sect. 5.

2 Related Work

In order to speed up EKF-SLAM, some FastSLAM algorithms are proposed that use Rao-Blackwell normalization on particle filters (RBPF) [8]. The FastSLAM algorithm separates the SLAM problem into localization and mapping phases, where localization uses particle filters (PF) and the mapping uses EKF. Unlike EKF-SLAM, FastSLAM allows each landmark to be calculated by a single EKF, respectively. This can avoid the cost of dealing with large matrices. FastSLAM has two versions, FastSLAM 1.0 [3] and FastSLAM 2.0 [1]. FastSLAM 1.0 only uses odometry to estimate the robot's position, while FastSLAM 2.0 adds the recent sensor measurements to improve estimation accuracy. Though FastSLAM 2.0 seems better than FastSLAM 1.0, the run-time of FastSLAM 2.0 can increase dramatically when the number of landmarks increases.

CESLAM [9] is proposed to solve this problem. Taking advantage of FastSLAM 1.0, and FastSLAM 2.0, CESLAM improves the estimation accuracy and computational efficiency. In the beginning, CESLAM only uses odometry information to estimate the robot's position and updates the particle state and landmark information when a measurement has a maximum likelihood. This paper proposes a modified SLAM algorithm derived from CESLAM, called adaptive computation SLAM (ACSLAM) which improves runtime performance of the algorithm.

3 ACSLAM for Mobile Robots

The number of particles is a crucial parameter in FastSLAM 1.0, FastSLAM 2.0, and CESLAM. Too few particles will result in poor localization accuracy or aliasing problems, but too many particles will result in poor runtime, since the runtime grows linearly with the number of particles.

The proposed ACSLAM generates a probabilistic guess of the robot's position similar to CESLAM. The main difference between the new ACSLAM algorithm and CESLAM is that ACSLAM tries to reduce the runtime of the algorithm by dynamically changing the number of particles.

To improve the runtime, ACSLAM uses the effective sample size (ESS) [10] to decide the next generation's particle number. If the value of ESS is small, it means the difference between each particle's importance weight is large. Then, we will increase the particle number. If the value of ESS is large, it means the differences between each particle's importance weight is small. Next, we decrease the particle number. Figure 1 shows the flowchart of the proposed ACSLAM. Detailed descriptions are listed below:

3.1 Generate a New Proposal

For particle $m = 1, \ldots, M$, consider the control input u_t to generate a probabilistic guess of the robot's position

$$x_t^m \sim p\left(x_t | x_{t-1}^m, u_t\right) \tag{1}$$

3.2 Data Association

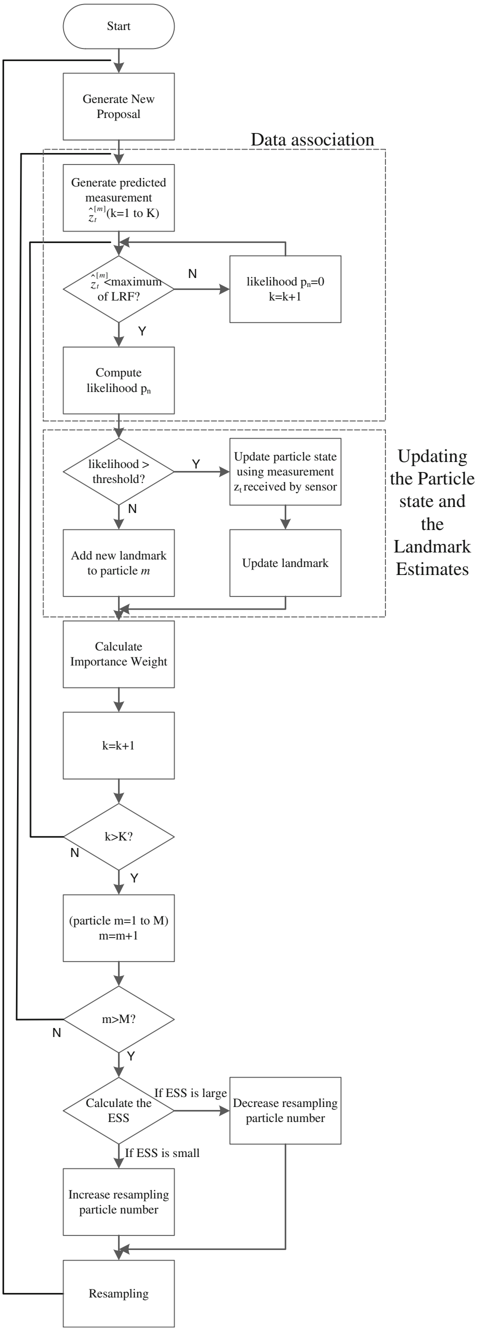
Use the robot's position and the information of landmarks $k = 1 \ldots, N$ in particle m to generate estimated measurement.

$$\hat{z}_t^m = g\left(\hat{x}_t^m, \theta_{k,t-1}^m\right) \tag{2}$$

If the estimated measurement is out of the laser range, then we will set the estimated measurement's likelihood as zero and skip further calculation. Otherwise, we use the estimated measurement to compute the likelihood by using the following equation.

$$p_k = \frac{1}{\sqrt{|2\pi S_k|}} \exp\left[-\frac{1}{2}(z_t - \hat{z}_{k,t})^T (S_k)^{-1} (z_t - \hat{z}_{k,t})\right] \tag{3}$$

Fig. 1 Flowchart showing the ACSLAM


Start
Generate New Proposal
Data association
Generate predicted measurement $\hat{z}_t^{[m]}$(k=1 to K)
$\hat{z}_t^{[m]}$<maximum of LRF?
N
likelihood p_n=0 k=k+1
Y
Compute likelihood p_n
likelihood > threshold?
Y
Update particle state using measurement z_t received by sensor
Updating the Particle state and the Landmark Estimates
N
Add new landmark to particle *m*
Update landmark
Calculate Importance Weight
k=k+1
k>K?
N
Y
(particle m=1 to M) m=m+1
m>M?
N
Y
Calculate the ESS
If ESS is large
Decrease resampling particle number
If ESS is small
Increase resampling particle number
Resampling

Where z_t is an actual measurement gathered by a sensor, $\hat{z}_{k,t}$ is the estimated measurement, and S_k is the innovation covariance matrix:

$$\begin{aligned} H_{\theta_k} &= \partial g\left(x_t^m, \theta_{k,t-1}^m\right)/\partial \theta_{k,t-1}^m \\ S_k &= H_{\theta_k} \Sigma_{k,t-1}^m H_{\theta_k}^T + R_t \end{aligned} \tag{4}$$

If any of the estimated measurement's likelihood is larger than the threshold, then the largest one will be used to update the landmark. If not, that means the landmark was not detected before, then we add it as a new landmark to the particle.

3.3 Updating the Particle State and the Landmark Estimates

As CESLAM, if the likelihood is larger than the threshold, this means the landmark has been recorded. Before updating the landmark, we have to use the value to correct the particle's state first. This is to make sure that our particle would not be updated to the wrong position.

$$\begin{aligned} H_{x_t} &= \partial g\left(x_t^m, \theta_{k,t-1}^m\right)/\partial x_t^m \\ \sum_{x_t,t}^{m} &= \left[H_{x_t}^T (S_k)^{-1} H_{x_t} + \left(\sum_{x_t,t}^{m}\right)^{-1}\right]^{-1} \\ u_{x_t,t}^m &= \sum_{x_t,t}^{m} H_{x_t}^T (S_k)^{-1}\left(z_t - \hat{z}_t^m\right) + u_{x_t,t-1}^m \end{aligned} \tag{5}$$

Next, we use the particle's state to calculate the new estimated measurement to update the landmark estimation by EKF.

$$\begin{aligned} K &= \sum_{n_t,t-1}^{m} H_{\theta_{\hat{n}}}^T S_{\hat{n}}^{-1} \\ u_{\hat{n},t}^m &= u_{\hat{n},t-1}^m + K\left(z_t - \hat{z}_t^m\right) \\ \sum_{\hat{n},t}^{m} &= \left(I - KH_{\theta_{\hat{n}}}\right) \sum_{\hat{n},t-1}^{m} \end{aligned} \tag{6}$$

We add the new landmark to the particle if there is no likelihood larger than the threshold.

$$\begin{aligned} N_t^m &= N_{t-1}^m + 1 \\ u_{\hat{n},t}^m &= g^{-1}\left(x_t^m, \hat{z}_{\hat{n},t}\right) \\ \sum_{\hat{n},t}^{m} &= \left(H_{\theta_{\hat{n}}}^T R_t^{-1} H_{\theta_{\hat{n}}}\right)^{-1} \end{aligned} \tag{7}$$

3.4 Calculating Importance Weight and Effective Sample Size

We calculate the importance weight by the following equation:

$$\begin{aligned} w_t^m &= \frac{\text{target distribution}}{\text{proposal distribution}} \\ &= \frac{p(x^{t,m} | z^t, u^t, n^t)}{p(x^{t-1,m} | z^{t-1}, u^{t-1}, n^{t-1}) p\left(x_t^m | x^{t-1,m}, z^t, u^t, n^t\right)} \end{aligned} \tag{8}$$

Then, we use the importance weight to calculate the effective sample size (ESS).

$$ESS = \frac{M}{1 + cv_t^2} \tag{9}$$

M is the particle number, cv_t^2 is the importance weight's coefficient of variation.

$$cv_t^2 = \frac{1}{M} \sum_{i=1}^{M} (Mw(i) - 1)^2 \tag{10}$$

According to the value of ESS, we decrease resampling particle number if ESS is large, and we increase the resampling particle number if ESS is small.

3.5 Resampling

After each particle is assigned an importance weight and we have the next generation's particle number, a tournament selection [11] mechanism is adopted for resampling.

4 Simulation Results

To evaluate the performance, we compare FastSLAM 1.0 and FastSLAM 2.0 with the proposed method. The software environment uses Visual Studio 2010 with the OpenCV library. The map can separate in sparse and compressed block, while the dimension is 500 * 500 pixels. A robot can move 10 pixels or rotate 15° per step. Since real world application of the odometry consists of errors, therefore, the Gaussian noise is introduced with distance and rotation. The simulation of laser range is 180°. The lasers and each landmark in the particle will generate 181 measurements. The maximum distance of the laser range finder is 100 pixels. A robot will need to move 100 steps along the path in each run.

In the proposed method, the maximum particle number we set is 20, and the minimum particle number we set is 10. Once we run the process, the average particle number is 16, then we set the other algorithm's particle number as 16.

Figure 2 depicts the simulation results of the three different SLAM algorithms. The pink points are the true path of the robot trajectory, and gray points are the localization from the best particle. The green circles are the particle, and the yellow circle which is overlapped with the convergent green particles is the robot. The black dots with brown circle are the landmarks, and the red circles are estimations of landmarks. The laser range scanning is shown by blue lines. As the simulation shown in Fig. 2, the robot's localization and landmark estimation in the ACSLAM are more accurate than FastSLAM 1.0 and FastSLAM 2.0. Figure 3 shows the relation between each step and the localization error. Apparently, the compressed block, three algorithms' localization errors are small before counting to the forty fifth step. But after the forty fifth step, the proposed method can adjust the particle number according to the environment as the landmark becomes sparse. As a result, the proposed method outperforms others.

Table 1 shows the localization results of the simulation in different SLAM algorithms. The 'localization error' represents the error between estimated and true robot's position; 'localization RMSE' means the RMSE between estimated and true robot's position. In the table, the performance of ACSLAM in localization is better than FastSLAM 1.0 and FastSLAM 2.0.

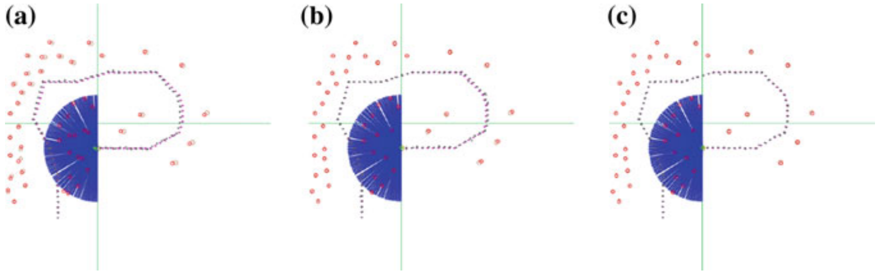


Fig. 2 Simulation results of different SLAM algorithms. **a** FastSLAM 1.0. **b** FastSLAM 2.0. **c** ACSLAM

Fig. 3 Simulation's localization error

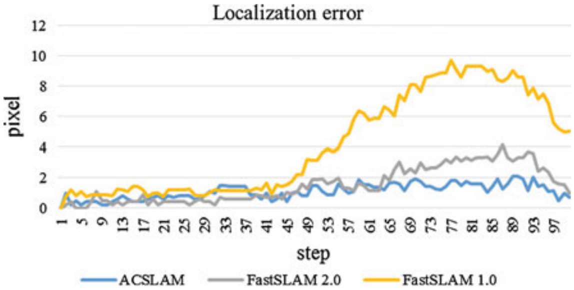


Table 1 Simulation localization results of different SLAM algorithms

SLAM algorithms	Localization error (pixel)	Localization RMSE (pixel)
FastSLAM 1.0	4.17	3.26
FastSLAM 2.0	1.51	1.16
ACSLAM	1.09	0.59

Table 2 Runtime efficiency of different SLAM algorithms

SLAM algorithms	Runtime (s)
FastSLAM 1.0	62.32
FastSLAM 2.0	100.67
ACSLAM	58.84

Table 2 shows the runtime efficiency in different SLAM algorithms. Since the computational time can be reduced, the proposed method performs better than FastSLAM 1.0 and FastSLAM 2.0.

5 Conclusion

In this paper, we propose an extension of CESLAM. The goal is to improve the accuracy, when the particle's measurement has a maximum likelihood in the known landmark. We update the particle state before updating landmark estimation. The proposed algorithm also uses the effective sample size to adjust the particle calculation number so that we can improve the runtime efficiency. As a result, the proposed ACSLAM adopts the advantages of FastSLAM 1.0, FastSLAM 2.0, and CESLAM to ensure better estimation accuracy and runtime in the simulations.

Acknowledge This research is partially supported by the "Aim for the Top University Project" and "Center of Learning Technology for Chinese" of National Taiwan Normal University (NTNU), sponsored by the Ministry of Education, Taiwan, R.O.C. and the "International Research-Intensive Center of Excellence Program" of NTNU and Ministry of Science and Technology, Taiwan, under Grants no. MOST 104-2911-I-003-301 and MOST 103-2221-E-003-027.

References

1. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1151–1156 (2003)
2. Smith, R.C., Cheeseman, P.: On the representation and estimation of spatial uncertainty. Int. J. Robot. **5**, 56–58 (1986)

3. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: Proceedings of AAAI National Conference on Artificial Intelligence, pp. 593–598 (2002)
4. Dissanayake, G., Williams, S.B., Durrant-Whyte, H., Bailey, T.: Map management for efficient simultaneous localization and mapping (SLAM). Auton. Robots **12**, 267–286 (2002)
5. Chatterjee, A.: Differential evolution tuned fuzzy supervisor adapted, extended Kalman filtering for SLAM problems in mobile robots. Robotica **27**, 411–423 (2009)
6. Chatterjee, A., Matsuno, F.: A neuro-fuzzy assisted extended Kalman based approach for simultaneous localization and mapping (SLAM) problems. IEEE Trans. Fuzzy Syst. **15**, 984–997 (2007)
7. Chatterjee, A., Matsuno, F.: A geese PSO tuned fuzzy supervisor for EKF based solutions of simultaneous localization and mapping (SLAM) problems in mobile robots. Expert Syst. Appl. **37**, 5542–5548 (2010)
8. Murphy, K.: Bayesian map learning in dynamic environments. Neural Inf. Proc. Syst. **12**, 1015–1021 (2000)
9. Yang, C.-K., Hsu, C.-C., Wang, Y.-T.: Computationally efficient algorithm for simultaneous localization and mapping (SLAM). In: Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC), pp. 328–332 (2013)
10. Liu, J.S., et al.: A theoretical framework for sequential importance sampling and resampling. In: Doucet, A., De Freitas, N., Gordon, N.J. (eds.) Sequential Monte Carlo in Practice. Springer (2001)
11. Miller, B. L., Goldberg, D. E.: Genetic Algorithms, Tournament Selection, and the Effects of Noise, IlliGAL Report No. 95006 (1995)

Interactive Markerless Augmented Reality System Based on Visual SLAM Algorithm

Luybov Shuvalova, Alexander Petrov, Vsevolod Khithov and Igor Tishchenko

Abstract The problem of development a sustainable efficient and accurate augmented reality system possessing the interactivity property and basically implementing markerless approach for supplementing the real scene is addressed in this paper. Custom implementation is based on applying of feature-based SLAM algorithm, which is resistant to change light conditions, able to track both textured and non-textured objects in real-time mode achieved by introducing a once-performed offline system teaching step. The proposed solution is universal because of the system's configurability in accordance with the desirable result.

Keywords AR • Augmented reality • SLAM • ROS • Marker detection • Lua script • Intelligent system • Automatic initialization • 3D model • Interactivity • Real-time • Monocular camera

1 Introduction

Augmented reality is one of the most promising modern IT-technology's development directions. The technology of augmented reality is a technology of real-world scene's supplement with virtual models by integrating them in the

L. Shuvalova · A. Petrov
NPP SATEK Plus, Rybinsk, Russia
e-mail: lifedetermined@mail.ru

A. Petrov
e-mail: gmdidro@gmail.com

V. Khithov
Soloviev Rybinsk State Aviation Technical University, Rybinsk, Russia
e-mail: vskhitkov@gmail.com

I. Tishchenko (✉)
Program System Institute of Russian Academy of Sciences,
Pereslavl-Zalessky, Russia
e-mail: igor.p.tishchenko@gmail.com

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_8

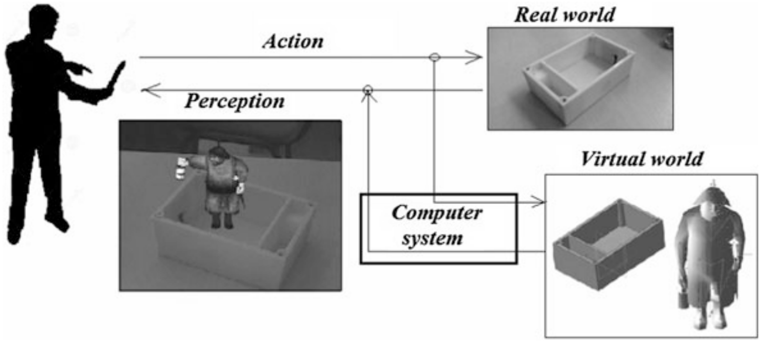


Fig. 1 The structure of augmented reality system

physical environment in real time with the help of computer devices and related software [1].

The Augmented reality (AR) System is the intermediary between a man and the perceived world and should produce a signal for one or more senses as a result of its functioning (see Fig. 1). They point out a visual, audio, and audio-visual AR-systems by the criteria of information representation type [2]. Depending on the generated system's output it is focused on the visual, auditory perception, or its combination.

The principle of virtual objects in the real scene formation is realized by system's obtaining the information about the environment with help of different sensors as the devices which are capable of recording various physical quantities.

Depending on the user's ability to influence the behavior of virtual objects and their response, and the degree of user's involvement in the interaction, virtual elements can be passive and interactive. Interactive systems allow the user to modify the virtual environment in an active way: to manipulate virtual objects at the structure level, to change their model properties, behavior and position in the scene.

The efforts of specialists in the field of computer vision and augmented reality nowadays are focused on the development of an efficient algorithm implementation of a mobile optical augmented reality system as one of the most promising direction of IT—technology's development with broad prospects for further applications.

We offer a possible approach to markerless implementation of an interactive augmented reality system in this article. Its purpose is the video stream supplement with the pre-prepared visual effects in real-time mode. The supplemented scene or object of interest in view of the specifics of the alleged application is represented by the architectural layout or someone else with the known 3D-model.

It is assumed that the system should be composed of the following components: a software tool for building scenarios and 3D-models preparation for future applying in the system and a demo application that detects and tracks the supplemented object of the interest in the video stream and integrate the virtual objects using the calculated absolute camera pose in each frame.

The proposed approach is supported by ROS implementation.

The interactive augmented reality system can be used in area of marketing communications and sales (for example, to increase the demand for products or for awakening the interest of potential buyers, investors, and so on), training and after-sales service (to demonstrate the flow of a complicated process such as the palpation of the femoral artery or aerodynamic processes in a vehicle, airplane, the movement of electrons in conductors, etc.).

1.1 Related Works

There are a number of systems that functional or technical specifications close to the proposed system, which found a successful application in the previously mentioned areas.

Works [3–6] offer solutions based on placed in the real environment marker's spatial position tracking. The article [7] is devoted to the development of a fast and efficient method of finding a marker in the video stream. Marker approach is considered to be the most studied and well known. There are even books [8] devoted to theoretical and applied aspects of augmented reality systems. However, apart from the obvious advantages of presented in these papers marker approach, such as the reliability, ease of implementation, the effectiveness of the solution, there are significant drawbacks. Placing a marker in the workspace of the user can be intrusive, the virtual objects are visible only in case of marker visibility and the markers can from time to time need recalibration. Also in the case of overlapping a part of a marker the virtual object is not displayed. There are also restrictions on the system working distance.

In article [9] an algorithm for textured or non-textured 3D geometric primitives (cubes, cylinders) recognizing and tracking in real-time mode is described. The BLORT [9] library does not require markers and is focused on its application in robot navigating tasks. The restriction of its application involves "complex" cases of working with objects, which can change their size, shape, color. There is a stage of online learning, which is an object sequential rotation (for the edges detection and tracking) in the monocular camera field of vision. The advantages of an approach consist in capability of both textured (is preferred) and non-textured objects recognition, moreover, the possibility of target loss for nondurable time. However, the object of an interest can be presented only by static 3D geometric primitive. Another one disadvantage is that it has online learning phase, which is required for each session.

There are solutions [10] in area of augmented reality based on markerless approach by applying a PTAM (Parallel Tracking and Mapping) algorithm as a basis. PTAM is a well-known method of mapping by tracking the position of the camera in observed point features space. The advantage of the approach is the absence of a priori knowledge about the environment and lack of the phase of data preprocessing. Disadvantages are the low accuracy of recognition and tracking for non-textured objects, dense calculation of the optical flow, excessive number of map feature points, and the lack of topological structure of the map. Also, the presence of repetitive textures and angular features in a field of camera view leads to gross localization errors, including mutual localization one.

Work [11] contains an algorithm which is not based on finding points to navigate, but on the depth map building. The algorithm shows improved resistance to occlusion due to the use of weighing, and the presence of a buffer to maintain a constant number of keyframes. As disadvantages we note that due to the inverse relation of depth and displacement, resolution of stereo vision systems operating on the basis of this method is better at close range, and worse at far. Also the whole system operation is not performed in real time at 15 fps frequency and high-density map.

The result of a partnership of French company Diotasoft, specializing in augmented reality, and the French research institute CEA (Commissariat à l'énergie atomique) is an effective algorithm for 3D markerless CAD data-based tracking, which is successfully used in industry sphere to demonstrate the industrial complex objects with virtual supplement. The implementation of markerless tracking technology is organized on the basis of constraint SLAM (Simultaneous Location and Mapping) algorithm. constraint SLAM is an efficient real-time algorithm for precise camera localization having a priori information about the environment (3D model of supplemented scene) [12]. The authors provide nonlinear refinement of structure from motion reconstruction by taking advantage of a partial knowledge of the environment [13]. The main disadvantage of the approach is that the first camera pose is defined by the user. It must be close to the real camera pose, so that the registration of the generated line model can be successful [14].

2 System's Organization

The proposed system's implementation is a feature-based one. Feature is a characteristic image attribute, the local neighborhood of which makes it possible to distinguish it from any other feature of the image. The features of the image as part of our implementation are FAST-angles with correspondingly associated descriptors. Descriptor here is a vector of numerical characteristics of the feature's local area.

We use DIRD (bit) descriptors [15], the main advantage of which is that they can be quickly evaluated and compared. Moreover, these descriptors are more invariant to light changing conditions than gray 256, BRIEF, USURF [15]. Since each of these descriptors is a bit vector, the measurement of the distance between the two vectors can be done by counting the number of different bits (Hamming distance), which may in practice be implemented using XOR operations.

Being extracted from a set of images, features are placed to the recognition database. Each extracted feature vector represents a point in the space of high dimension (256 for DIRD descriptors). Points characterized by close proximity to each other (small value of Hamming distance), determine the similarity of the images. It is rather useful to organize the recognition database using a hierarchical structure for fast and efficient search for similar objects in it.

We resort to the use building of a visual words vocabulary at the preparatory stage of system's functioning in order to achieve efficiency of images searching by criterion of semantic similarity. Visual words vocabulary [16] is a hierarchically structured tree consisting of nodes, each of which is a visual word—a numerical representation of certain image key fragment obtained by the DIRD descriptor quantization. Quantization is carried out by clustering the space of features using hierarchical k-means method. Visual words vocabulary of our system is constructed in the form of a tree with 6 tiers and 10 branches per each tier. Visual words vocabulary formation can be a prolonged process, so it takes place at the preparatory stage (offline learning stage) of the system's functioning.

We use the algorithm for simultaneous localization and mapping with the absence of prior knowledge about the environment—ORB-SLAM in order to know at any given time the camera pose relatively to supplemented scene (and vice versa, for future augmented reality building), to ensure the system's ability to relocalization, to carry out high-quality tracking of features. ORB-SLAM is a versatile and fairly accurate algorithm for simultaneous navigation and map building with the open-source implementation, presented by Raul Mur-Artal, JMM Montiel, and Juan D. Tardos in early 2015 [17].

The main advantage of the algorithm we see in the use of covisibility graph appear in systems such as SLAM. That gives an indication of the keyframes correlation level, permits frames, and their characteristics comparing, promotes the possibility of local tracking regardless of the global map, thereby speeding up the search. The structure was proposed by the British researcher in area of computer vision—Strasdat [18]. Covisibility graph is undirected weighted graph, each node of which is the keyframe. An edge between two nodes exists if keyframes share map points observations. The weight of the edge is the number of points in common. We use the recognition database as proposed by ORB-SLAM authors. This allows to perform images comparison only in that case when they have common words with the query image. It speeds up the processing of the search operation significantly.

The proposed approach to the implementation of an interactive markerless augmented reality system system's functioning is carried out in two stages: preparatory stage (offline learning stage) and the main stage of the functioning.

The block diagram of the system and its components is shown in Fig. 2.

The complete cycle system's functioning involves a single execution of the preparatory stage for the structured data on supplemented scene creating and the main stage, which is responsible for camera pose estimating and in augmented reality rendering in real-time mode. Real-time performance is achieved by the presence of a priori analyzed data about the environment (at the preparatory stage) and active their using at the main stage of functioning.

The preparatory stage involves the following steps:

1. the obtaining of internal camera calibration parameters and distortion coefficients;
2. manual creating of configuration file with augmented reality system's settings (script file that contains information about a supplemented object, handlers of user actions and paths to corresponding lua-scripts (.lua), which allow to modify the model) and the organization of virtual storage facilities;

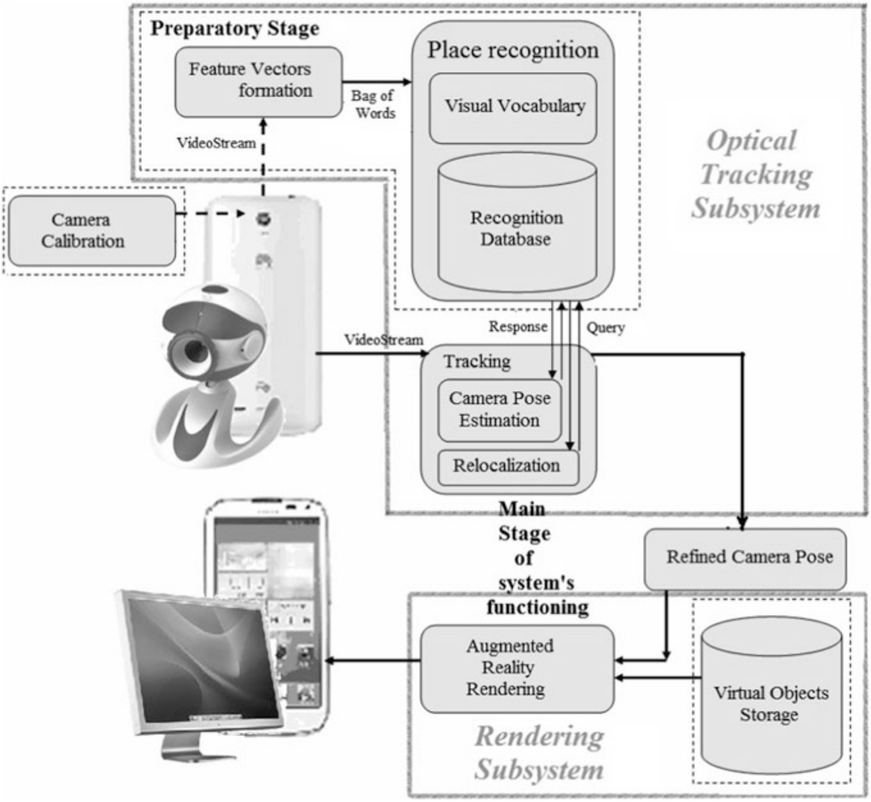


Fig. 2 The block diagram of the system's structure

3. the visual words vocabulary formation using the sample images of supplemented scene with help of tools developed by us;
4. the offline system's learning stage.

During the preparatory stage system processes the incoming video stream, using the generated previously visual words vocabulary and settings, detects features, place them in an internal representation, full-fills and improves the map of the scene and estimates the camera pose relatively supplemented scene in each moment of time. We shall have already got everything to calculate the absolute camera pose by the time the main stage of functioning is active. This becomes possible due to temporary (only on offline learning stage) involving the marker into the scene and setting its accurate position relatively to the scene in the configuration file.

Key stages of the system's running on both stages are the following.

1. Tracking. It is a process of finding a solution to the problem of determining the location of the tracked objects at each time moment. At this stage features of the image are also are found, the estimation of objects position by analyzing their

previous pose and motion model is performed, system initialization and relocalization (the tracked objects are out of camera scope or an insufficient number of key features) take place in order to provide a quality tracking.

2. Local mapping. It is the mapping and maintaining it optimal in terms of memory allocation under the structures and points' statistical significance process. The system's functioning at this stage involves adding keyframes, organization of statistically significant samples of points, adding points to the map, the implementation of the local bundle adjustment and the exclusion of keyframes with the redundant information.
3. Loop closing (reducing additive error). System identifies the situation when it has already analyzed the current conditions and is able to execute the correct handling, having the aim of keeping the additive error accumulated over the cycle. The detection of frame candidates for processing the loop closing, the calculation of the similarity transformation between the current frame and the frame candidate, loop closing and removal of duplicated points of the map, optimization of keyframes graph are performed at this stage.
4. Refined camera pose. It is the final estimate of camera absolute position relative to the object of the interest estimated from the established 3D points correspondences in the global coordinate system and 2D image points for stream of images. The task of camera pose estimation is reduced to the search for external calibration parameters (rotation matrix and transfer vector) which describe the data matching in a best way.

Rendering is performed only at the stage of main functioning. At this stage the integration of objects from the virtual effects storage in the real environment takes place taking into account the geometric parameters of the model, estimated camera pose, orientation of the image and the user's actions aimed at the interactive elements.

The result of the preparatory stage is a serialized state of the system—the saving of the internal representation of camera movement and observed objects history. Saving includes the constructed map of the area, covisibility graph and other structures which by the system's booting at the main stage (in the absence of the marker) would help to avoid time-consuming system's initialization. This became possible by obtaining the correction matrix calculated at the preparatory stage. The correction matrix is a matrix of transition from the SLAM coordinate system to the object of interest (scene) coordinate system.

Thus, if the preparatory stage aims SLAM functioning (environment observing, "memorizing" and analysis), receipt of the correction matrix, at the stage of the main functioning SLAM subsystem gives the exact position of the scene relatively to the camera pose. So that knowing the relative data about virtual content pose in the scene, we can carry out augmented reality rendering, having its calculated absolute position.

Iterative result of the main stage in normal conditions is a 2D image of the scene with correctly integrated augmented reality observed by the user on the computing device screen. Full result of the main stage can be considered as monocular camera optical flow supplement in real-time mode and interactive virtual objects' behavior according to wanted output.

3 System's Components

Our interactive augmented reality system can be divided into two main subsystems: a SLAM-subsystem and rendering augmented reality subsystem. The purpose of using SLAM-subsystem is to obtain an absolute camera position in object of interest coordinate system at each moment. SLAM-subsystem functioning is a parallel implementation of three streams: tracking, mapping, and loop closing. Rendering subsystem is designed to integrate predefined computer graphics objects into the real scene.

Figures 3 and 4 show the components of the system (the schema of the preparatory stage and the main stage's schema, respectively) and the relationship between them.

Let us consider the algorithm of obtaining the correction matrix at the preparatory stage. Correction matrix refers to the transition matrix from the SLAM coordinate system to the object of interest coordinate system.

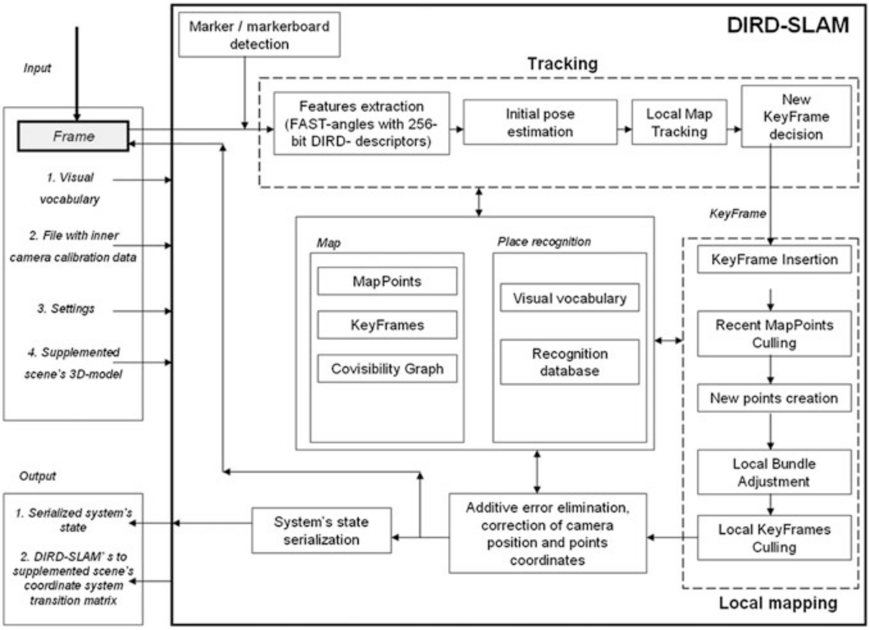

DIRD-SLAM
Input
Frame
1. Visual vocabulary
2. File with inner camera calibration data
3. Settings
4. Supplemented scene's 3D-model
Marker / markerboard detection
Tracking
Features extraction (FAST-angles with 256-bit DIRD- descriptors)
Initial pose estimation
Local Map Tracking
New KeyFrame decision
KeyFrame
Map
MapPoints
KeyFrames
Covisibility Graph
Place recognition
Visual vocabulary
Recognition database
KeyFrame Insertion
Recent MapPoints Culling
New points creation
Local Bundle Adjustment
Local KeyFrames Culling
Local mapping
Output
1. Serialized system's state
2. DIRD-SLAM' s to supplemented scene's coordinate system transition matrix
System's state serialization
Additive error elimination, correction of camera position and points coordinates

Fig. 3 The schema of the preparatory stage of AR system's functioning

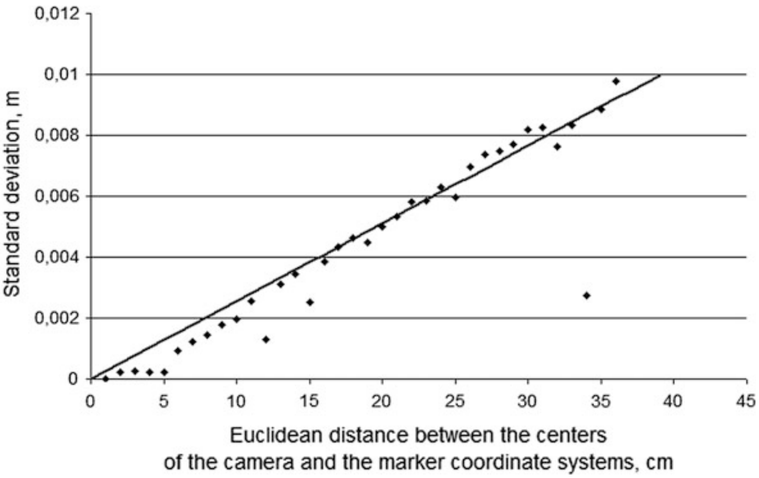


Fig. 4 Dependence of standard deviation of measurements on the Euclidean distance between the centers of the camera and the marker coordinate systems

The system is considered to have already been initialized, i.e., we know camera rotation matrix R_{slam} and transfer vector T_{slam} relatively to the center of SLAM coordinate system at any time. And by receiving a video sequence of a certain length and its analysis the scale of the map in the internal representation of SLAM has already become permanent (it scales with the size of the map, since pose and structure estimation are decoupled).

Denote that "m" is a designation of the marker coordinate system, "cam"—camera coordinate system, "slam"—SLAM coordinate system, "obj"—supplemented scene coordinate system. The notation of the type "obj1_in_obj2" should be interpreted as spatial position (rotation, translation, rotation and translation) of the object "obj1" in the coordinate system of the object "obj2."

When a next frame coming in it is analyzed (using ArUco library [19]) in order to detect a special marker with the specified identifier.

If the process is successful (i.e., rotation matrix $R_{m\_in\_cam}$ and transfer vector $T_{m\_in\_cam}$ are obtained) the system will solve the problem of providing a more accurate measurement of the marker position in the camera coordinate system. We apply second-order Kalman filter to reduce the impact of noise and to obtain a more accurate assessment of the present position $T_{m\_in\_cam}$' having a number of inaccurate measurements. The expected value and standard deviation are calculated experimentally. Thus, Fig. 4 shows how the standard deviation of measurements depends on the Euclidean distance between the centers of the camera and the marker coordinate systems.

It is necessary to obtain a matrix defining an unambiguous marker position in the coordinate system of the camera (it is already calculated), and the matrix defining uniquely the position of the marker in the object of interest coordinate system (quaternion $RT_{m\_in\_obj}$) to determine the position of the object of interest in the camera coordinate system. The position $T_{m\_in\_obj}$ is defined by the user depending on the placement of the marker relatively to the center of the model coordinate system. The rotation matrix $R_{m\_in\_obj}$ is set by three Euler angles, depending on the orientation of the marker relatively to the center of the model coordinate system.

Then the matrix that defines camera position in the scene (object of interest) coordinate system is calculated as follows:

$$\begin{bmatrix} R_{\mathrm{cam\_in\_obj}} & T_{\mathrm{cam\_in\_obj}} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{\mathrm{m\_in\_obj}} & T_{\mathrm{m\_in\_obj}} \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} R_{\mathrm{m\_in\_cam}} & T_{\mathrm{m\_in\_cam}} \\ 0 & 1 \end{bmatrix}^{-1} \tag{1}$$

Then we fix two trajectories: the camera trajectory in SLAM coordinate system (transfer vector in $RT_{cam\_in\_slam}$, calculated by SLAM subsystem) and the camera trajectory in scene coordinate system (the transfer vector in $RT_{cam\_in\_obj}$). The resulting correction matrix dRT (as the transition from SLAM coordinate system to scene coordinate system) is found by using the OpenCV implementation of affine transformation between two corresponding clouds of 3D points estimating—estimateAffine3D(). We also tried to find a rigid transformation using the method based on SVD decomposition of the covariance matrix, but the results obtained were less accurate.

ModelView matrix which is necessary on the main stage of system's functioning for augmented reality rendering is calculated as follows:

$$\mathrm{ModelView} = RT_{\mathrm{cam\_in\_slam}} * dRT^{-1} \tag{2}$$

4 System Application Component

Figure 2 shows a block diagram of the system's components interaction at the main stage of system's functioning. SLAM subsystem reports the rendering subsystem camera position in the SLAM coordinate system for each analyzed frame. Rendering subsystem using data (from the configuration files) of the relative mutual position of the real scene and virtual objects (3D models, possibly animated) processes the supplementing 3D models rotating them at the desired angle, placing in a desired position relatively to the scene and calculating the scope using a stencil buffer method. After that rendering subsystem imposes obtained in the internal memory image of a supplementing object with alpha mask on the analyzed by SLAM subsystem frame. Augmented reality rendering subsystem performs loading models in MD5 format, their placement in the internal system's representation, provides configuration files parsing, and executes lua-scripts (Fig. 5).

System supports MD5 3D data import at current stage of developing. MD5 allows you to save an animated three-dimensional objects in two files: a file with the extension .md5mesh, storing data about vertices, polygons, vertices' weight

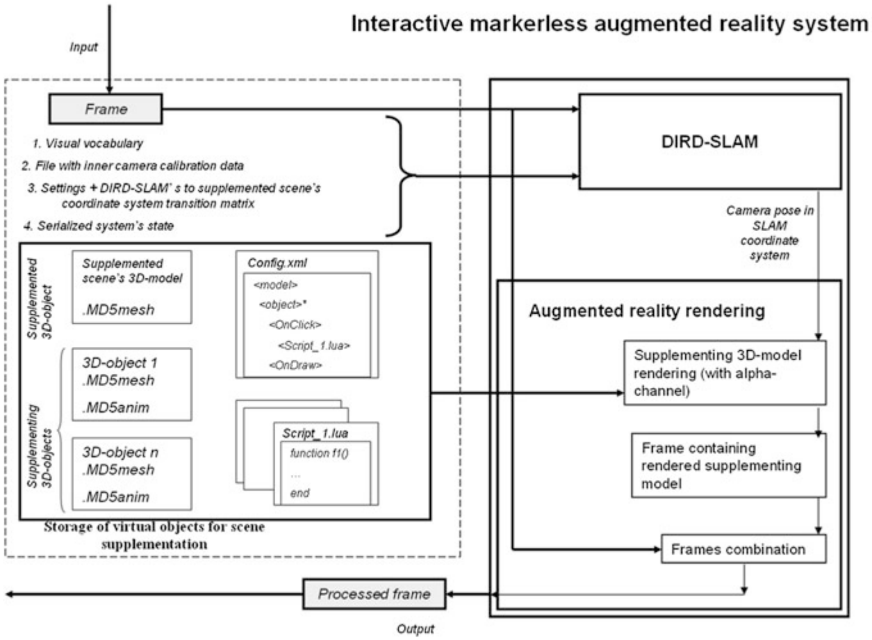
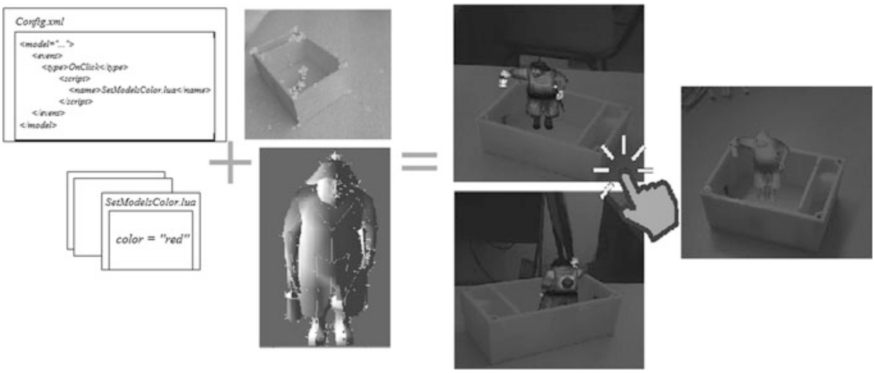

Interactive markerless augmented reality system
Input
Frame
1. Visual vocabulary
2. File with inner camera calibration data
3. Settings + DIRD-SLAM' s to supplemented scene's coordinate system transition matrix
4. Serialized system's state
Supplemented 3D-object
Supplemented scene's 3D-model
.MD5mesh
Config.xml
<model>
<object>*
<OnClick>
<Script_1.lua>
<OnDraw>
Supplementing 3D-objects
3D-object 1
.MD5mesh
.MD5anim
3D-object n
.MD5mesh
.MD5anim
Script_1.lua
function f1()
...
end
Storage of virtual objects for scene supplementation
DIRD-SLAM
Camera pose in SLAM coordinate system
Augmented reality rendering
Supplementing 3D-model rendering (with alpha-channel)
Frame containing rendered supplementing model
Frames combination
Processed frame
Output

Fig. 5 The schema of the main stage of AR system's functioning

coefficients, determining the geometric properties of polyhedral object, vertices, and textures (TGA, DDS) relations, the name of the shader, and file with the extension . md5anim, which contains information about the skeletal animation for model in . md5mesh. File .md5anim stores data about the hierarchy of the model's skeleton with the flags of states for the nodal points for animation, data about the limiting cube for each frame of animation, the basic skeleton of the main frame, which is necessary to calculate the position of skeleton's nodes for individual animation frames, etc. Skeletal animation with weighted bones of 3D-models is implemented by rendering into the vertex buffer. Each model's vertex can be linked with multiple bones in accordance with the concept of skeletal animation (as opposed to the concept of normal skeletal animation). Each bone is associated with some weight as the degree of its influence on vertice's movement. Thus, the greater weight of bone causes the greater translation value for related vertices. Skeletal animation with weighted bones allows you to animate even the smooth surfaces curves.

Rendering into the vertex buffer is a method according to which constructing a texture using a fragment shader and copying the resulting data to the vertex buffer becomes possible. We use this method in order to minimize the number of data transfers (from the CPU to the GPU), that is quite an expensive operation, and be able to build vertex arrays directly on GPU.

For providing its functioning system needs the script files which contain the handling code, written in Lua language, for supplementing virtual models



```
<model="...">
    <event>
        <type>OnClick</type>
            <script>
                <name>SetModelsColor.lua</name>
            </script>
    </event>
</model>
```

```
color = "red"
```

Fig. 6 The main idea of system's functioning

processing. At this stage of development lua-scripts provide functionality of changing the color of supplementing MD5-model (see Fig. 6), changing its size, setting the starting point of the animation, if any, and setting a text label relatively to the object.

5 Experiments

The experiments for ROS implemetation of interactive markerless AR system were performed on a personal computer with the × 86 processor Pentium (R) Dual-Core CPU with 2.00 GHz frequency and a performance index of 3.3. The amount of installed RAM is 3.00 GB. Monocular camera Microsoft LifeCam Studio with HD video quality is used for taking pictures with 640 × 480 pixels a resolution at frame rate of 30 frames per sec.

We provide (as part of the experiments) in terms of natural light in office room the yellow non-textured box supplement with a computer graphics object—animated three-dimensional model of the forester ("Bob with lamp" model), located inside the box. The experiments have proved that the proposed system works in real-time mode under the certain conditions and carries out fairly realistic rendering of three-dimensional virtual objects into the scene. The system successfully fulfills relocalization with serialized data volume of about 15 MB.

The system has a robust localization, however, based on the number of tests it can be concluded that the camera localization accuracy with one marker (2 cm * 2 cm) and estimateAffine3D-method is not sufficient to sustain the correct display of supplementing elements of the scene. Distance error is up to 1.5 cm.

The Fig. 7 below shows the camera trajectories for one of the test cases: blue line shows the camera trajectory in SLAM coordinate system, red—in yellow box coordinate system, green shows the reconstructed trajectory in SLAM coordinate system.

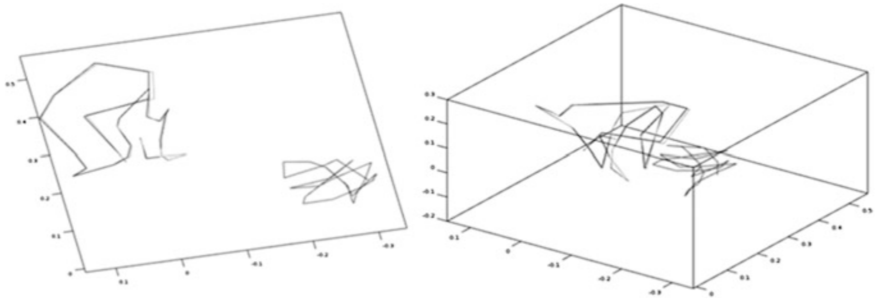


Fig. 7 Three camera trajectories, estimated by system: *red*—camera's movement in yellow box coordinate system (CS), *blue*—in SLAM CS, *green*—in SLAM CS (reconstructed one)

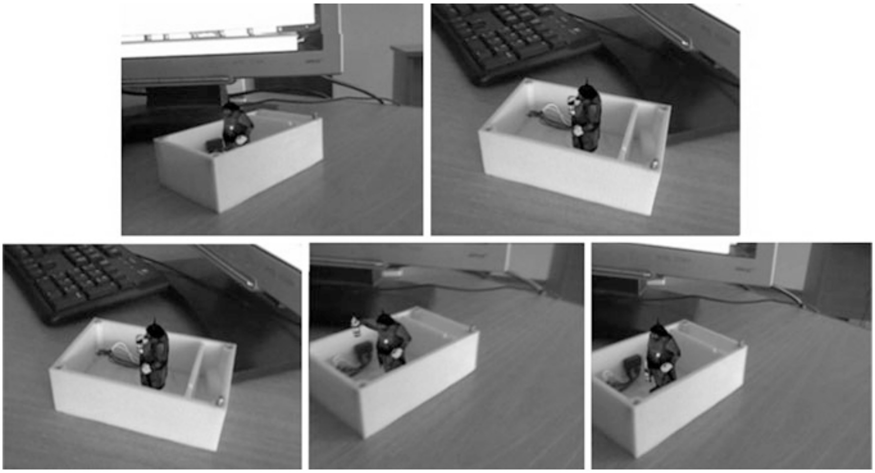


Fig. 8 Interactive markerless AR system's results

The following illustration (see Fig. 8) shows the output of the application for that test case and calculated correction matrix—frames from the video sequence with integrated into the real scene virtual object.

6 Mobile Phone Implementation

We assume that system's functioning in real-time mode could be achieved on a mobile device, taking into account the following implementation moments.

1. Features (FAST-angles) should satisfy Shi-Tomasi criterion of good features to track on first three levels of the image pyramid. Angles as the image features

according to Harris are fragments, for which the second derivatives of the autocorrelation matrix have two sufficiently large eigenvalues. According to Shi-Tomasi criterion, feature tracking is appropriate while the smaller of the two eigenvalues is greater than a preset threshold [20].

2. Each keyframe should be represented in five different spatial scales in contrast to the algorithm for PC with more processing power and 8-levels image pyramid.
3. When the camera rapidly changes its spatial position even the first-level image of the pyramid may be too vague. Then, the search is performed only on the second level, and the error threshold for a successful pattern matching is reduced. This strategy allows the system to keep track of a moderately quick turns, but frames tracked in this way are inaccurate and are never used as the keyframes.
4. It is essential to reduce the upper limit of the adaptive threshold variation, which determines the number of the key features extracted from the cell grid, superimposed on the image for a sustainable tracking of objects in the scene.

7 Conclusion

The development of an interactive augmented reality system based on markerless technology is the result of the related sources studying affecting the theoretical and applicating aspects of modern technology of augmented reality systems' implementation. The existing computer vision algorithms were analyzed, and a review of existing analogues of the proposed system was made including the assessment of their capacity in comparison with each other and our system based on economic, technical, mathematical criteria. The developed system's prototype has such advantages as a combination of marker technology's precision and reliability of the localization and markerless principle of building an augmented reality in the end-use, resistance to occlusion (partial or complete visibility overlap of one object by another), the ability to detect and track both textured and non-textured objects, to work in real-time mode.

As a future directions of project development it is possible to eliminate the use of a marker at the stage of preprocessing by comparing 3D points' clouds generated by analyzing the model in the computer memory and streaming video from the monocular camera. We are going to use the provided three-dimensional model of the scene as much as it is possible, including the bundle adjustment's algorithm. We also plan the use of edge-based method and particles filter for detection and tracking. Also it is expected to increase the number of supported 3D-data formats in order to enhance user experience.

Acknowledgments This work was supported by the Ministry of Education and Science of the Russian Federation (RFMEFI60714X0088) agreement for a grant on 'Development of methods and means of processing and intelligent image analysis and flow of data obtained from a set of stationary and mobile sensors, using high-performance distributed computing for the tasks of monitoring the indoor placement and adjacent outdoor territories'.

References

1. Olwal, A.: Unobtrusive augmentation of physical environments: interaction techniques, spatial displays and ubiquitous sensing. Doctoral Thesis, KTH, Department of Numerical Analysis and Computer Science, Trita-CSC-A. ISSN 1653-5723; 2009:09
2. Kostaras, N.N., Xenos, M.N.: Assessing the usability of augmented reality systems. In: Proceedings, 13th Panhellenic Conference on Informatics, PCI2009, pp. 197–201. Corfu, Greece (2009)
3. Neumann,U., Cho, Y.: A selftracking augmented reality system. In: Proceedings of the ACM Symposium on Virtual Reality and Applications, pp. 109–115 (1996)
4. Kato, H., Billinghurst, M.: Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In: Proceedings of IWAR. San Francisco, CA, USA (1999)
5. Ferrin, F.J.: Survey of helmet tracking technologies. In: Large-Screen Projection, Avionic and Helmet-Mounted Displays, vol. 1456, pp. 86–94. SPIE (1991)
6. Pentenrieder, K., Meier, P., Klinker, G.: Analysis of tracking accuracy for single-camera square-marker-based tracking. In: Proceedings Dritter Workshop Virtuelle und Erweiterte Realitt der GI-Fachgruppe VR/AR. Koblenz, Germany (2006)
7. Hirzer, M.: Marker Detection for Augmented Reality Applications. Technical Report ICG–TR–08/05 Graz (2008)
8. Siltanen, S.: Theory and applications of marker-based augmented reality. Espoo 2012. VTT Science 3 (2012)
9. Mörwald, T., Prankl, J., Richtsfeld, A., Zillich, M., Vincze, M.: Blort-the blocks world robotic vision toolbox. In: Proceedings of ICRA Workshop Best Practice in 3D Perception and Modeling for Mobile Manipulation (2010)
10. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proceedings of Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR' 07). Nara, Japan, (2007)
11. Schöps, T., Engel, J., Cremers, D.: Semi-dense visual odometry for ar on a smartphone. In: International Symposium on Mixed and Augmented Reality (2014)
12. Tamaazousti, M., Gay-Bellile, V.S., Collette, N., Bourgeois, S., Dhome, M.: Real-time accurate localization in a partially known environment: Application to augmented reality on textureless 3d objects. In: The 2nd International Workshop on AR/MR Registration, Tracking and Benchmarking (2011)
13. Tamaazousti, M., Gay-Bellile, V., Naudet-Collette, S., Bourgeois, S., Dhome, M.: Nonlinear Refinement of Structure from Motion Reconstruction by Taking Advantage of a Partial Knowledge of the Environment. CVPR (2011)
14. Wuest, H., Stricker, D.: Tracking of industrial objects by using cad models. J. Virtual Reality Broadcast. **4**(1) (2007)
15. Lategahn, H., Beck, J., Kitt, B., Stiller, C.: How to learn an illumination robust image feature for place recognition. In: IEEE, 2013 Intelligent Vehicles Symposium (IV), pp. 285–291. IEEE (2013)
16. Galvez-L´opez, D., Tard´os, J.D.: Bags of binary words for fast ´ place recognition in image sequences. IEEE Trans. Robot. **28**(5), 1188–1197 (2012)

17. Mur-Artal, R., Montiel, J.M.M., Juan, D.: Tardos ORB-SLAM: a Versatile and Accurate Monocular SLAM System. Instituto de Investigacion en Ingenierıa de Aragon Universidad de Zaragoza. Spain (2015)
18. Strasdat, H., Davison, A.J., Montiel, J.M.M., Konolige, K.: Double window optimisation for constant time visual SLAM. In: IEEE International Conference on Computer Vision (ICCV), pp. 2352–2359. Barcelona, Spain (2011)
19. Munoz-Salinas, R., Garrido-Jurado, S.: Aruco library. http://sourceforge.net/projects/aruco/ (2013). Accessed 01 Dec 2013
20. Shi, J., Tomasi, C.: Good features to track the Euclidean distance between the centers of the camera and the marker coordinate systems. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR94). Seattle (1994)

Control Strategy Design for Throw-in Challenge in a Humanoid Robot Soccer Game

Ya-Fang Ho, Ping-Huan Kuo, Kiah-Yang Chong and Tzuu-Hseng S. Li

Abstract In this paper, a control strategy of RoboCup throw-in technical challenge for small-sized humanoid robot is proposed. Throw-in is one of the technical challenges of RoboCup competition, which is a well-known robot competition held annually in different countries. To complete this challenge, the robot must possess object detection ability, stable walking ability, ball-holding ability, and ball-throwing ability. Hence, a small-sized humanoid robot, aiRobots-V, is presented to accomplish the throw-in challenge. Moreover, the control strategy and the abilities mentioned above are introduced. The experiment results demonstrate the performance of the proposed method. Furthermore, aiRobots-V won the second place in this technical challenge of RoboCup soccer game.

Keywords Ball-throwing · Humanoid robot soccer game · Throw-in challenge

1 Introduction

Bill Gates, the leader of the PC revolution, predicted that robotics will be the next hot field in the world in Scientific American January 2007, and it is evident that robotics is one of the most researchable fields in present days. In fact, robotics has been developed and functioned well in the industry for decades, but it gets more

Y.-F. Ho · P.-H. Kuo · K.-Y. Chong · T.-H.S. Li (✉)
Department of Electrical Engineering, National Cheng Kung University,
Tainan, Taiwan, R.O.C
e-mail: thsli@mail.ncku.edu.tw

Y.-F. Ho
e-mail: makinosakuya@hotmail.com

P.-H. Kuo
e-mail: coll22000@hotmail.com

K.-Y. Chong
e-mail: kriz.chong@gmail.com

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_9

and more related to our life time recently. Moreover, there are many kinds of robots that have been developed, and one of the most exciting types of robots is humanoid robots, because they have similar structure as human beings and have the potential to adapt to the human environment as human beings. Many small humanoid robot kits such as KHR, Robonova, Bioloid, and Nao can be purchased in the market easily, and are used for entertaining and educating.

To stimulate the development of multifunction humanoid robot, there are two well-known robot competitions held annually in the world, namely FIRA Robo-World Cup [1] and RoboCup [2]. Both of them contain several challenge events to encourage research in practical such as stable walk pattern generation [3–7], real-time image processing, intelligent decision making system, and fully autonomous ability.

Throw-in is one of the technical challenges in RoboCup competition. In this challenge, not only a stable walking ability is required, but the robot must be able to manipulate a ball with both hands. Through this challenge, the multifunction of the robot can be clearly demonstrated and validated. Hence, this paper presents the implementation and a control strategy for the throw-in technical challenge.

2 A Brief Review of Basic Image Processing

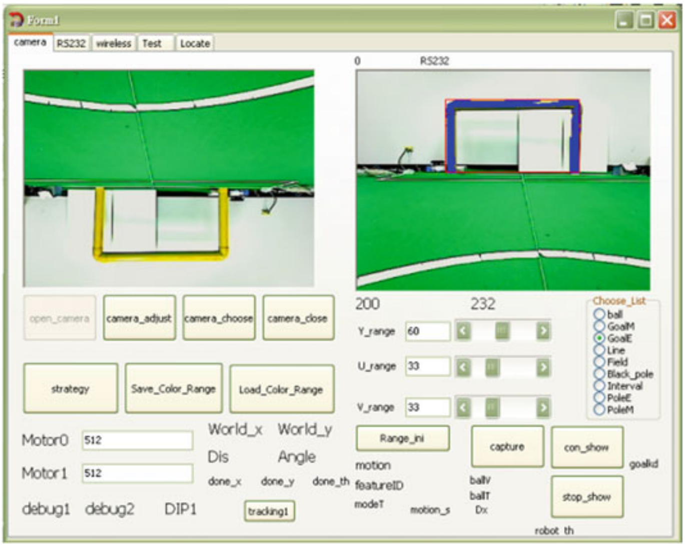
2.1 Color Space

There are many color spaces that can be used for image processing. Users choose the most suitable color space depending on different circumstances. In this paper, continuous image data is captured and sent in a format of Motion Joint Photographic Experts Group (MJPEG), and then each single frame is transformed into Bitmap (BMP) format, which is a typical image data format in Windows OS. Image data of BMP is represented by RGB color space. However, RGB space is easily influenced by variation of brightness. On the game field, lighting condition is not guaranteed and the shadow of robots may also affect the brightness of objects. Hence, the data is transformed again into YUV color space which is more robust for the illumination changing. The resolution of each image input is 320 × 240. A fast transformation method without any floating computation is used and shown in (1) [8].

$$\begin{cases} Y = (9798R + 19235G + 3736B) \gg 15 \\ U = 18514(B - Y) \gg 15 + 128 \\ V = 713(R - Y) \gg 15 + 128 \end{cases} \tag{1}$$

where “$\gg$” is a shift-right operator. It replaces the division operation.

The objects on the game field have their distinguish organization of colors. It is a fast way to segment them via the difference of colors. All the color ranges are set



Y_range 60

U_range 33

V_range 33

Motor0 512

Motor1 512

Choose_List: ○ ball ○ GoalM ● GoalE ○ Line ○ Field ○ Black_pole ○ Interval ○ PoleE ○ PoleM

Fig. 1 The operation of color setting interface

via the human–machine interface as shown in Fig. 1. The definition of the color range of *Y*, *U*, and *V* is done by [8]

$$\begin{cases} Y\max = Y + Y\_\text{range\_num} \\ Y\min = Y - Y\_\text{range\_num} \end{cases} \tag{2}$$

$$\begin{cases} U\max = U + U\_\text{range\_num} \\ U\min = U - U\_\text{range\_num} \end{cases} \tag{3}$$

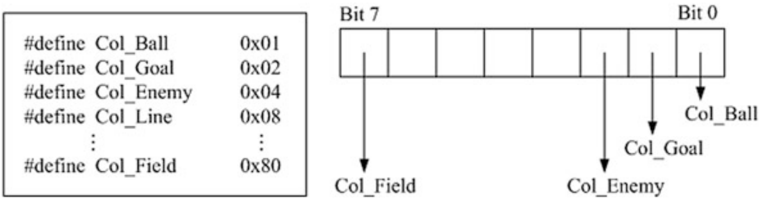
$$\begin{cases} V\max = V + V\_\text{range\_num} \\ V\min = V - V\_\text{range\_num} \end{cases} \tag{4}$$

where these color ranges are all bounded to [0–255]. The six values: *Y*max, *Y*min, *U*max, *U*min, *V*max, *V*min, are the color regions of each channel in *YUV* color space of the target object. It is also the regions that we are interested in. *Y*_range_num, *U*_range_num, *V*_range_num are the number of range that can be determined by users. So, the next step is to segment the object whose *YUV* range is in that region.

2.2 Image Segmentation

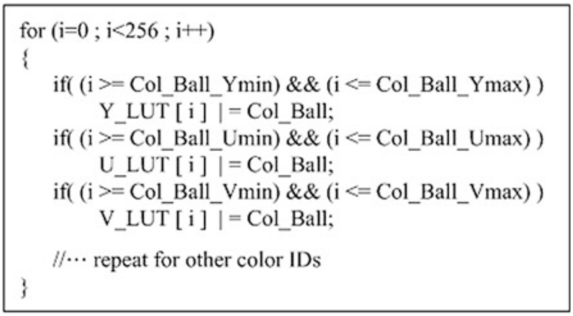
In order to segment the color in the specified regions, a *YUV* look-up table [8–10] is built. The captured image data are compared with the look-up table only using logical calculations and then the object segmentation by color will be finished. The diagram of building *YUV* look-up table is shown in Figs. 2, 3 and 4, respectively, and the logical computation used to identify the object is shown in Fig. 5.

In this way, the required objects of the right color can be segmented by scanning each frame of image data pixel by pixel. Figure 6 shows an illustration of segmentation method. The segmentation result is shown in Fig. 7.



```
#define Col_Ball      0x01
#define Col_Goal      0x02
#define Col_Enemy     0x04
#define Col_Line      0x08
          ⋮             ⋮
#define Col_Field     0x80
```

Fig. 2 Defining color IDs



```
for (i=0 ; i<256 ; i++)
{
    if( (i >= Col_Ball_Ymin) && (i <= Col_Ball_Ymax) )
        Y_LUT [ i ] |= Col_Ball;
    if( (i >= Col_Ball_Umin) && (i <= Col_Ball_Umax) )
        U_LUT [ i ] |= Col_Ball;
    if( (i >= Col_Ball_Vmin) && (i <= Col_Ball_Vmax) )
        V_LUT [ i ] |= Col_Ball;

    //··· repeat for other color IDs
}
```

Fig. 3 Building the look-up table with object color IDs

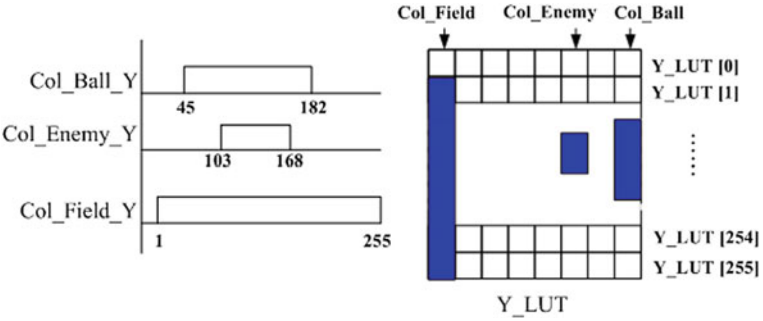
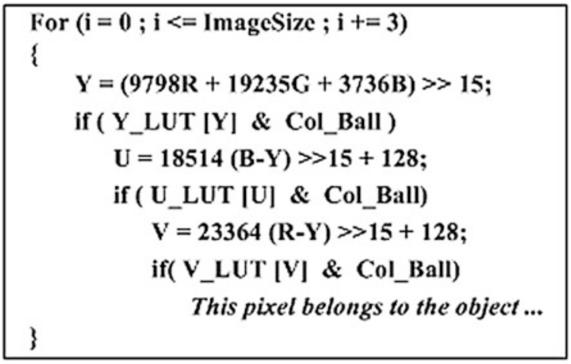


Fig. 4 An illustration of the LUT information



```
For (i = 0 ; i <= ImageSize ; i += 3)
{
      Y = (9798R + 19235G + 3736B) >> 15;
      if ( Y_LUT [Y]  &  Col_Ball )
          U = 18514 (B-Y) >>15 + 128;
          if ( U_LUT [U]  &  Col_Ball)
              V = 23364 (R-Y) >>15 + 128;
              if( V_LUT [V]  &  Col_Ball)
                  This pixel belongs to the object ...
}
```

Fig. 5 An example of object color segmentation using YUV look-up table

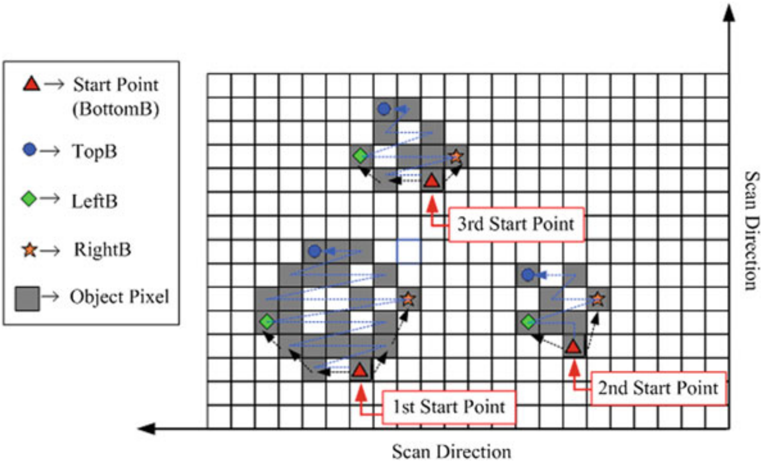


Fig. 6 An illustration of the segmentation method

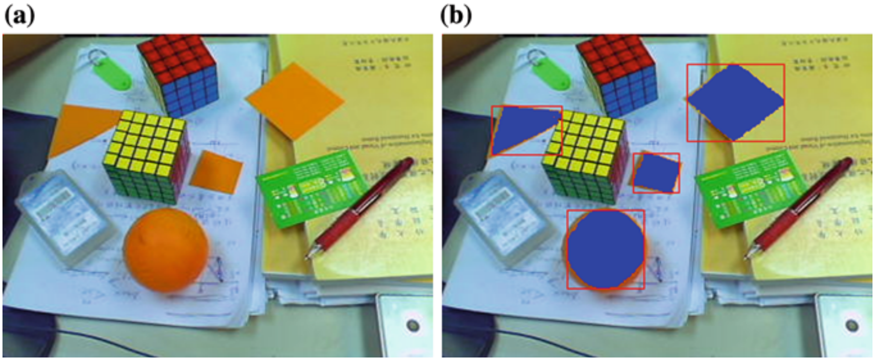


Fig. 7 An example of filtering out the color of the object with YUV look-up table. **a** The original image. **b** The result of segmentation

3 Control Strategy of Throw-in Technical Challenge

In a throw-in technical challenge, a humanoid robot should participate as a human soccer player that can walk and pick up the ball which is out of the field, and then throw the ball into the field to continue the game. In this challenge, the robot starts in the center standing on the center point and facing one of the two goals. The ball is placed 1.5 times the ball's diameter outside the field around one of the end points of the halfway line. The ball is placed at least 30 cm away from the nearest pole. Then, the robot must go to the ball, move with it towards the touch line, pick it up with its hands, and throw it back from there into the field. The thrower must face the field of play and has part of each foot either on the touch line or on the ground outside the touch line while it is throwing. The thrower must use one or both hands and delivers the ball from behind and over its head. The robots are ranked by the distance the ball has been thrown into the field. It is measured on the line connecting the robot position at throw-in and the final position of the ball. This distance is taken from the intersection of this line with the center of the touch line.

An example trajectory of the robot and the ball in a throw-in technical challenge is shown in Fig. 8. A possible motion of the ball is indicated by a gray, dash-dotted line, and the line with double arrows shows the distance that is representing the result of this throw-in.

The flow chart of the strategy is shown in Fig. 9. First of all, aiRobots-V is searching the ball while standing in the center of the field. If ball is not found, the robot turns left and searches the ball again until it finds out the ball. When the ball is found, aiRobots-V checks the pole near the ball to determine the direction of turn around the ball to face the goal in the following motion. Then, the robot walks to the ball and turns around it to face to the opponent goal. After turning, aiRobots-V looks down and makes sure the ball is in the proper position. Then, the direction of the opponent goal is checked again before picking up the ball. When the ball is in the proper position and the goal is in the proper direction, aiRobots-V picks up the

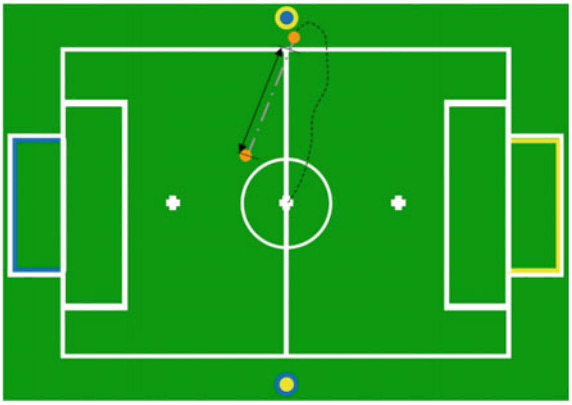


Fig. 8 An example trajectory of the robot and the ball in a throw-in technical challenge [11]

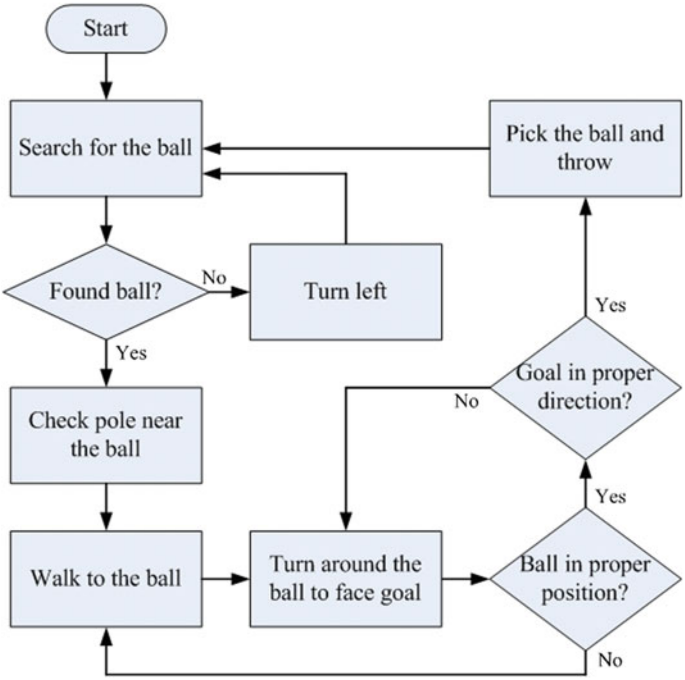


Fig. 9 The flow chart of strategy in a throw-in technical challenge

ball and throws it into the field. In order to make sure the task can be completely done, the robot restarts the strategy with searching the ball to check if the ball has been thrown into the field.

4 Experimental Results

4.1 Robot Specification

The robot used in the experiments, aiRobots-V, is the fifth generation humanoid robot developed in aiRobots laboratory [12]. aiRobots-V weighs 3.6 kg and is 55 cm tall, and consists of 21 DOF. Moreover, aiRobots-V is equipped with two different controllers. One is PICO 820 Series All-In-One PICO ITX Board with Windows XP OS, and it is responsible for image processing and decision-making. The other is Arduino MEGA 2560 microcontroller, and it is responsible for motion control and sensor measurement. The appearance of aiRobots-V is shown in Fig. 10, and its specification is listed in Table 1. The walking pattern of aiRobots-V is generated by Fuzzy Policy Gradient Gait Learning (FPGL) method [12, 13], and the walking speed can achieve 162.27 mm/s.

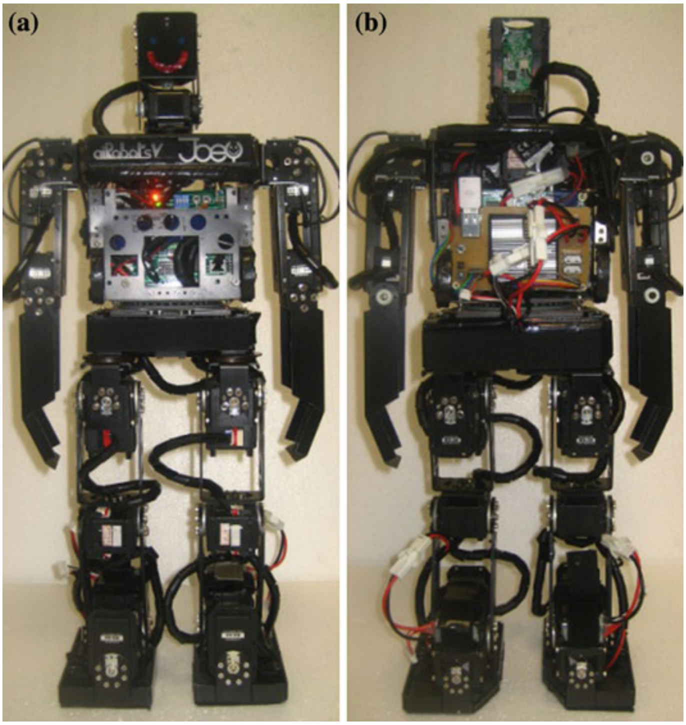


Fig. 10 The appearance of aiRobots-V. **a** Front view. **b** Rear view

Table 1 The specification of aiRobots-V

Height	55 cm
Width	24 cm
Depth	15 cm
Weight	3.6 kg
Size of Foot	11.6 cm × 7.6 cm
Sensors	Accelerometer/CMOS vision sensor
Degree of freedom	21 DOF
Material of structure	A6061 aluminum-magnesium alloy
Actuators	Dynamixel AX-12/RX-28/RX-64
Controller	PICO820 Series All-In-One PICO ITX Board/Arduino MEGA
Battery	Li-Po battery, 1300 mah, 11.1 V/14.8 V

4.2 Throw-in Experiments in Laboratory

Figure 11 is the trajectories of the robot in a throw-in technical challenge that is tested in laboratory. The best trial of aiRobots-V in the trials of throw-in technical

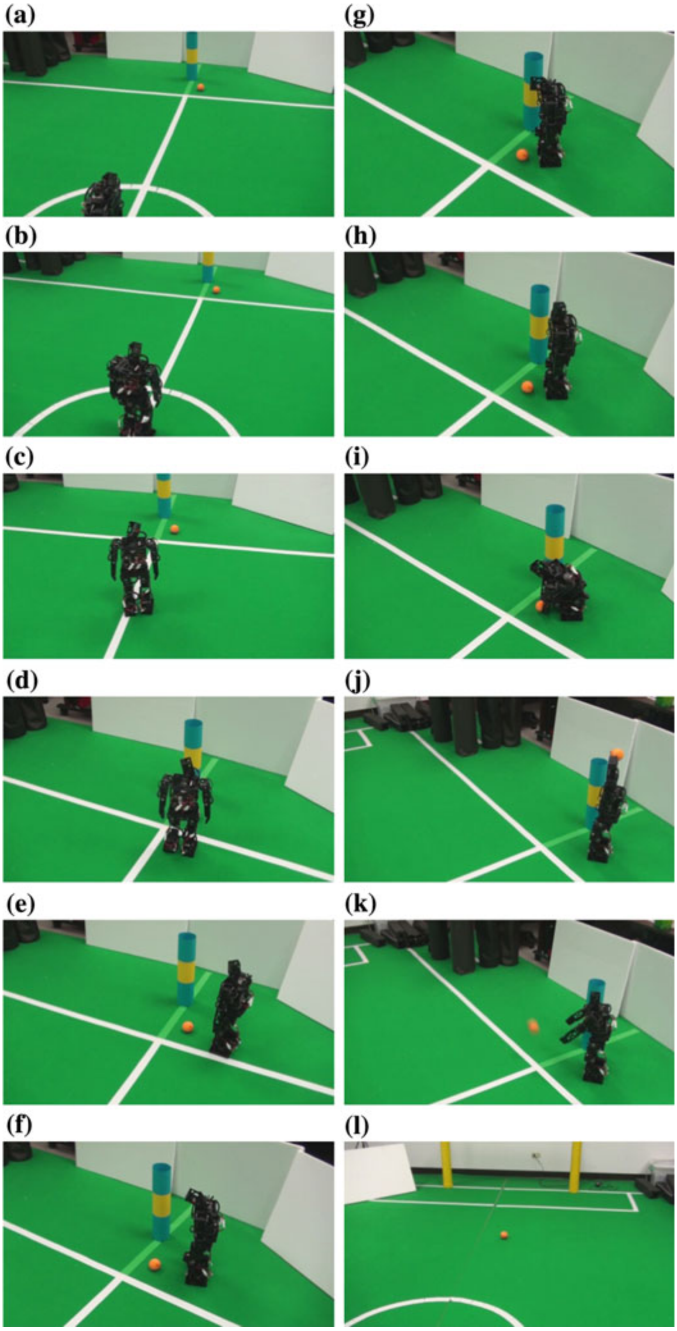


Fig. 11 The snapshots of throw-in technical challenge in laboratory

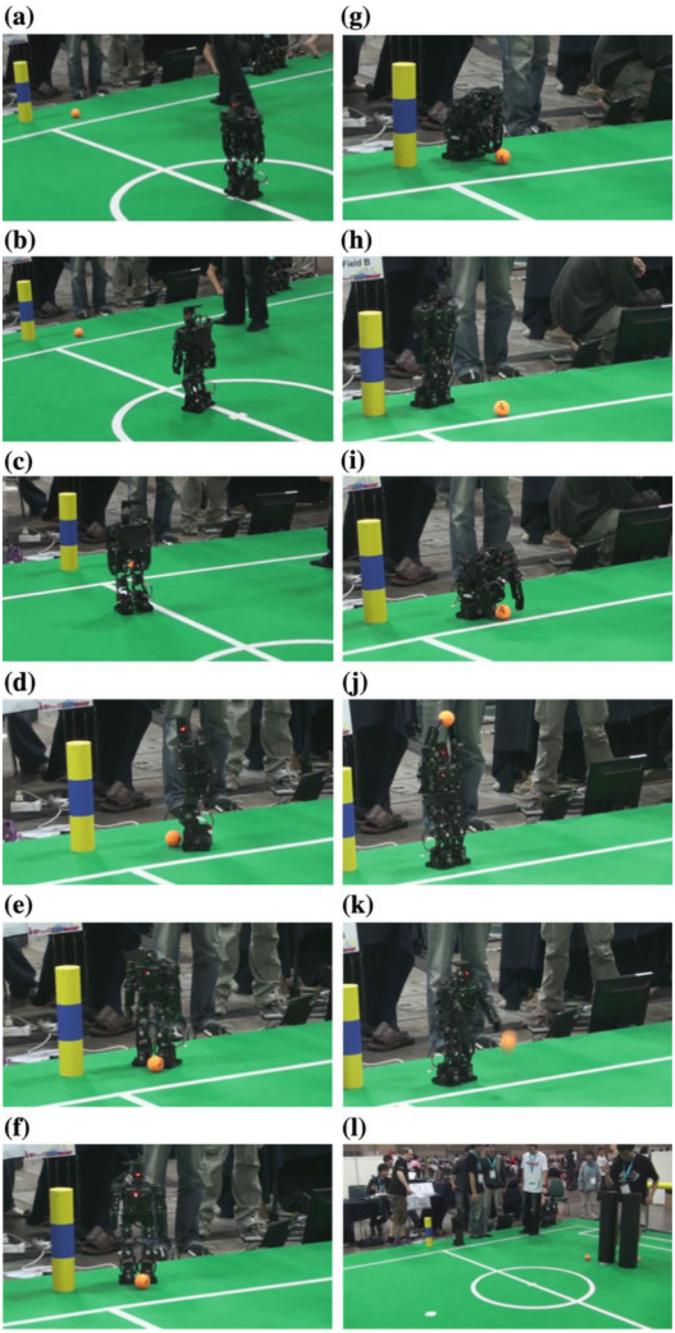


Fig. 12 The trajectories of robot in a throw-in technical challenge in RoboCup 2010

challenge in laboratory is done in about 1.5 min and throws over 215 cm. aiRobots-V starts searching the ball, and then the robot is looking at the ball and checking the pole in Fig. 11a. The robot walks to the ball in Fig. 11b, c then it looks to the goal in Fig. 11d and turns around the ball to face the goal in Fig. 11e. The position of the ball is checked in Fig. 11f and the robot moves to a proper position in Fig. 11g. In Fig. 11h, the robot checks the direction of the opponent goal again. The robot picks up the ball in Fig. 11i, holds it over the head in Fig. 11j and throws it into the field in Fig. 11k. The final position of the ball in the field is shown in Fig. 11l.

4.3 Throw-in Technical Challenge in RoboCup 2010

Figure 12 is the trajectories of the robot in a throw-in technical challenge in RoboCup 2010. In the competition, the best trial of aiRobots-V is done in about 2.5 min and throws over 190 cm. aiRobots-V starts searching the ball and checks the pole in Fig. 12a. The robot walks to the ball in Fig. 12b, c then it looks to the goal in Fig. 12d and turns around the ball to face the goal in Fig. 12e. After checking the position of the ball, the robot checks the direction of the opponent goal in Fig. 12f. The robot picks up the ball in Fig. 12g but it fails and moves the ball a bit forward. The robot looks to the ball again and walks to it in Fig. 12h. The ball is picked up in Fig. 12i and held over the head in Fig. 12j. aiRobots-V throws the ball into the field in Fig. 12k. The final position of the ball in the field is shown in Fig. 12l. This performance of aiRobots-V in the game made the team win the second place in this technical challenge of RoboCup 2010.

5 Conclusions

In this paper, the implementation of RoboCup throw-in technical challenge is proposed. First, the basic concept of real-time object detection based on color information is introduced. Then, the control strategy of the whole process is described. Eventually, the experiment results demonstrate the performance of the proposed method.

Furthermore, the robot threw the ball up to 190 cm away in the best trial of the competition. This task took about 2.5 min to be done, and this result made the robot win the second place of the technical challenge of RoboCup 2010.

Acknowledgments This work supported by Ministry of Science and Technology of Taiwan, R.O.C, Grants MOST 103-2221-E-006-252, MOST 104-2221-E-006-228 -MY2, and the aim for the Top University Project to the National Cheng Kung University (NCKU) are greatly appreciated.

References

1. FIRA RoboWorld Cup. http://www.fira.net/
2. Robocup. http://www.robocup.org/
3. Li, T.-H.S., Su, Y.-T., Lai, S.-W., Hu, J.-J.: Walking motion generation, synthesis, and control for biped robot by using PGRL, LPI and fuzzy logic. IEEE Trans. Syst., Man, Cybern. B **41** (3), 736–748 (2011)
4. Kuo, P.-H., Ho, Y.-F., Lee, K.-F., Tai, L.-H., S. Li, T.-H.: Development of humanoid robot simulator for gait learning by using particle swarm optimization. In: Proceedings of 2013 IEEE International Conference on Systems, Man, and Cybernetics, pp. 2683–2688 (2013)
5. Li, T.-H.S., Kuo, P.-H., Ho, Y.-F., Kao, M.-C., Tai, L.-H.: A biped gait learning algorithm for humanoid robots based on environmental impact assessed artificial bee colony. IEEE Access **3**, 13–26 (2015)
6. Morimoto, J., Endo, G., Nakanishi, J., Cheng, G.: A biologically inspired biped locomotion strategy for humanoid robots: modulation of sinusoidal patterns by a coupled oscillator model. IEEE Trans. Robot. **24**(1), 185–191 (2008)
7. Liu, C., Wang, D., Chen, Q.: "Central pattern generator inspired control for adaptive walking of biped robots. IEEE Trans. Syst. Man Cybern. Syst. **43**(5), 1206–1215 (2013)
8. Chang, C.M.: Design and implementation of vision and strategy system for humanoid robot soccer competition, Master Thesis, Department of E.E., N.C.K.U., Taiwan, July 2009
9. Sen Gupta, G., Bailey, D.: Discrete YUV look-up tables for fast colour segmentation for robotic applications. In: Canadian Conference on Electrical and Computer Engineering, CCECE, 2008, pp. 000963-000968
10. Wang, T.-K.: Design and implementation of double passing strategy for humanoid soccer robot, Master Thesis, Dept. of E.E., N.C.K.U., Taiwan, Aug. 2010
11. Rules of Humanoid League in RoboCup 2010. http://www.robocuphumanoid.org/wp-content/uploads/HumanoidLeagueRules2010.pdf
12. Chong, K.Y.: Design and implementation of fuzzy policy gradient gait learning method for humanoid robot, Master Thesis, Department of E.E., N.C.K.U., Taiwan, July, 2010
13. Su, Y.-T., Chong, K.-Y., Li, T.-H.S.: Design and implementation of fuzzy policy gradient gait learning method for walking pattern generation of humanoid robots. Int. J. Fuzzy Syst. **13**(4), 369–382 (2011)

Study on a Two-Staged Control of a Lower-Limb Exoskeleton Performing Standing-Up Motion from a Chair

Sergey Jatsun, Sergei Savin, Andrey Yatsun and Igor Gaponov

Abstract The paper is concerned with control over a lower-limb exoskeleton device when it performs a sit-to-stand motion from a chair. A mathematical model describing full dynamics of the device is derived, and the strategies to facilitate the desired motion are outlined. A control system based on a modified Jacobian transpose is proposed, and its performance is evaluated in a series of numerical experiments. The simulations proved that the proposed control method can be successfully used to provide stable sit-to-stand motion from a chair.

1 Introduction

A significant amount of research efforts was directed in recent years into development of devices which augment human capabilities. Exoskeletons are among such devices. Two principal applications of exoskeletons include augmentation of human capabilities to boost their performance, and compensation for motoric functions of people who suffer from certain locomotion disabilities or require rehabilitation.

In case of exoskeletons for lower-limb assistance, the devices should support execution of such basic movements as walking, standing up, tilting, and others, regardless of the areas of exoskeleton application. Various researchers have studied

S. Jatsun · S. Savin (✉) · A. Yatsun
Department of Mechanics, Mechatronics and Robotics, Southwest State University,
305040 Kursk, Russia
e-mail: sergey89mtkgtu@mail.ru

S. Jatsun
e-mail: teormeh@inbox.ru

I. Gaponov
School of Mechanical Engineering, Korea University of Technology and Education,
Cheonan 31253, Republic of Korea
e-mail: igor@koreatech.ac.kr

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_10

in details control of bipedal robots during walking [1–6], generation of correct gait, and walking patterns [1–3, 5, 7], gait stabilization [2, 4], and obstacle avoidance [8, 9], while another study was concerned with dynamic weight lifting by robot's body [10]. Several research works are dedicated to study of the use of exoskeletons to assist people during sit-to-stand motion [11–17]. Another work considers standing up from crouching position and proposes an exoskeleton control system to support such motion [18].

At the same time, relatively few studies were dedicated to the problem of exoskeleton control when standing up from a chair. This paper is concerned with the ways to realize such motion by dividing it into two stages. This work also investigates the problem of control system design that ensures static stabilization of the mechanism.

2 The Object of Study

This work considers an exoskeleton to be moving in a sagittal plane. Assuming that both thighs are moving synchronously, we can describe their orientation by a single generalized coordinate. The same assumption holds for both shins and feet. Under these assumptions, the exoskeleton can be modeled as a 4-link mechanism.

Figure 1 presents a schematic diagram of the exoskeleton when 'sitting' on a chair.

In Fig. 1, $\varphi_i(i=\overline{2,4})$ denote generalized joint coordinates, O_i are rotational joints connecting the links, $C_i(i=\overline{1,4})$ denote the respective centers of masses, N_1, N_2, N_4 represent normal reaction forces, and F_{fr1}, F_{fr4} denote friction forces.

We distinguish two stages of motion when standing up from a chair. The first stage lasts while the reaction force N_4 remains positive. At the end of this phase, the force N_4 becomes zero, and the second stage of motion starts. Such distinction is motivated by the fact that, during the second stage it is much more difficult to

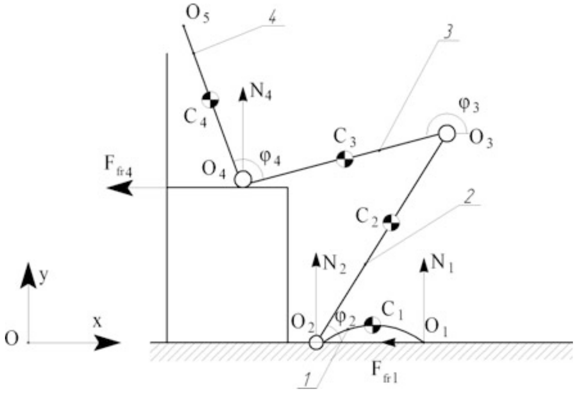


Fig. 1 A principal force diagram of an exoskeleton 'sitting' on a chair: 1–4 denote the links of the device

ensure stabilization of the exoskeleton. One way to provide such stabilization is to use the zero moment point (ZMP) control [19, 20]. However, depending on the configuration of the mechanism at the start of the second motion stage, its stabilization may require substantial accelerations, which may cause discomfort to the wearer. In order to avoid this issue, during the first motion stage the exoskeleton should be brought into a configuration that will simplify its stabilization when standing up. This can be achieved by moving the center of mass of the mechanism and the foot (the first link) so that the following inequality holds:

$$x_{O1} > x_C > x_{O2} \tag{1}$$

where x_C denotes the projection of radius-vector connecting the origin to the center of mass of the mechanism on the x-axis, while the variables x_{O1}, x_{O2} denote the locations of the points O_1 and O_2 on the horizontal axis.

We can derive three strategies of exoskeleton control during the first motion stage that lead to desired result. The first strategy lies in moving the foot horizontally in the direction of the current location of the center of mass (CM). Here, the foot may slide along the base surface or move without any contact with it, thus making a step-like motion. The second strategy requires tilting the body (the fourth joint of the mechanism) so that its center of mass moves towards the foot, thus translating the CM of the whole mechanism in that direction. Finally, the third strategy requires the point O_4 to slide toward the foot, which also shifts to the CM of the whole mechanism in that direction. Figure 2 shows a series of photographs of a human subject standing up from a chair while wearing the exoskeleton. These pictures were taken during the experimental study of the exoskeleton with the help of the "ExoMeasurer" measurement system developed at the SWSU [21].

The first of the above-mentioned strategies cannot be implemented if the geometry of the chair does not allow the foot to move sufficiently close to the current CM location (for instance, if the wearer is sitting in an armchair). The second strategy may also violate condition (1) in some cases, e.g., if the foot is extended far ahead. Therefore, in this work we are concerned with the realization of the third control strategy.

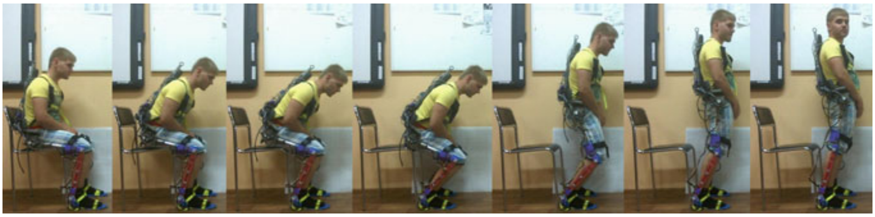


Fig. 2 Configurations of the exoskeleton under combined second and third control strategies

3 Mathematical Model

The main difficulty in the realization of the third control strategy is the need to overcome the friction force F_{fr4} in the first motion stage. We assume that the exoskeleton is equipped with rollers that allow eliminating tangential force acting at point O_4. Under this assumption, motion of the mechanism can be described by two differential equations, which can be written in the following form:

$$\begin{aligned} & I_3\ddot{\varphi}_3 + \mu_3\dot{\varphi}_3 + M_{23} - M_{43} + 0.25\ddot{\varphi}_3 l_3^2 m_3 + \ddot{\varphi}_3 l_3^2 m_4 + 0.5 l_3 m_3 g \cos\varphi_3 \\ & + l_3 m_4 g \cos\varphi_3 - 0.5\dot{\varphi}_2^2 l_2 l_3 m_3 \sin(\varphi_2 - \varphi_3) - \dot{\varphi}_2^2 l_2 l_3 m_4 \sin(\varphi_2 - \varphi_3) \\ & + 0.5\dot{\varphi}_4^2 l_3 l_4 m_4 \sin(\varphi_3 - \varphi_4) + 0.5\ddot{\varphi}_2 l_2 l_3 m_3 \cos(\varphi_2 - \varphi_3) \\ & + \ddot{\varphi}_3 l_2 l_3 m_4 \cos(\varphi_2 - \varphi_3) + 0.5\ddot{\varphi}_4 l_3 l_4 m_4 \cos(\varphi_3 - \varphi_4) - N_4 \cos\varphi_3 l_3 = 0, \end{aligned} \tag{2}$$

$$\begin{aligned} & 0.5 l_4 m_4 g \cos(\varphi_4) - 0.5\dot{\varphi}_2^2 l_2 l_4 m_4 \sin(\varphi_2 - \varphi_4) + 0.5\ddot{\varphi}_3 l_3 l_4 m_4 \cos(\varphi_3 - \varphi_4) \\ & - 0.5\dot{\varphi}_3^2 l_3 l_4 m_4 \sin(\varphi_3 - \varphi_4) + 0.25\ddot{\varphi}_4 l_4^2 m_4 + \ddot{\varphi}_3 l_2 l_3 m_4 \cos(\varphi_2 - \varphi_3) \\ & + 0.5\ddot{\varphi}_2 l_2 l_4 m_4 \cos(\varphi_2 - \varphi_4) + I_4\ddot{\varphi}_4 + \mu_4\dot{\varphi}_4 + M_{34} = 0, \end{aligned} \tag{3}$$

where I_i denotes the moment of inertia of an i-th link, terms l_i represents the lengths of the links, variables μ_i denotes the viscous friction coefficients of the i-th joint, and g corresponds to the gravity constant.

These two differential equations contain three unknown variables, namely, $\varphi_2, \varphi_3, \varphi_4$. Two of these variables (φ_2 and φ_3) can be found from the above-mentioned equations, given the third variable φ_4 is calculated from the constraint equation of the point O_4:

$$y_{O2} + \sin\varphi_2 \cdot l_2 + \sin\varphi_3 \cdot l_3 = h, \tag{4}$$

where h corresponds to the height of the chair, and y_{O2} is a constant which determines the location of the joint O_2 in projection on the vertical axis.

In order to find the reaction force N_4, one can use the following equation:

$$\begin{aligned} N_4 \cos\varphi_2 l_2 = {} & I_2\ddot{\varphi}_2 + \mu_2\dot{\varphi}_2 + 0.5\ddot{\varphi}_4 l_2 l_4 m_4 \cos(\varphi_2 - \varphi_4) + 0.25\ddot{\varphi}_2 l_2^2 m_2 \\ & + \ddot{\varphi}_2 l_2^2 m_3 + \ddot{\varphi}_2 l_2^2 m_4 + 0.5\dot{\phi}_3^2 l_2 l_3 m_3 \sin(\varphi_2 - \varphi_3) + \dot{\varphi}_3^2 l_2 l_3 m_4 \sin(\varphi_2 - \varphi_3) \\ & + 0.5 l_2 m_2 g \cos\varphi_2 + l_2 m_3 g \cos\varphi_2 + M_{12} - M_{32} + l_2 m_4 g \cos\varphi_2 \\ & + 0.5\dot{\varphi}_4^2 l_2 l_4 m_4 \sin(\varphi_2 - \varphi_4) + 1.5\ddot{\varphi}_3 l_2 l_3 m_3 \cos(\varphi_2 - \varphi_3). \end{aligned} \tag{5}$$

Using the Eqs. (2)–(5), we can model the dynamics of the system during the first motion stage for various values of the input torques M_{ij}. The equations of motion of the device during the second stage of motion are presented in [22].

4 Control System

This study employs an automatic control system for control over the exoskeleton, with the regulators proposed in [23]. This approach to generate control torques is known as the Jacobian transpose control. The structure of the control system is presented in Fig. 3.

The control system uses the error of the position of the mechanism's CM. The error e_x is fed into the regulator, and the control input is generated according to

$$\left(\frac{\partial x_{O4}}{\partial q}\right)^T k_1 e_x = \begin{bmatrix} M_{12} - M_{32} \\ M_{23} - M_{43} \\ M_{34} \end{bmatrix}, \tag{6}$$

where $q = [\varphi_2 \quad \varphi_3 \quad \varphi_4]^T$ is the generalized coordinate vector, $x^*_{O4}(t)$ denotes desired position of point O_4 in projection onto the x-axis at time t, $e_x = x^*_{O4}(t) - x_{O4}$, and the term k_1 denotes the regulator's gain.

The desired position can be described as a polynomial function

$$x^*_{O4}(t) = \sum_{p=0}^{3} a_p t^p, \tag{7}$$

where a_p denotes a constant polynomial coefficient.

As a result, the following expressions for the motors' torques can be derived:

$$\begin{aligned} M^J_{12} &= k_1 \cdot \left(\sum_{p=0}^{3} a_p t^p - x_{O2} - \sum_{j=2}^{3} l_j \cos \varphi_j\right)(-l_2 \sin \varphi_2 - l_3 \sin \varphi_3) \\ M^J_{23} &= k_1 \cdot \left(\sum_{p=0}^{3} a_p t^p - x_{O2} - \sum_{j=2}^{3} l_j \cos \varphi_j\right)(-l_3 \sin \varphi_3) \\ M^J_{34} &= 0, \end{aligned} \tag{8}$$

where M^J_{12}, M^J_{23}, M^J_{34} denote the input torques generated by the Jacobian transpose controller.

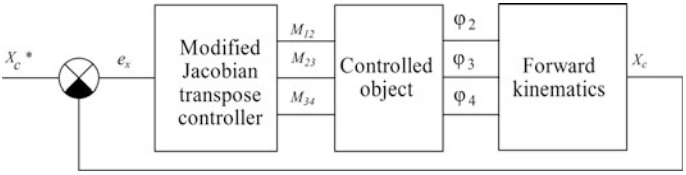


Fig. 3 Control system architecture with a Jacobian transpose controller

One may notice that, with such configuration, the fourth link is not controlled. This allows implementation of a different control system for the fourth link which will realize the second control strategy—tilting the body before standing up. Taking into account this modification, one may rewrite expressions (8) for the torques M_{12}^J, M_{23}^J, M_{34}^J in the following final form:

$$
\begin{aligned}
M_{12}^J &= k_1 \cdot \left(\sum_{p=0}^{3} a_p t^p - x_{O2} - \sum_{j=2}^{3} l_j \cos \varphi_j \right) (-l_2 \sin \varphi_2 - l_3 \sin \varphi_3) \\
M_{23}^J &= k_1 \cdot \left(\sum_{p=0}^{3} a_p t^p - x_{O2} - \sum_{j=2}^{3} l_j \cos \varphi_j \right) (-l_3 \sin \varphi_3) \\
M_{34}^J &= k_2 \cdot \left(\varphi_4^*(t) - \varphi_4 \right),
\end{aligned}
\tag{9}
$$

where k_2 is the regulator's constant, $\varphi_4^*(t) = \sum_{p=0}^{3} b_p t^p$ denotes desired value of angle φ_4 at time t, and b_p represents constant polynomial coefficients.

As a result, the proposed modified control system combines two control strategies of motion of the exoskeleton in the first motion stage. Control over the device during the second motion phase, when the point O_4 loses contact with the chair, can be defined by using the methods proposed in [18]. This controller is based on a multichannel PI regulator and a linear compensation unit. The results obtained in [18] are directly applicable to the case studies in this paper.

5 Numerical Results

Using the derived dynamic model of the device and control system architecture described in the previous section, we performed numerical modeling of the exoskeleton system. Below is the table of parameter values used in the numerical simulation Table 1.

Table 1 Numerical values for parameters used in the numerical simulation

#	Parameter	Value	#	Parameter	Value
1	m_1	1.6 kg	10	$\varphi_4(0)$	$\pi/3$
2	m_2	5.8 kg	11	a_0	−2.27
3	m_3	12.6 kg	12	a_1	1.7
4	m_4	39.6 kg	13	a_2	0
5	l_1	0.28 m	14	a_3	0.058
6	l_2	0.45 m	15	b_0	4.94
7	l_3	0.41 m	16	b_1	−2.46
8	l_4	0.72 m	17	b_2	0
9	$\varphi_2(0)$	$\pi/3$	18	b_3	5.18

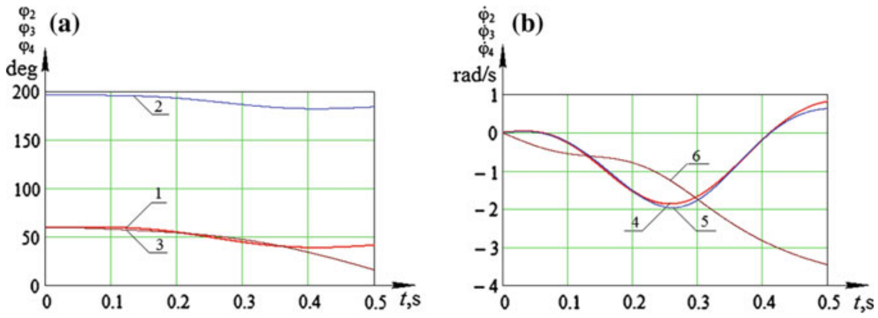


Fig. 4 Numerical simulation results of the motion of the device during the first phase: **a** angles, **b** angular velocities. Captions 1–3 correspond to $\varphi_2 - \varphi_4$, while 4–6 denote $\dot{\varphi}_2 - \dot{\varphi}_4$

Simulation was performed in Mathcad software, numeric approximation for the solution of the differential Eqs. (2)–(3) was obtained with a constant step iterative method. Figure 4 depicts the time plots of the generalized coordinates and velocities of the mechanism in the first motion phase.

It can be noted that trunk and shins move synchronously throughout most of the first motion phase. Similar pattern was observed during the experiment shown in Fig. 2.

To assess the performance of the developed control system, we can compare the error between the desired and simulated positions of the center of mass. Both curves are presented in Fig. 5, and the RMS error was measured to be 9.567 mm.

It can be noticed that the center of mass does not reach the desired position at the end of the motion stage, with positioning error being approximately 7.5 %. In addition, the simulated curve exhibits a local maximum that is not present in the desired curve. However, the CM's projection on the horizontal axis lies within that of the foot throughout the whole simulation, as it was required by the proposed control strategy.

The motion of the exoskeleton can be also modeled in the second motion phase. Figure 6 depicts time plots of the generalized coordinates and velocities.

One may notice that the control law employed during the second phase leads to increased oscillations in the system, which becomes especially evident after

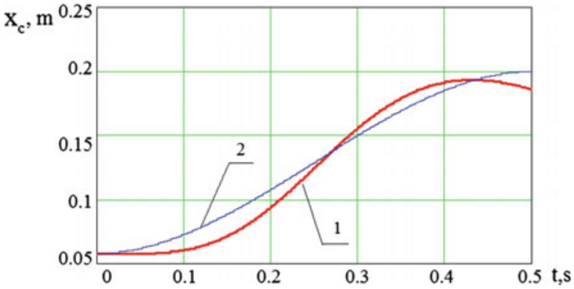


Fig. 5 Motion of the center of mass of the mechanism along the horizontal axis during the first stage: *1*—actual motion; *2*—desired motion

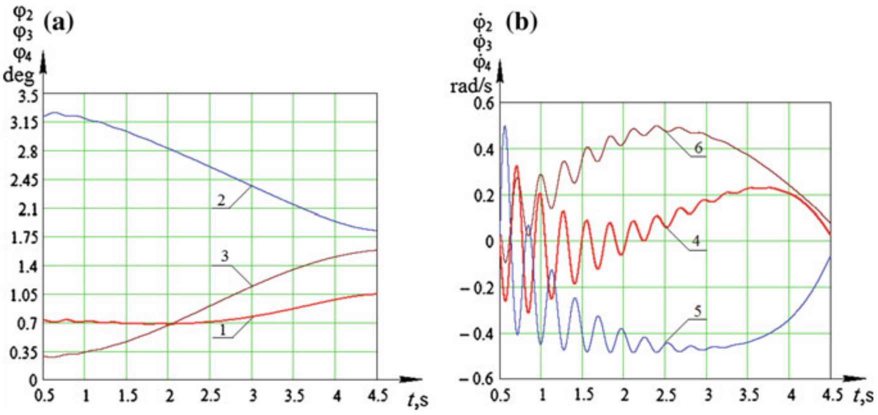


Fig. 6 Numerical simulation results of the exoskeleton's motion during the second phase: **a** generalized angles, **b** angular velocities

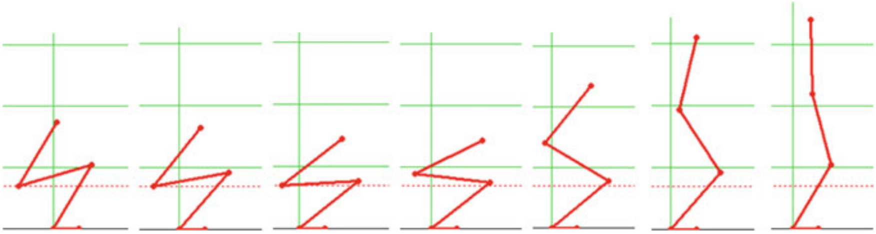


Fig. 7 Exoskeleton configurations during the sit-to-stand motion

examining the velocity plots. In order to mitigate these undesired effects, Jacobian transpose control can be applied at the second motion phase, which is one of the subjects of our future work.

Figure 7 denotes the sequence of simulated exoskeleton configurations during standing up from a chair. The dashed line corresponds to the height of the chair.

6 Conclusion

This work studied the problems of modeling and control of a lower-limb exoskeleton during the sit-to-stand motion. Three possible control approaches capable of solving this problem are described, and a combination of two of these strategies was investigated. The dynamic model of the device was derived, and the numerical simulation results are presented. A modified Jacobian transpose controller was designed, which allowed to implement the proposed combined control strategy. We performed mathematical modeling of the system, and numerical results suggest that the proposed control system can be effectively applied to the lower-limb exoskeleton to realize the sit-to-stand motion from a chair, with and RMS error being approximately 1 cm. We are planning to verify these results experimentally in the nearest future using developed practical exoskeleton system.

Acknowledgments Work is performed with RSF, Project № 14-39-00008 "The establishment of the research laboratory of modern methods and robotic systems to improve the human environment".

References

1. Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., Hirukawa, H.: The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001. Proceedings. 2001, vol. 1, pp. 239–246. IEEE (2001)
2. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H.: Biped walking pattern generation by using preview control of zero-moment point. In: IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA'03. vol. 2, pp. 1620–1626. IEEE (2003)
3. Harada, K., Kajita, S., Kanehiro, F., Fujiwara, K., Kaneko, K., Yokoi, K., Hirukawa, H.: Real-time planning of humanoid robot's gait for force-controlled manipulation. IEEE/ASME Trans. Mechatron. **12**(1), 53–62 (2007)
4. Kajita, S., Morisawa, M., Miura, K., Nakaoka, S., Harada, K., Kaneko, K., Yokoi, K.: Biped walking stabilization based on linear inverted pendulum tracking. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4489–4496. IEEE (2010)
5. Huang, Q., Yokoi, K., Kajita, S., Kaneko, K., Arai, H., Koyachi, N., Tanie, K.: Planning walking patterns for a biped robot. IEEE Trans. Robot. Autom. **17**(3), 280–289 (2001)
6. 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)
7. Brock, O., Khatib, O.: Elastic strips: a framework for motion generation in human environments. Int. J. Robot. Res. **21**(12), 1031–1052 (2002)
8. Yoshida, E., Esteves, C., Sakaguchi, T., Laumond, J.P., Yokoi, K.: Smooth collision avoidance: Practical issues in dynamic humanoid motion. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 827–832. IEEE ((2006)
9. Verrelst, B., Stasse, O., Yokoi, K., Vanderborght, B.: Dynamically stepping over obstacles by the humanoid robot HRP-2. In: 2006 6th IEEE-RAS International Conference on Humanoid Robots, pp. 117–123. IEEE (2006)
10. Arisumi, H., Miossec, S., Chardonnet, J.R., Yokoi, K.: Dynamic lifting by whole body motion of humanoid robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2008, pp. 668–675. IEEE (2008)
11. Tsukahara, A., Hasegawa, Y., Sankai, Y.: Standing-up motion support for paraplegic patient with Robot Suit HAL. In: IEEE International Conference on Rehabilitation Robotics. ICORR 2009, pp. 211–217. IEEE (2009)
12. Tsukahara, A., Kawanishi, R., Hasegawa, Y., Sankai, Y.: Sit-to-stand and stand-to-sit transfer support for complete paraplegic patients with robot suit HAL. Adv. Robot. **24**(11), 1615–1638 (2010)

13. Jun, H.G., Chang, Y.Y., Dan, B.J., Jo, B.R., Min, B.H., Yang, H., Kim, J.: Walking and sit-to-stand support system for elderly and disabled. In: 2011 IEEE International Conference on Rehabilitation Robotics (ICORR), pp. 1–5. IEEE (2011)
14. Taslim Reza, S.M., Ahmad, N., Choudhury, I.A., Ghazilla, R.A.R.: A fuzzy controller for lower limb exoskeletons during sit-to-stand and stand-to-sit movement using wearable sensors. Sensors **14**(3), 4342–4363 (2014)
15. Salah, O., Ramadan, A.A., Sessa, S., Ismail, A.A., Fujie, M., Takanishi, A.: Anfis-based sensor fusion system of sit-to-stand for elderly people assistive device protocols. Int. J. Autom. Comput. **10**(5), 405–413 (2013)
16. Mughal, A.M., Iqbal, K.: 3D bipedal model for biomechanical sit-to-stand movement with coupled torque optimization and experimental analysis. In: 2010 IEEE International Conference on Systems Man and Cybernetics (SMC), pp. 568–573. IEEE (2010)
17. López, A.M., Vaillant, J., Keith, F., Fraisse, P., Kheddar, A.: Compliant control of a humanoid robot helping a person stand up from a seated position. In: 2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 817–822. IEEE (201)
18. Jatsun, S., Savin, S., Yatsun, A., Malchikov, A.: Study of controlled motion of exoskeleton moving from sitting to standing position. In: 24th International Conference on Robotics in Alpe-Adria-Danube Region, vol. 201
19. Mitobe, Kazuhisa, Capi, Genci, Nasu, Yasuo: Control of walking robots based on manipulation of the zero moment point. Robotica **18**(06), 651–657 (2000)
20. Chi, Z., et al.: Biped walking with variable ZMP, frictional constraint, and inverted pendulum model. In: IEEE International Conference on Robotics and Biomimetics. ROBIO 2004. IEEE, 2004
21. Jatsun S., Savin S., Bezmen P.: Modelling of exoskeleton movement in verticalization process, New Developments in Pure and Applied Mathematics p. 83–87 (2015)
22. Jatsun, S.F., Yu, L., Vorochaeva, A., Yatsun, S., Savin, S.I.: The modeling of the standing-up process of the anthropomorphic mechanism. In: Proceedings of International Conference on Climbing and Walking Robots CLAWAR 2015 [in publishing], 2015
23. Sunada, C. et al.: A coordinated jacobian transpose control for mobile multi-limbed robotic systems. In: 1994 IEEE International Conference on Robotics and Automation. Proceedings. IEEE (1994)

Adaptive Control for Directional Drilling Systems with Delay and Parameter Uncertainty

Arief B. Koesdwiady, Sami Elferik and Fakhri Karray

Abstract In this paper, an adaptive tracking control of a directional drilling system subject to state delays and parameter uncertainties is investigated. The Explicit Force, Finitely Sharp, Zero Mass (EFFSZM) system is used to represent a directional drilling system. Invariance and Immersion (I&I) adaptive control approach is used to ensure the EFFSZM system to follow a set of predefined path autonomously. The performance of the proposed approach is compared to the one $\mathcal{L}_1$ adaptive control under the identical condition. To observe and validate the performance of the proposed approach, several simulation scenarios are conducted. The results show that both the controllers are able to stabilize the system in the presence of parameter uncertainties while maintaining the trajectory tracking error minimum.

Keywords Adaptive control · Directional drilling · Uncertain systems

1 Introduction

In oil and gas industry, directional drilling systems play an important role in enhancing oil and gas reservoir production. The directional drilling systems are usually implemented to get around a problem where vertical drilling systems are very difficult or impossible to implement, such as to extract the oil or gas that are located under an urban or protected area. Directional drilling systems allow the drill bit to

A.B. Koesdwiady (✉) · F. Karray
Centre for Pattern Analysis and Machine Intelligence,
University of Waterloo, Waterloo, ON, Canada
e-mail: abkoesdw@uwaterloo.ca

F. Karray
e-mail: karray@uwaterloo.ca

S. Elferik
Systems Engineering Department,
King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia
e-mail: selferik@kfupm.edu.sa

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_11

be directed towards a desired complex location in underground and to have a curved trajectory. Therefore, the directional systems are also suitable for extracting unconventional gas, e.g. shale and tight gas, that is trapped in the source rock instead of migrated into a reservoir. In addition, the directional drilling systems have the potential to extract 2–25 times more oil and gas than the conventional vertical drilling systems have [1, 4, 7].

One of the major challenges in directional drilling systems is the tracking control of the drill bit to follow a predefined trajectory designed by the drilling engineer. In most cases, in order to design a proper control for the directional drilling system, a model that represents the dynamics of the directional drilling system needs to be formulated. In [5], a transfer function relating the centre line of the drilling hole to the actuator stimuli is derived. A quasi-polynomial is used to approximate the space curve of the centre line of the hole, with distance m instead of t as the independent variable. However, it is almost impossible to get 100 % accuracy in the modelling, hence a robust controller that can handle wide uncertainties is needed.

In [8, 9], a recent adaptive control technique, i.e. $\mathcal{L}_1$ adaptive control, is used for tracking a specified trajectory of a directional drilling system, where some parameters in the model are considered to be uncertain but bounded. An output-feedback control strategy is developed in [10]. In this work, the curved wellbore generation is formulated as a tracking problem that is stabilized using an output-feedback control consisting of a model-based observer with integral action. In [3], a model predictive control (MPC) algorithm using a linear state-space plant model augmented with pure delays is investigated. The MPC is used to control the attitude of a drilling tool while mitigating for the attitude limit cycles about the target attitude caused by measurement feedback delays. Recently, control and optimization of a directional drilling system with quad motor drilling heads is developed [6]. This work proposes a real-time control and optimization of the quadbit drilling systems that combines the conventional drilling parameters as well as the directional steering control.

In this paper, a relatively recent technique on adaptive control, namely Invariance and Immersion (I&I) adaptive control, is applied to the Explicit Force, Finitely Sharp, Zero Mass (EFFSZM) directional drilling model [5]. This technique was first introduced in [2]. The idea of this technique is to shape the manifold in which the adaptive system will be immersed. By assuming the existence of the controller, the globally asymptotic stable system is used as the target dynamic. Due to the characteristics of the EFFSZM system, the I&I adaptive control is specifically designed to a class of system with unknown parameters and uncertainty in the presence of internal delay. The goal of the controller is to autonomously control the EFFSZM to follow a set of predefined trajectory. Finally, the proposed controller will be validated by simulations, and its performance will be compared to $\mathcal{L}_1$ adaptive control performance.

2 Directional Drilling System

In [5], a directional drilling transfer function relating the centre line of the drilling hole to the actuator stimuli is derived. The distance drilled is used as the independent variable instead of time, which means the differential equation that represents the directional drilling system is presented in distance drilled (m). There are several types of model derived in [5], one of them is EFFSZM model. In EFFSZM, from the known geometry and action of actuators, the force at the drill bit are explicitly calculated. The pipe work is assumed to be infinitely stiff with zero mass and the drill bit is assumed to be finitely sharp. The EFFSZM drilling system is depicted in Fig. 1.

Consider EFFSZM drilling system differential equation given by the following:

$$
\begin{aligned}
\frac{dH(m)}{dm} &= \left(\frac{1+C_f}{b} - \frac{Cf}{d}\right) H(m) + \frac{1+C_f}{b}\left(V(m) - H(m-b)\right) \\
&\quad + \frac{C_f}{d} H(m-d) + \frac{b-a}{b\ WOB\ K_{anis}} F_{pad}(m) \\
\Psi(m) &= \frac{dH(m)}{dm}
\end{aligned}
\tag{1}
$$

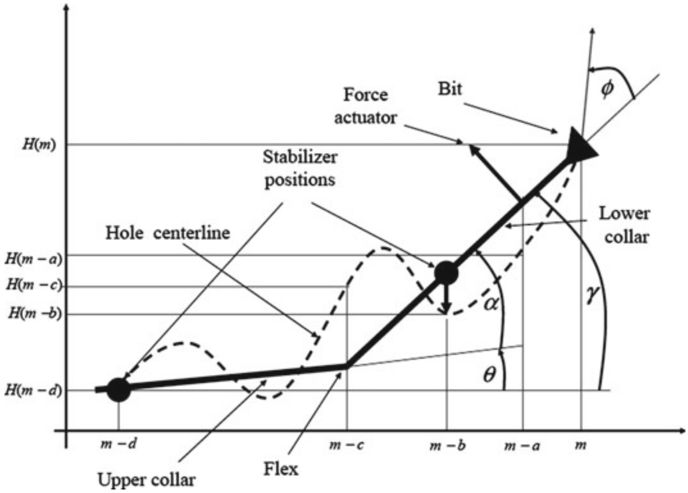


Fig. 1 Flex-hinge directional drilling system [5]

where

$$C_f = \left(\frac{c-b}{K_{anis} - \frac{d-b}{d-c}} \frac{K_{flex}}{WOB} \right) \frac{d}{b(d-c)}, \qquad m \triangleq \text{the distance drilled,}$$

$H(m) \triangleq$ lateral displacement of the borehole

$a \triangleq$ distance between the force actuator and the bit

$b \triangleq$ distance between the lower stabilizer and the bit

$c \triangleq$ the relative position of the flex joint to the bit

$d \triangleq$ the position of the upper stabilizer to the bit

$K_{flex} \triangleq$ angular spring rate of the flex joint

$K_{anis} \triangleq$ ration of rates of penetration along and across the bit

$V(m) \triangleq$ the actuator displacement at the lower stabilizer

$WOB \triangleq$ the applied drilling load, $F_{pad}(m) \triangleq$ the force actuator output

$\Psi(m) \triangleq$ the angle of borehole propagation w.r.t the m—axis

In the directional drilling dynamics, some parameters such as a, b, c, d, and K_{flex} are known, while WOB and K_{anis} are unknown but bounded. The values for these unknown parameters vary in the following range:

$$1 \leq K_{anis} \leq 10, \qquad 5 \times 10^4 \leq WOB \leq 8.9 \times 10^4 \tag{2}$$

3 Adaptive Control Design

In this section, the design of two recent adaptive control techniques is proposed. The first is the immersion and invariance (I&I) adaptive control and the second is $\mathcal{L}_1$ adaptive control, which is used for comparison. The performances of both the techniques in the presence of unknown and uncertain parameters will be compared using simulations. The following subsections provide a comprehensive derivation of both the controllers and the stability analysis.

3.1 I&I Adaptive Control for Linearly Parameterized Plant

The I&I adaptive control technique is first introduced in [2]. For systems than can be constructed in a linear parameterized form, the dynamics can be written as

$$\dot{x} = f_0(x) + f_1(x)\theta + g(x)u \tag{3}$$

where $x \in \mathcal{R}^n$ is the state, $u \in \mathcal{R}^m$ is the input, and $\theta \in \mathcal{R}^q$ is the unknown or uncertain parameters. The adaptive state feedback controller law of the system is given in the following form:

$$\begin{aligned} \dot{\hat{\theta}} &= w(x, \hat{\theta}) \\ u &= v(x, \hat{\theta}) \end{aligned} \tag{4}$$

such that all the trajectories of the closed-loop system is bounded and satisfies $\lim_{t\to\infty} x(t) = x^*$. In the I&I adaptive control, the design of the control law is decoupled from that of the adaptation law, which leads to more flexible control scheme. If the following assumption holds:

Assumption 1 $\exists u = v(x, \theta)$, such that the system

$$\dot{x} = f^*(x) = f(x) + g(x)v(x, \theta) \tag{5}$$

has a global asymptotically stable equilibrium at $x = x^*$.

Then the following theorem can be used in the adaptive control design:

Theorem 1 [2] *There exists a mapping $\beta : \mathcal{R}^n \to \mathcal{R}^q$, such that all the trajectories of the following system*

$$\begin{aligned} \dot{z} &= -\left(\frac{\partial \beta}{\partial x} f_1(x)\right) z \\ \dot{x} &= f^*(x) + g(x)(v(x, \theta + z) - v(x, \theta)) \end{aligned} \tag{6}$$

are bounded and satisfy

$$\lim_{t\to\infty} (g(x)(v(x, \theta + z) - v(x, \theta)) = 0 \tag{7}$$

then the system 3 is adaptively I&I stabilizable.

Proof see [2]. □

The EFFSZM system in (1) can be presented as a linearly parameterized plant in the following form:

$$\frac{dH(m)}{dm} = f_0 + f_1(H)\theta_1 + f_2(H)\theta_2 + f_3(H)\theta_3 + G\theta_4 u(m) \tag{8}$$

where $\theta_i,\ i = 1, 2, 3, 4$ are the unknown parameters, and

$$\begin{aligned} & f_0(H) \triangleq \frac{1}{b} H(m),\ f_1(H) \triangleq H(m), \\ & f_2(H) \triangleq H(m - b),\ f_3(H) \triangleq H(m - d), \end{aligned}$$

$$
\begin{aligned}
G &\triangleq \frac{b-a}{b\ WOB\ K_{anis}} \text{ for nominal } K_{anis} \text{ and } WOB \\
u(m) &\triangleq F_{pad}(m)
\end{aligned}
\tag{9}
$$

To satisfy Assumption 1, the following controller is derived:

$$
\begin{aligned}
u(m) \triangleq v(H,\theta) = -(G\theta_4)^{-1}(&f_0(H) + f_1(H)\theta_1 + f_2(H)\theta_2 + f_3(H)\theta_3 + KH(m) \\
&- K_r r)
\end{aligned}
\tag{10}
$$

where $K_r > 0$ is the feedforward gain, r is the bounded reference and for feedback gain $K > 0$, the following globally exponentially stable closed-loop system is obtained

$$
\frac{dH(m)}{dm} = -KH(m) + K_r r \tag{11}
$$

In order to apply I&I adaptive control on the EFFSZM system, the following implicit manifold is considered:

$$
z_i = \hat{\theta}_i - \theta_i + \beta_i(H) \tag{12}
$$

Based on Eq. 12, the dynamics of the given manifold is derived as follows:

$$
\begin{aligned}
\frac{dz_i}{dm} &= w_i + \frac{d\beta_i(H)}{dH}\frac{dH}{dm} \\
&= w_i + \frac{d\beta_i(H)}{dH}(f_0(H) + f_1(H)\theta_1 + f_2(H)\theta_2 + f_3(H)\theta_3 + G\theta_4 u(m))
\end{aligned}
\tag{13}
$$

If the updating laws are selected to be

$$
\begin{aligned}
w_1 = &-\frac{d\beta_1(H)}{dH}(f_0(H) + f_1(H)(\hat{\theta}_1 + \beta_2(H)) + f_2(H)(\hat{\theta}_2 + \beta_2(H) - z_2) \\
&+ f_3(H)(\hat{\theta}_3 + \beta_3(H) - z_3) + G(\hat{\theta}_4 + \beta_4 - z_4)u(m)) \\
w_2 = &-\frac{d\beta_2(H)}{dH}(f_0(H) + f_1(H)(\hat{\theta}_1 + \beta_1(H) - z_1) + f_2(H)(\hat{\theta}_2 + \beta_2(H)) \\
&+ f_3(H)(\hat{\theta}_3 + \beta_3(H) - z_3) + G(\hat{\theta}_4 + \beta_4 - z_4)u(m)) \\
w_3 = &-\frac{d\beta_3(H)}{dH}(f_0(H) + f_1(H)(\hat{\theta}_1 + \beta_1(H) - z_1) + f_2(H)(\hat{\theta}_2 + \beta_2(H) - z_2) \\
&+ f_3(H)(\hat{\theta}_3 + \beta_3(H)) + G(\hat{\theta}_4 + \beta_4 - z_4)u(m)) \\
w_4 = &-\frac{d\beta_4(H)}{dH}(f_0(H) + f_1(H)(\hat{\theta}_1 + \beta_1(H) - z_1) + f_2(H)(\hat{\theta}_2 + \beta_2(H) - z_2) \\
&+ f_3(H)(\hat{\theta}_3 + \beta_3(H) - z_3) + G(\hat{\theta}_4 + \beta_4)u(m))
\end{aligned}
\tag{14}
$$

the following error dynamic is obtained:

$$\begin{aligned}\frac{dz_j}{dm} &= -\left(\frac{d\beta_j(H)}{dH}f_j(H)\right)z_j \ \ \text{(for j = 1, 2, 3)}\\ \frac{dz_4}{dm} &= -\left(\frac{d\beta_4(H)}{dH}Gu(m)\right)z_4\end{aligned} \tag{15}$$

In the condition of $\frac{d\beta_j(H)}{dH} = \gamma_j f_j(H)$, and $\frac{d\beta_4(H)}{dH} = \gamma_4 Gu(m)$ where $\gamma_i > 0$, the following stabilization is achieved:

$$\lim_{m\to\infty} z_i(m) = 0 \qquad \Rightarrow \qquad \theta_i = \hat{\theta}_i + \beta_i(H) \tag{16}$$

Finally, the I&I adaptive controller is given as the following:

$$\begin{aligned}v(H, \hat{\theta}_i + \beta_i) =& -(G(\hat{\theta}_4 + \beta_4))^{-1}(f_0(H) + f_1(H)(\hat{\theta}_1 + \beta_1(H)) + f_2(H)(\hat{\theta}_2 + \beta_2(H))\\ &+ f_3(H)(\hat{\theta}_3 + \beta_3(H)) + KH(m) - K_r r)\end{aligned} \tag{17}$$

The selection of $\beta_4(H)$ has to be done considering the possibility of the singularity in the control law. If the following is selected

$$\beta_4(H) = c + Gu(m)H(m) \tag{18}$$

for $c \in \mathcal{R}$ and $\beta_4 \neq -\hat{\theta}_4$, then the singularity problem does not exist.

3.2 I&I Adaptive Control Stability Analysis

To investigate the condition in which the controller are able to guarantee its stability, a stability analysis is necessary. First, consider the following Lyapunov function candidate

$$V(z) = z^\top Pz \tag{19}$$

where $P^\top = P > 0$. The derivative of the Lyapunov function along the trajectories of the (15) is given as

$$\dot{V}(z) = z^\top(\Phi^\top P + P\Phi)z \tag{20}$$

where

$$\boldsymbol{\Phi} = -\begin{bmatrix} \gamma_1 f_1^2 & 0 & 0 & 0 \\ 0 & \gamma_2 f_2^2 & 0 & 0 \\ 0 & 0 & \gamma_3 f_3^2 & 0 \\ 0 & 0 & 0 & \gamma_4 (Gu)^2 \end{bmatrix} \tag{21}$$

Since $\boldsymbol{\Phi}$ is Hurwitz, the following Lyapunov equation is satisfied

$$\boldsymbol{\Phi}^\top P + P\boldsymbol{\Phi} = -Q \tag{22}$$

where $Q^\top = Q > 0$; hence, the following is true

$$\dot{V}(z) = -z^\top Qz \le 0 \tag{23}$$

Based on the Lyapunov condition, the system (15) has a globally stable equilibrium at zero, and the asymptotic stability is followed by the LaSalle's Invariance Principle. Since $V(z)$ is a positive definite function and $\dot{V}(z) \le 0$, it implies $V(z) \in \mathcal{L}_\infty$ and $z \in \mathcal{L}_\infty$. Therefore, the off-manifold coordinates are bounded. By integrating $\dot{V}(z)$ it follows that

$$V(\infty) - V(0) = -\int_0^\infty z^\top Qzdt \implies z \in \mathcal{L}_2 \cap \mathcal{L}_\infty \tag{24}$$

Note that the manifold coordinate is given by $z = \hat{\theta} - \theta + \beta(x)$, which implies $\hat{\theta}, \beta \in \mathcal{L}_2$. In conclusion, all the signals in the closed-loop system are bounded.

3.3 $\mathcal{L}_1$ Adaptive Control

This section provides a brief derivation of $\mathcal{L}_1$ adaptive control for EFFSZM, which is presented in [9]. The performance of $\mathcal{L}_1$ will be compared to that of I&I in the simulations. First, consider the EFFSZM system dynamics presented in (1), which can be rewritten in the form of

$$\begin{aligned} \dot{x}(m) &= A_m x(m) + b_0(\omega u(m) + \theta_0^\top(m)x(m) + \theta_1^\top(m)x(m-\tau_1) \\ &\qquad + \theta_2^\top(m)x(m-\tau_2) + \sigma(m)) \\ x(m) &= 0 \;\; \forall\, m \in [-\tau_2, 0] \\ y(m) &= c_0^\top x(m) \end{aligned} \tag{25}$$

In these dynamics, $A_m \in \mathbb{R}$ is Hurwitz; $\theta_0, \theta_1, \theta_2 \in \mathbb{R}$ are unknown; ω is unknown with known sign; b_0, c_0 are known; the delay $\tau_1, \tau_2 \in \mathbb{R}^+,\ (\tau_1 < \tau_2)$ are known; $\sigma(m)$ is the input disturbance; and the following assumption holds:

Assumption 2 ([9])

1. The parameters $\theta_0, \theta_1, \theta_2$ belong to given compact convex sets $\varTheta_0, \varTheta_1, \varTheta_2$, respectively, and continuously differentiable with uniformly bounded derivatives.

$$\begin{aligned} &\theta_0(m) \in \varTheta_0, \qquad \theta_1(m) \in \varTheta_1, \quad \theta_2(m) \in \varTheta_2, \ \ \forall\, m \geq 0 \\ &\theta_{\max 0} \triangleq \max_{\theta \in \varTheta_0} \|\theta\|_1, \quad \theta_{\max 1} \triangleq \max_{\theta \in \varTheta_1} \|\theta\|_1, \quad \theta_{\max 2} \triangleq \max_{\theta \in \varTheta_2} \|\theta\|_1, \\ &\|\dot{\theta}_0(m)\| \leq d_{\theta_0}, \ \ \|\dot{\theta}_1(m)\| \leq d_{\theta_1}, \quad \|\dot{\theta}_2(m)\| \leq d_{\theta_2}, \ \ \|\dot{\sigma}(m)\| \leq d_\sigma, \end{aligned} \tag{26}$$

The disturbance is upper bounded by $|\sigma(m)| \leq \varDelta, \ \ \forall\, m \geq 0$, where $\varDelta > 0$ is a known conservative bound. The unknown parameter ω is lower and upper bounded by

$$\omega_l \leq \omega \leq \omega_u, \ \ 0 < \omega_l < \omega_u \tag{27}$$

Using Eq. 25, the predictor for $\mathcal{L}_1$ adaptive control is selected as follows:

$$\begin{aligned} \dot{\hat{x}}(m) &= A_m \hat{x}(m) + b_0(\hat{\omega} u(m) + \hat{\theta}_0^\top(m) x(m) + \hat{\theta}_1^\top(m) x(m - \tau_1) \\ &\qquad + \hat{\theta}_2^\top(m) x(m - \tau_2) + \hat{\sigma}(m)) \\ \hat{x}(m) &= 0 \ \ \forall\, m \in [-\tau_2, 0] \\ \hat{y}(m) &= c_0^\top \hat{x}(m) \end{aligned} \tag{28}$$

where $\hat{x}(m) \in \mathbb{R}$ is the predicted state; $\hat{y}(m) \in \mathbb{R}$ is the predicted output; and $\hat{\theta}, \hat{\omega}, \hat{\sigma}$ are the estimated parameters. The projection-type adaptive laws for these estimated parameters are given as follows:

$$\begin{aligned} \dot{\hat{\theta}}_0(m) &= \varGamma \mathrm{Pr}(\hat{\theta}_0(m), -\tilde{x}^\top(m) P b_0 x(m)), \ \ \hat{\theta}_0(0) = \hat{\theta}_{00} \\ \dot{\hat{\theta}}_1(m) &= \varGamma \mathrm{Pr}(\hat{\theta}_1(m), -\tilde{x}^\top(m) P b_0 x(m - \tau_1)), \ \ \hat{\theta}_1(0) = \hat{\theta}_{10} \\ \dot{\hat{\theta}}_2(m) &= \varGamma \mathrm{Pr}(\hat{\theta}_2(m), -\tilde{x}^\top(m) P b_0 x(m - \tau_2)), \ \ \hat{\theta}_2(0) = \hat{\theta}_{20} \\ \dot{\hat{\sigma}}(m) &= \varGamma \mathrm{Pr}(\hat{\sigma}(m), -\tilde{x}^\top(m) P b_0), \ \ \hat{\sigma}_0(0) = \hat{\sigma}_0 \\ \dot{\hat{\omega}}(m) &= \varGamma \mathrm{Pr}(\hat{\omega}(m), -\tilde{x}^\top(m) P b_0 u(m)), \ \ \hat{\omega}_0(0) = \hat{\omega}_0 \end{aligned} \tag{29}$$

where $\varGamma > 0$ is the adaptation law rate, and $\tilde{x}(m) = \hat{x}(m) - x(m)$. For $Q = Q^\top > 0$, $P = P^\top > 0$ satisfies the Lyapunov equation

$$A_m^\top P + P A_m = -Q \tag{30}$$

Subsequently, the control signal is defined as

$$u(s) = -kD(s)(\hat{\eta} - k_g r(s)) \tag{31}$$

where $\hat{\eta} \triangleq \hat{\omega}u(m) + \hat{\theta}_0^{\top}(m)x(m) + \hat{\theta}_1^{\top}(m)x(m-\tau_1) + \hat{\theta}_2^{\top}(m)x(m-\tau_2) + \hat{\sigma}(m)$ and $r(m)$ are the inverse Laplace transform of $\hat{\eta}(s)$ and $r(s)$, respectively. The feedforward gain is selected as $k_g \triangleq -\frac{1}{c_0^{\top} Am^{-1} b_0}$, and the feedback gain satisfies $k > 0$. The strictly proper transfer function $D(s)$ is selected, which leads to a strictly proper stable $C(s)$ as the following:

$$C(s) \triangleq \frac{\omega k D(s)}{1 + \omega k D(s)} \tag{32}$$

with DC gain $C(0) = 1$. The selection of $D(s) = \frac{1}{s}$ yields first order strictly proper transfer function

$$C(s) = \frac{\omega k}{s + \omega k} \tag{33}$$

In addition, the following norm condition has to be satisfied for $\mathcal{L}_1$ adaptive control

$$\|G(s)\|_{\mathcal{L}_1}(\theta_{\max 0} + \theta_{\max 1} + \theta_{\max 2}) < 1 \tag{34}$$

where

$$H(s) \triangleq (s\mathbb{I} - A_m)^{-1} b_0, \quad G(s) \triangleq H(s)(C(s) - 1) \tag{35}$$

The detailed analysis of this $\mathcal{L}_1$ adaptive control can be seen in [9].

4 Simulation

To investigate the performance of the proposed adaptive controllers under parameter uncertainty, two scenarios of simulations are performed. In each scenario, the adaptive controllers are tested with incorrect constant parameters and random time-varying uncertain parameters. In the case of uncertain parameters, both of the controllers are tested with smooth and non-smooth reference so that the ability of the controllers in handling non-continuously differentiable reference can be analyzed. The uncertain parameters K_{anis} and WOB are bounded and belong to the following ranges:

$$1 \leq K_{anis} \leq 10, \qquad 5 \times 10^4 \leq WOB \leq 8.9 \times 10^4 \tag{36}$$

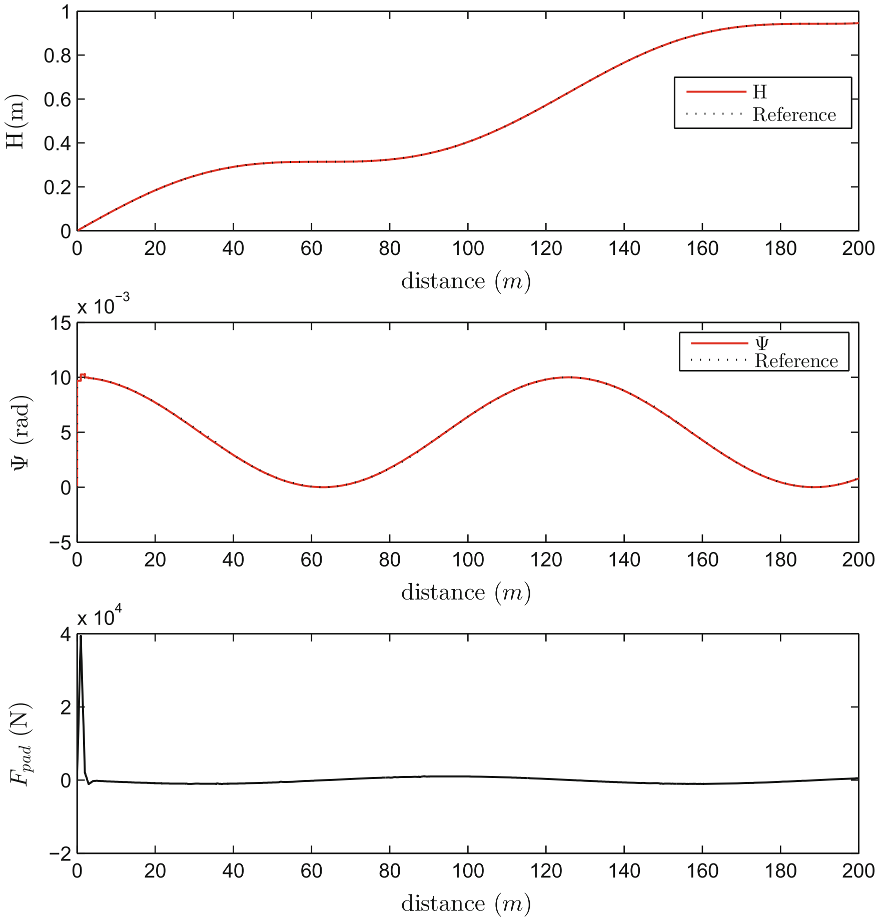


Fig. 2 I&I adaptive control ($K_{anis} = 1$ and $WOB = 5 \times 10^4$)

The nominal values of the parameters are $K_{anis} = 10$ and $WOB = 8.9 \times 10^4$, and

$$\begin{aligned} &a = 0.305\,\text{m}, \quad b = 0.953\,\text{m}, \quad c = 1.407\,\text{m}, \\ &d = 2\,\text{m}, \quad K_{flex} = 8.577 \times 10^5\,\text{N.m/rad} \end{aligned} \tag{37}$$

are known parameters with exact values. The control input $F_{pad}(m)$ is constrained as follows:

$$|F_{pad}(m)| \le 10^7\,\text{N} \tag{38}$$

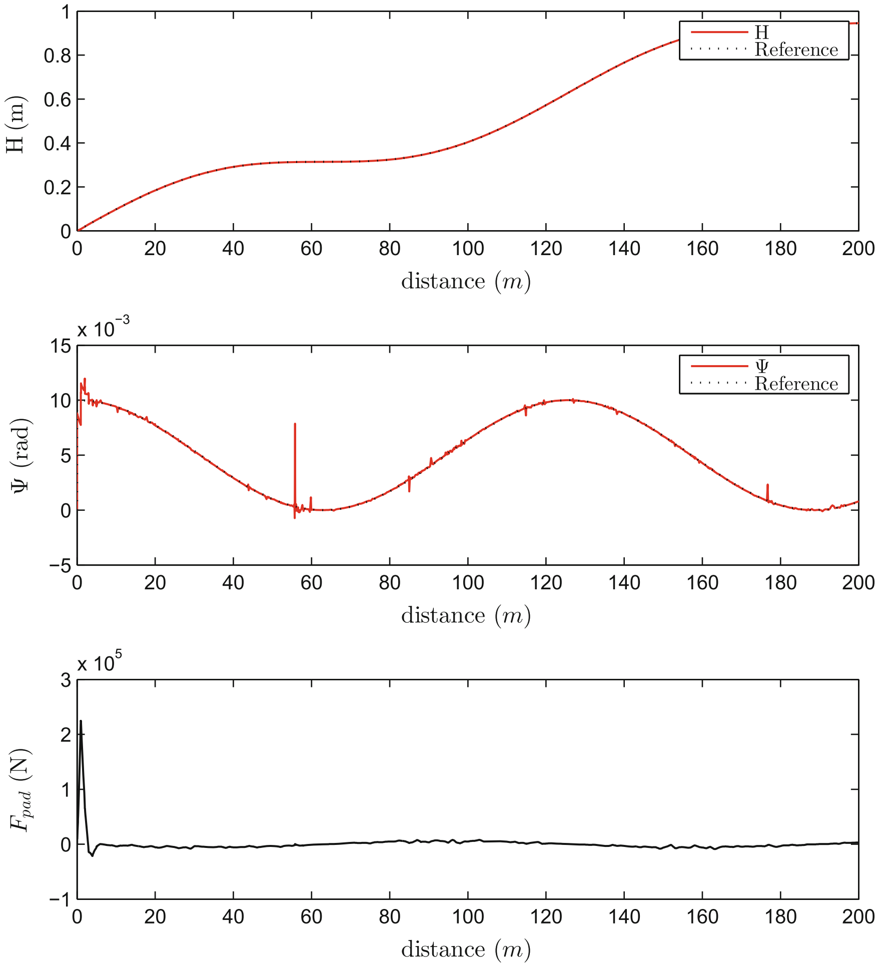


Fig. 3 I&I adaptive control (K_{anis} and *WOB* are changing)

The first case of simulation for I&I adaptive control is conducted with reference $H_{ref}(m) = 0.1\,\sin(0.05m) + (m/200)$, and the constant uncertain parameters are simulated as follows:

$$K_{anis} = 1, \quad WOB = 5 \times 10^4 \tag{39}$$

which are different from the nominal values. As depicted in Fig. 2, the proposed I&I adaptive control is able to stabilize the system and follow the predefined reference as expected. The control signal produced by the controller is also still within the constrained range and is smooth.

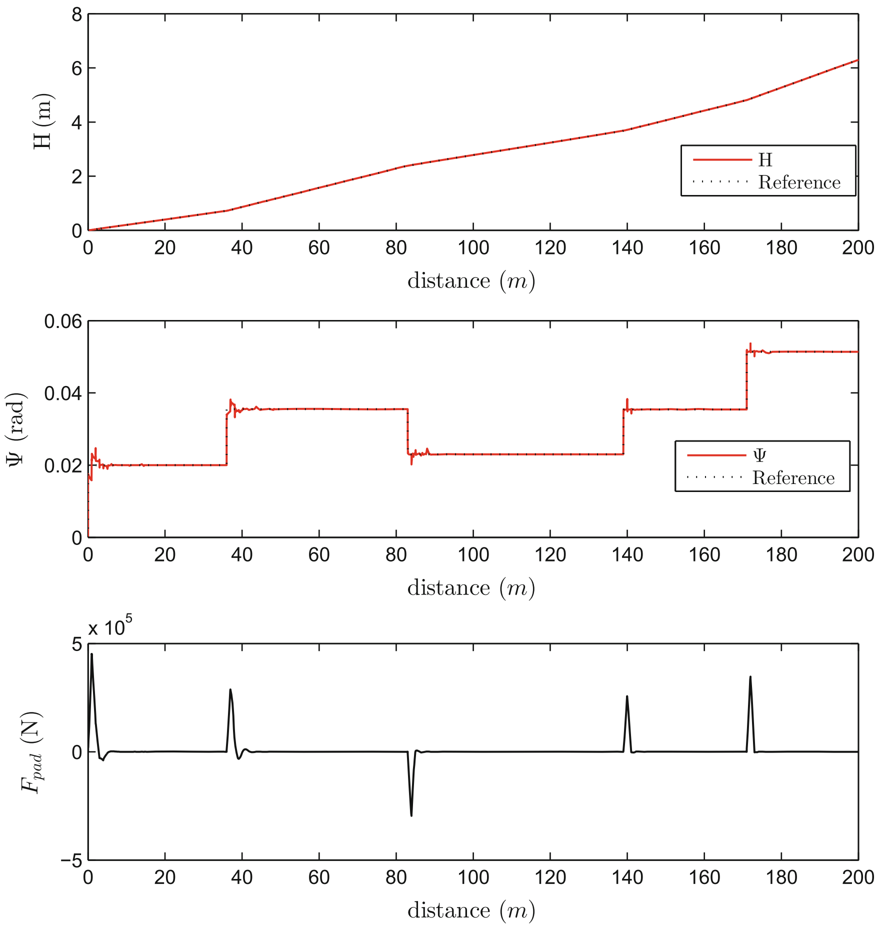


Fig. 4 I&I adaptive control (K_{anis} and WOB are changing, and the reference is non-smooth)

In the second case of simulation, the reference is selected to be $H_{ref}(m) = 0.1\sin(0.05m) + (m/200)$, and the uncertain parameters K_{anis} and WOB are varying randomly with Gaussian distribution. The tracking performance of the controller is presented in Fig. 3.

From Figs. 2 and 3, it can be seen that in both cases the I&I adaptive control is able to produce excellent stabilization performance while maintaining small tracking errors and acceptable control signals.

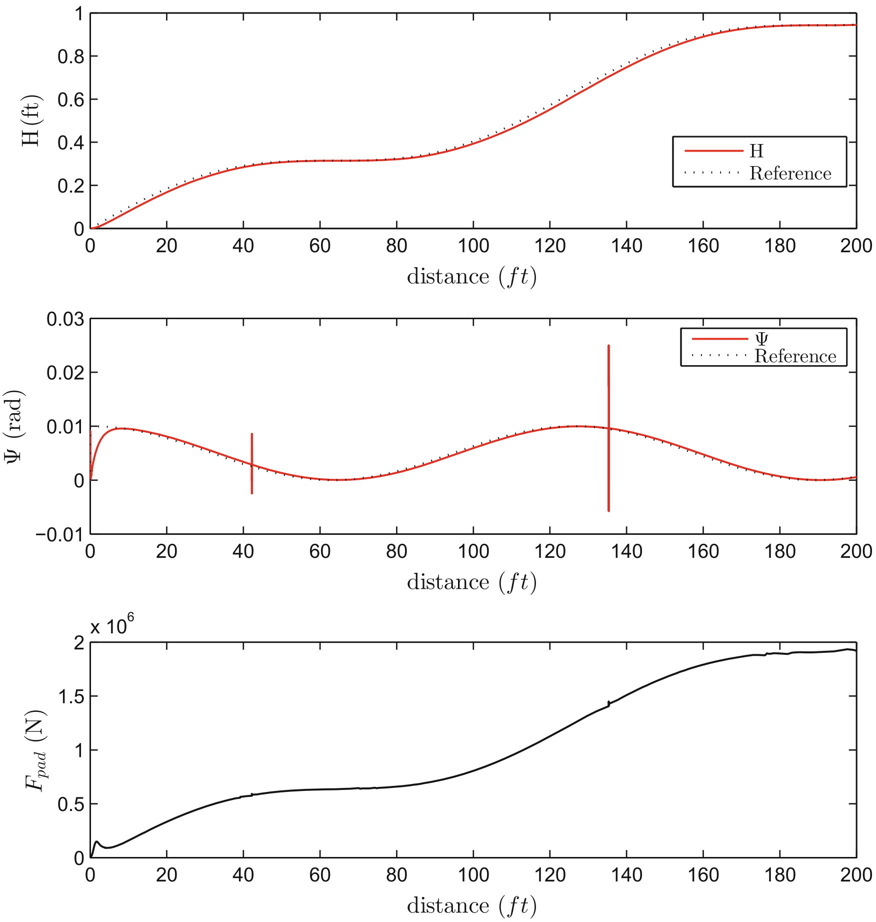


Fig. 5 $\mathcal{L}_1$ adaptive control ($K_{anis} = 10$ and $WOB = 8.9 \times 10^4$)

The last case of the I&I adaptive control simulation uses a non-smooth reference for the angle of borehole propagation with respect to the m-axis $\Psi_{ref}(rad)$ and Gaussian distributed uncertain parameters K_{anis} and WOB. Figure 4 shows that although the reference signal is non-smooth, the proposed controller is able to maintain its stability and performance in the presence of unknown and uncertain parameters. The control signal produced by the controller is also still bounded and does not exceed the constraint (Fig. 5).

Similar cases of simulations are conducted for $\mathcal{L}_1$ adaptive control scenario. The tuning parameters of the controller are selected according to the following norm condition:

$$\|G(s)\|_{\mathcal{L}_1}(\theta_{\max 0} + \theta_{\max 1} + \theta_{\max 2}) = 9.0181 \times 10^{-17} < 1 \tag{40}$$

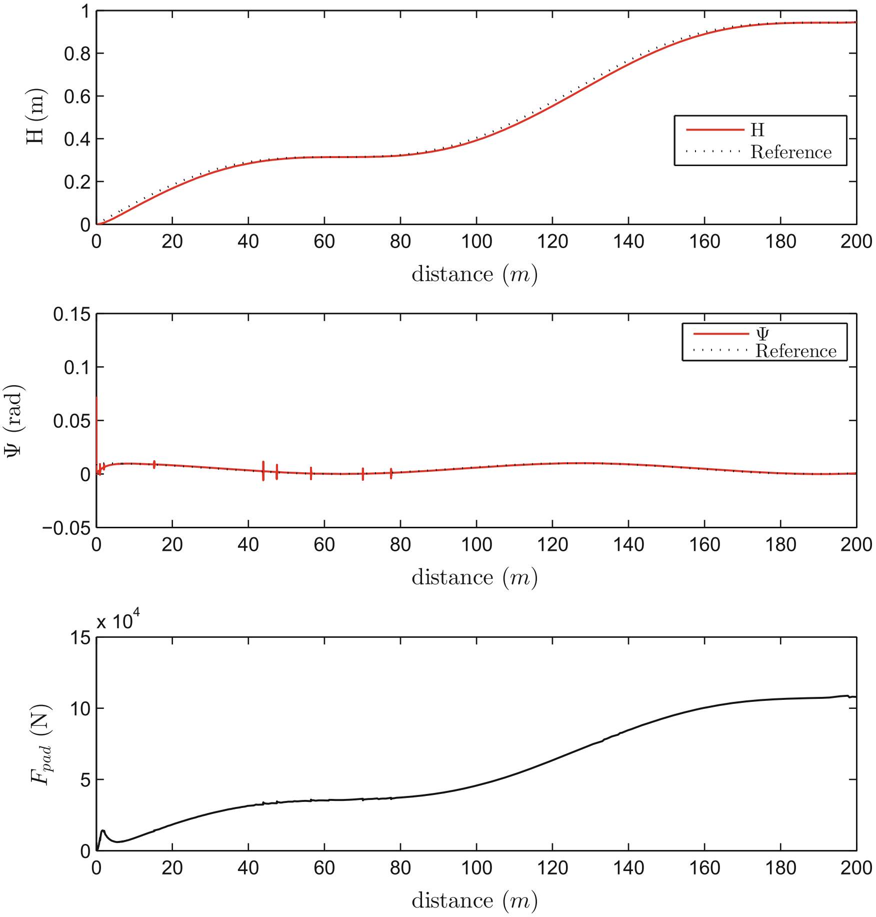


Fig. 6 $\mathcal{L}_1$ adaptive control ($K_{anis} = 1$ and $WOB = 5 \times 10^4$)

The results of the proposed $\mathcal{L}_1$ adaptive control can be seen in Figs. 6, 7 and 8. The performances of this controller are acceptable. However, if the results of the $\mathcal{L}_1$ controller are compared to the ones of I&I controller, it can be observed that the I&I control produces better performances than the $\mathcal{L}_1$ control does. In terms of the magnitude of the control signals, the I&I controller produces smaller and smoother signals than the $\mathcal{L}_1$ controller does. Hence, the I&I adaptive controller consumes less energy than $\mathcal{L}_1$ adaptive control does.

In terms of the tuning and implementation of the controllers in the simulation, for EFFSZM drilling system in particular, the I&I controller is better since it does not require complicated parameter tuning. In the $\mathcal{L}_1$ controller, even all the selection of the tuning parameters satisfy the $\mathcal{L}_1$-norm condition, several trials need to be

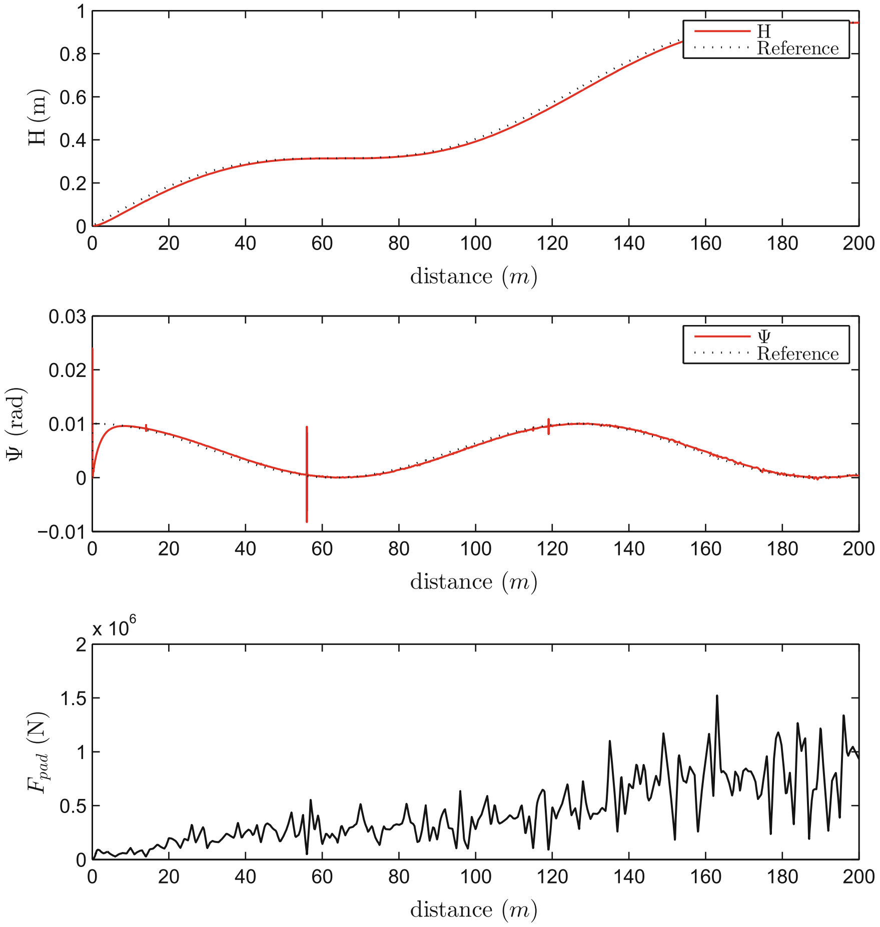


Fig. 7 $\mathcal{L}_1$ adaptive control (K_{anis} and *WOB* are changing)

done to obtain the suitable controller parameters. Overall, both the controllers produce acceptable tracking performances and are robust with respect to the parameter uncertainty and non-smooth references.

5 Conclusions

This paper presents two recent techniques in adaptive control, i.e. I&I adaptive control and $\mathcal{L}_1$ adaptive control, for the tracking stabilization of EFFSZM directional drilling system. The EFFSZM directional drilling system belongs to a class of uncertain systems with internal delays. Based on the simulations, both of the controllers

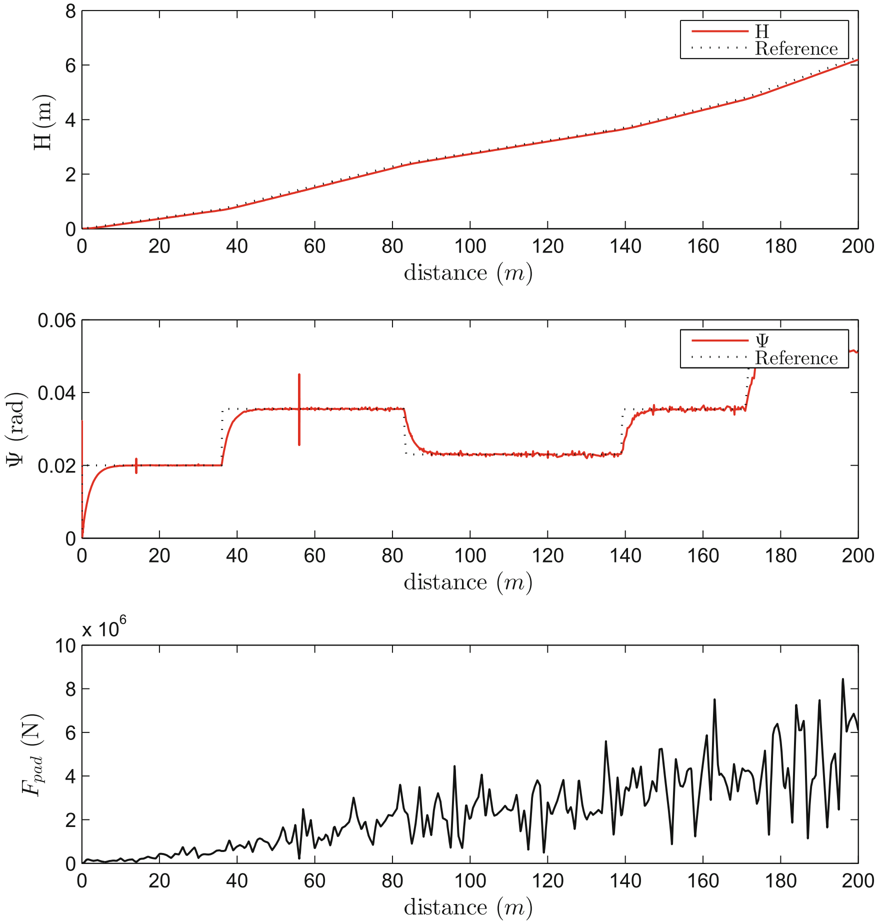


Fig. 8 $\mathcal{L}_1$ Adaptive Control (K_{anis} and *WOB* are changing, and the reference is non-smooth)

show promising tracking results in the presence of parameter uncertainty and non-smooth reference signal. However, the I&I adaptive control shows better performance compared to $\mathcal{L}_1$ adaptive control in terms of tracking error and energy consumption of the system. It is also interesting to note that the implementation of the I&I controller is simpler compared to that of $\mathcal{L}_1$ controller in terms of number of tuning parameters and their tuning process. In future work, more complex models of directional drilling system with more realistic assumptions could be considered and investigated.

References

1. Aalund, L., Rappold, K.: Horizontal drilling taps more oil in the middle east. Oil Gas J. **91**(25), 47–51 (1993)
2. Astolfi, A., Karagiannis, D., Ortega, R.: Nonlinear and Adaptive Control with Applications. Springer (2008)
3. Bayliss, M., Bogath, C., Whidborne, J., et al.: Mpc-based feedback delay compensation scheme for directional drilling attitude control. In: SPE/IADC Drilling Conference and Exhibition. Society of Petroleum Engineers (2015)
4. Deskins, W.G., McDonald, W.J., Reid, T.B.: Survey shows successes, failures of horizontal wells. Oil Gas J. **93**(25) (1995)
5. Downton, G.: Directional drilling system response and stability. In: IEEE International Conference on Control Applications, 2007. CCA 2007, pp. 1543–1550. IEEE (2007)
6. Elshafei, M., Baig, M., Mysorewala, M., Al-Majed, A., et al.: Control and optimization of directional drilling system. In: SPE Middle East Intelligent Oil and Gas Conference and Exhibition. Society of Petroleum Engineers (2015)
7. Molvar, E.M.: Drilling smarter: Using directional drilling to reduce oil and gas impacts in the intermountain west. Prepared by Biodiversity Conservation Alliance, Report issued Feb 18, 34 (2003)
8. Sun, H., Li, Z., Hovakimyan, N., Basar, T., Downton, G.: $\mathcal{L}_1$ adaptive controller for a rotary steerable system. In: 2011 IEEE International Symposium on Intelligent Control (ISIC), pp. 1020–1025. IEEE (2011)
9. Sun, H., Li, Z., Hovakimyan, N., Basar, T., Downton, G.: $\mathcal{L}_1$ adaptive control for directional drilling systems. In: Proceedings of the 2012 IFAC Workshop on Automatic Control in Offshore Oil and Gas Production, pp. 72–77. IFAC (2012)
10. van de Wouw, N., Kremers, N., Detournay, E.: Output-feedback inclination control of directional drilling systems. In: Proceedings of The 2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production, Florianpolis, Brazil, pp. 266–271. IFAC (2015)

Design and FPGA Implementation of a Fuzzy-PI Controller for Omnidirectional Robot System

Mohamed Slim Masmoudi, Najla Krichen, Arief B. Koesdwiady, Fakhri Karray and Mohamed Masmoudi

Abstract Fuzzy logic systems have been implemented successfully for the design of a wide variety of control systems. They provide a powerful way for designing nonlinear controllers using human expert knowledge. In this article, we present an approach to design and implement a fuzzy logic proportional integral controller (Fuzzy-PI) for omnidirectional robot navigation system, using a field-programmable gate array (FPGA). First, we define the kinematic model of the robot system and then we design, simulate, and optimize the controller navigation system using MATLAB and Robotino Sim platforms. The main goal of this work is the design of the Fuzzy-PI controller and the hardware implementation using FPGA resources. The controller can be implemented on an FPGA using software or hardware approach. For the latter approach, the Fuzzy-PI algorithm is implemented in VHDL language, synthesized, optimized, placed and routed, and downloaded on an FPGA board.

M.S. Masmoudi
METS Research Group-National School of Electronic and Communications of Sfax,
Technopark, BP 1163, CP 3018 Sfax, Tunisia
e-mail: med_slim_mas@yahoo.fr

N. Krichen
CEM Lab-National Engineers School of Sfax, BP 3000 Sfax, Tunisia
e-mail: najla_krichen@yahoo.fr

A.B. Koesdwiady (✉) · F. Karray
Centre for Pattern Analysis and Machine Intelligence,
University of Waterloo, Waterloo, ON, Canada
e-mail: abkoesdw@uwaterloo.ca

F. Karray
e-mail: karray@uwaterloo.ca

M. Masmoudi
METS Research Group-National Engineers School of Sfax,
Tunisia Route Soukra, Km 3.5, BP 1173, 3038 Sfax, Tunisia
e-mail: mohamed.masmoudi@enis.rnu.tn

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_12

1 Introduction

Omnidirectional mobile robots offer a great agility and flexibility during autonomous task due to their special structure. However, they are endowed with complex dynamics that make the extraction of an analytical model difficult. Therefore, we are motivated to propose the realization of a navigational task performed by an omnidirectional robot without deriving its mathematical model. This task is carried out through the development of an algorithm using a fuzzy logic-based proportional integral (PI) controller that controls the robot to navigate from an initial point to a desired final point. In this work, we are implementing a Fuzzy-PI algorithm to a robot, namely Robotino, which is developed by Festo Didactic.

Over the last few years, embedded systems are becoming more efficient and effective. They keep invading our daily lives in the means of communication and transport as well as the industrial world. Mobile robotics in particular have attracted a lot of attention in the recent studies done on certain branches of Artificial Intelligence (AI). For instance, soft computing algorithms have proven their ability to learn thanks to the their good performance for the approximation of nonlinear functions [4]. The most used algorithms in soft computing are neural networks, fuzzy logic, and genetic algorithm [1, 6, 8, 9]. Currently, academic and industrial researchers are trying to equip their mobile navigation platforms with some intelligence to perform tasks otherwise realized by human. This kind of robots provides some important faculties during navigational tasks such as perception, data processing, recognition, learning, reasoning, interpreting, decision-making, and action capacities [2]. Intelligent navigation systems currently utilize encoders to compute the actual position and speeds of the robot using kinematic model of omnidirectional robot. Several applications of a fuzzy logic controller to control the navigation of Robotino were developed in [9–11]. In addition, the most used algorithm to achieve mobile robot navigation task is PI or Fuzzy-PI algorithm, which is developed in [3, 5, 7].

This paper presents the design of a soft computing-based control algorithm, which is an ongoing project, intended to facilitate intelligent control of Robotino mobile robot navigation. It is in this context that this work involves the conceiving and testing of a Fuzzy-PI controller that will allow us to drive the Robotino from a starting point to a destination point. This choice stems from the need to avoid some of the drawbacks faced using the conventional PI controller. For example, each coordinate implicates a new operating point which means that a new set of PI parameters must be adjusted. This work intends to present the different steps to design, simulate, and experimentally test a Fuzzy-PI navigation controller of Robotino mobile robot.

The paper consists of three parts: first we started the navigation with a simple PI controller, then we improve the performances using a Fuzzy-PI navigation system, and finally we improve the implementation performances of this fuzzy controller using a more suitable platform based on a field-programmable gate array (FPGA) circuits. Furthermore, the paper is organized as follows: Sect. 2 introduces the kinematic model, PI controller, Fuzzy-PI architecture, and simulation results obtained using MATLAB environment. Section 3 presents the hardware design steps of the Fuzzy-PI controller. Finally, Sect. 4 draws the conclusion.

2 Fuzzy-PI Control for Robotino Navigation

2.1 Robotino Kinematics

Robotino is an omnidirectional navigation system with three coaxial driving wheels. The wheels are independently driven by three DC motors to achieve desired motion and orientation position, as shown in Fig. 1.

Robotino is equipped with three DC motors; nine GP2D120 sensors to measure the distance; a webcam; three optical encoder to provide position and speeds; bumper with integrated sensor; embedded PC-104 processor; and WLAN. The kinematic model of the robot is given as follows: [3]

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \frac{1}{r} \underbrace{\begin{bmatrix} -\sin(\phi + \psi_1) & \cos(\phi + \psi_1) & R \\ -\sin(\phi + \psi_2) & \cos(\phi + \psi_2) & R \\ -\sin(\phi + \psi_3) & \cos(\phi + \psi_3) & R \end{bmatrix}}_{P(\phi)} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \tag{1}$$

where

$$\begin{aligned} \dot{\theta}_i &\triangleq \text{angular speed of the whell } i. \\ V_i &\triangleq \text{linear speed of the wheel } i. \\ r &\triangleq \text{radius of the wheel.} \\ \dot{\phi} &\triangleq \text{angular speed of the robot.} \\ \psi_i &\triangleq \text{angular location of the wheel } i. \\ P(\phi) &\triangleq \text{transformation matrix.} \end{aligned} \tag{2}$$

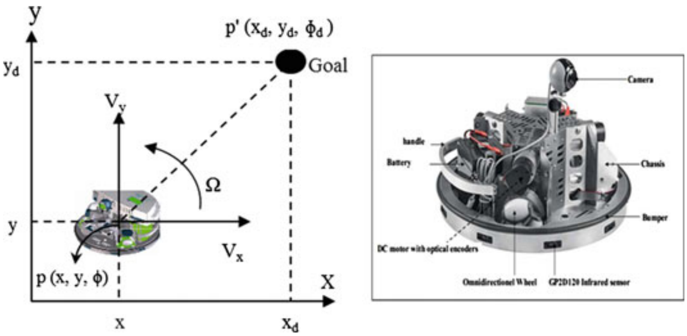


Fig. 1 Robotino schematic model and the image platform

2.2 PI Controller

The main objective of this work is to design a controller for the Robotino navigation. This controller determines the appropriate wheel speeds that enable the robot to move from its initial position $[x_0, y_0, \phi_0]^\top$ to a predefined desired position $[x_d, y_d, \phi_d]^\top$. The error equation between the actual and the desired position of the robot is given as

$$\begin{bmatrix} x_e \\ y_e \\ \phi_e \end{bmatrix} = \begin{bmatrix} x \\ y \\ \phi \end{bmatrix} - \begin{bmatrix} x_d \\ y_d \\ \phi_d \end{bmatrix} \tag{3}$$

According to [3], the derivative of the error equation is given as follows:

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\phi}_e \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = P^{-1}(\phi) \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \tag{4}$$

For the position control of the mobile robot, we have to design a stabilizer and find the controlled wheels velocity vector to navigate Robotino from any starting position to any final destination. To stabilize the robot, the following control law is proposed

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = P(\phi) \left(-K_p \begin{bmatrix} x_e \\ y_e \\ \phi_3 \end{bmatrix} - K_I \begin{bmatrix} \int x_e \\ \int y_e \\ \int \phi_e \end{bmatrix} \right) \tag{5}$$

where K_p and K_I are symmetric and positive definite proportional and integral gains. The robot speed according to the error position equation becomes

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\phi}_e \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} V_x \\ V_y \\ \Omega \end{bmatrix} = -K_p \begin{bmatrix} x_e \\ y_e \\ \phi_e \end{bmatrix} - K_I \begin{bmatrix} \int x_e \\ \int y_e \\ \int \phi_e \end{bmatrix} \tag{6}$$

The reference navigation error e is described in (1), and the PI discrete control law $u(k)$ is defined as follows:

$$u(k) = K_{pi} e_i(k) + K_{Ii} \sum_{j=1}^{k} e_i(j).T \tag{7}$$

where

$$\begin{aligned} &i = 1, 2, 3 \triangleq \text{ which wheel or omnidirectional speed is considered} \\ &k \triangleq \text{ the current discrete time instant} \\ &T \triangleq \text{ the sampling period} \end{aligned} \tag{8}$$

To determine the appropriate K_p and K_I parameters of the PI navigation controller, we use the successive approximation approach. This experimental method consists of gradual change of the controller parameters and observes Robotino navigation reaction to reach the optimal response. This method is simple and suitable for rapid processes with negligible dead time. The experimental results show that the PI controller has acceptable performance in all four quadrants navigations. However, this controller requires an adjustment to its parameter for each navigation target. Furthermore, the proposed Fuzzy-PI controller can be viewed as an improvement of the conventional PI navigation system.

2.3 *Fuzzy Adaptive PI Controller*

To improve the Robotino navigation performances, we develop a fuzzy adaptive PI controller that combines the advances of conventional PI with the fuzzy logic theory. Figure 2 depicts the structure of the control loop.

The fuzzy logic adaptive tuner has two inputs, i.e., the reference navigation error *e* and its variation *de*/*dt*; and two outputs ΔK_p and ΔK_I. The latter are the variations of proportional and integral gains, respectively. The input and output variables of the Sugeno fuzzy controller are defined by five membership functions *NB*, *NS*, *Z*, *PS*, and *PB*, as shown in Figs. 3 and 4.

Relying on human expertise and different experimental tests, we define a set of 25 linguistic rules to describe the behavior of the PI-Fuzzy logic controller as presented in Table 1.

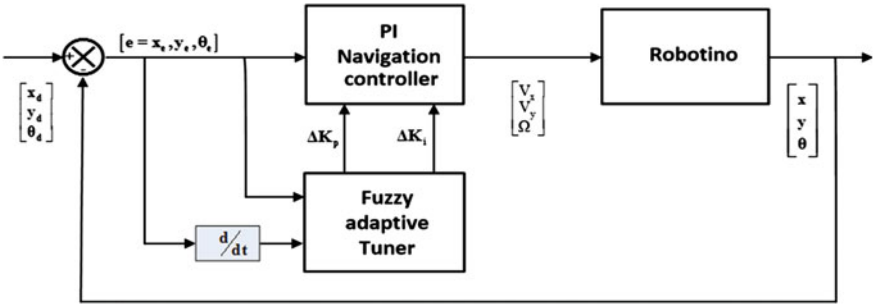

$[e = x_e, y_e, \theta_e]$
$\begin{bmatrix} x_d \\ y_d \\ \theta_d \end{bmatrix}$
PI Navigation controller
ΔK_p
ΔK_i
Fuzzy adaptive Tuner
d/dt
$\begin{bmatrix} V_x \\ V_y \\ \Omega \end{bmatrix}$
Robotino
$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$

Fig. 2 Fuzzy adaptive PI control loop

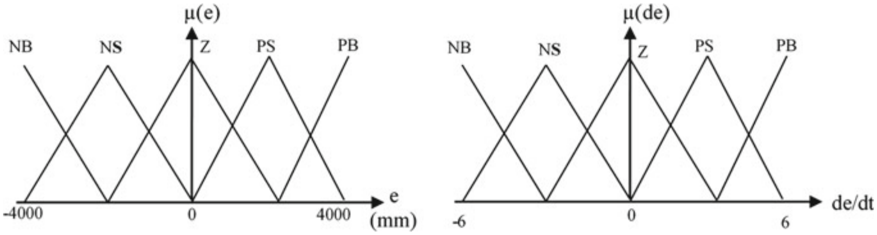


Fig. 3 Distances of input error (*left*) and error variation (*right*)

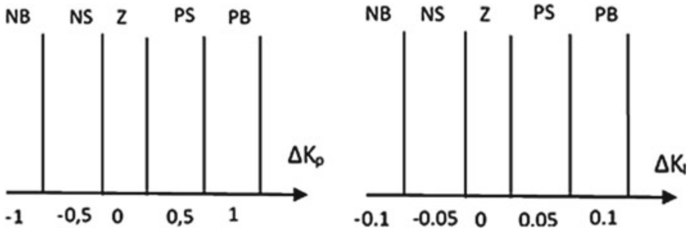


Fig. 4 ΔK_p output (*left*) and ΔK_I input (*right*)

Table 1 Rule-base and output weights of the Fuzzy adaptive PI

		Error variation (*de*/*dt*)									
		NB2		NS2		Z2		PS2		PB2	
		ΔK_p	ΔK_I	ΔK_p	ΔK_I	ΔK_p	ΔK_I	ΔK_p	ΔK_I	ΔK_p	ΔK_I
Error (e)	NB1	PB	NB	PS	NS	PS	NS	PS	NS	Z	Z
	NS1	PS	NB	PS	NS	PS	NS	PS	Z	NS	PS
	Z1	PS	NS	PS	NS	Z	Z	NS	PS	NS	PS
	PS1	PS	NS	Z	Z	NS	PS	NS	PS	NS	PB
	PB1	Z	Z	NS	PS	NS	PS	NS	PS	NB	PB

After the defuzzification task, the new PI controller parameters can be calculated using the following equations:

$$K_p = K_{p0} + \Delta K_p \tag{9}$$

$$K_I = K_{I0} + \Delta K_I \tag{10}$$

where K_{p0} and K_{I0} are the PI controller original parameters.

To simulate Robotino PI navigation algorithm, we use MATLAB and Robotino SIM environment tools. Subsequently, we validate the experimental step on Robotino. Simulation tests have shown that every time we change the destination, the robot reaches the final position and the desired angle (Fig. 5). The arrival to the destination position depends on the errors position values. The Robotino has to move from the

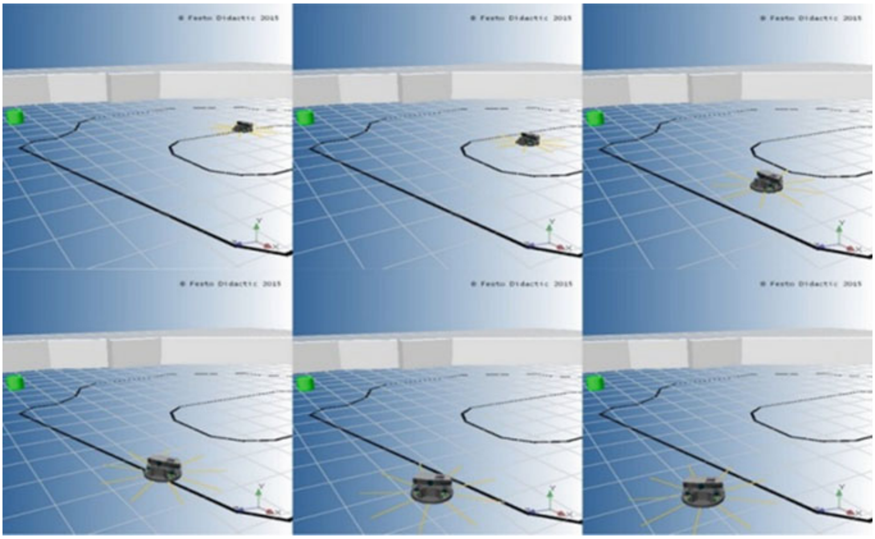


Fig. 5 Robotino SIM trajectory: target destination is (3500, 2000), and the final orientation angle is 30°

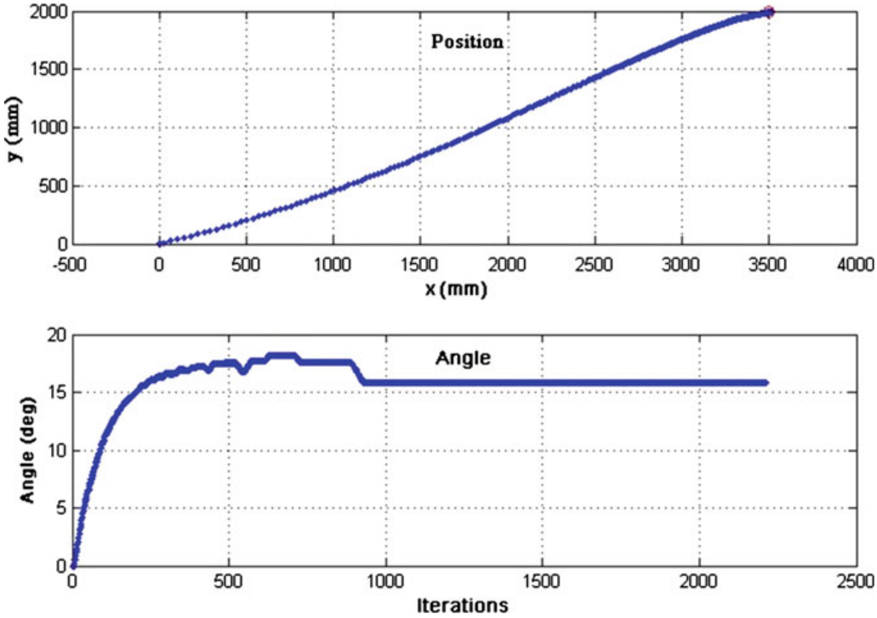


Fig. 6 Trajectory for the target position (3500, 2000), and the final orientation angle is 15°

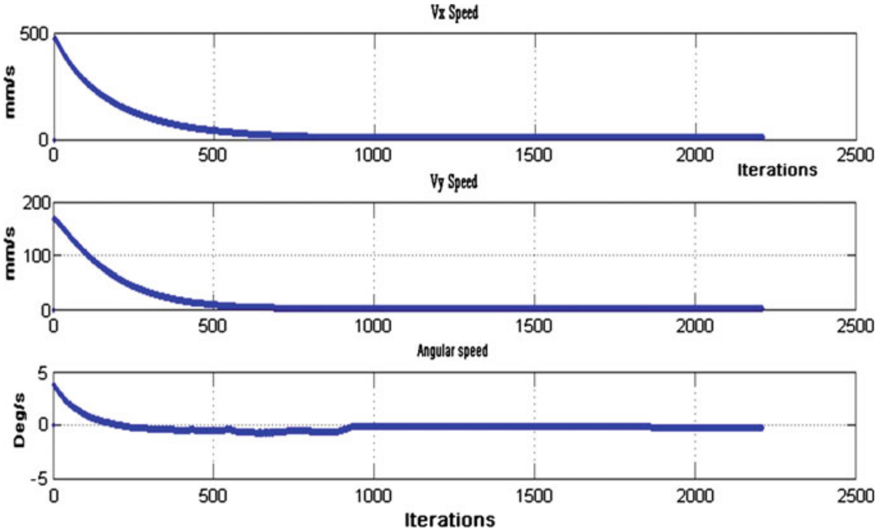
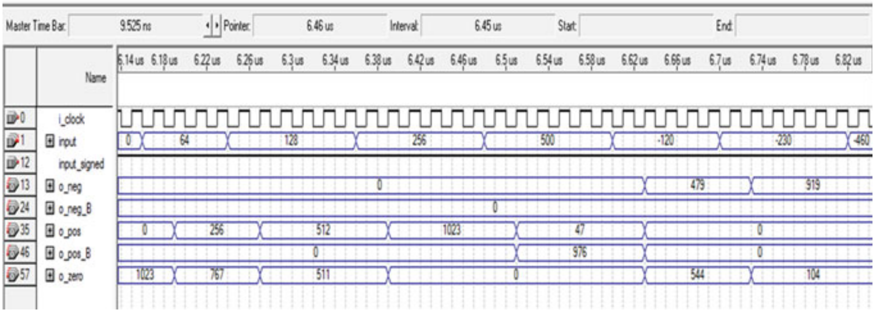


Fig. 7 Speed values to reach the final destinations, and the final orientation angle is 10°

initial position $[x_0, y_0, \phi_0]^\top$ to the desired position $[x_d, y_d, \phi_d]^\top$. As can be seen in Fig. 6, the Robotino can navigate in the four quadrants with different final positions. Figure 7 presents the output velocities of the proposed fuzzy-PI controller to achieve the stable trajectory tracking.

3 Hardware Implementation of the Fuzzy-PI Controller

Currently, our Robotino is controlled with a PC104 processor, which are clocked at a frequency of 300 MHz with 1GB DDRRAM, 10/100 MBit Ethernet connection, 2 USB and used a real-time Linux operating system. Our future project aims at the real-time control, navigation, and obstacle avoidance of Robotino using HD cameras and the integration of an absolute location system. These applications require more powerful processing platforms such as FPGAs. Part of this project is to validate the VHDL description of the Fuzzy-PI controller navigation using Altera Quartus environment. As detailed in the second part of this paper, the Sugeno Fuzzy-PI controller uses two inputs (e and de/dt) and two outputs (ΔK_p and ΔK_I). The VHDL algorithm is composed of three main components: the fuzzifier, the inference engine, and the defuzzifier.



Master Time Bar: 9.525 ns

Pointer: 6.46 us

Interval: 6.45 us

Start:

End:

Fig. 8 Fuzzifier simulation result

3.1 Fuzzifier Hardware Module

Two fuzzifiers are used to compute the membership degree of crisp inputs in each membership function. The first input and the second input variables are fuzzified with five membership functions. Two fuzzification blocks described by ten membership functions (NB1, NB2, N1, N2, Z1, Z2, P1, P2, PB1, and PB2). In this work, triangular membership functions are applied, as it can be easily implemented in VHDL. To generate the hardware model of the FLC, the equation approach is used. FPGA implementation is a complicated task, which requires the VHDL description of each elementary computing operation for the description of the fuzzy controller. Figure 8 presents a screen shot associated with the functional simulation results of the fuzzifier module for signed input.

3.2 Inference Engine and Defuzzifier Hardware Module

The defuzzifier module is used to generate the outputs ΔK_p and ΔK_I according to (11). The module needs twenty-five linguistic rules and uses arithmetic processing units, e.g., minimum, multiplication, addition, and division blocks, in order to compute the Fuzzy-PI tuning outputs.

$$\Delta K_x = \frac{\sum_{j=1}^{25} \omega_j z_j}{\sum_{j=1}^{25} \omega_j} \tag{11}$$

Figure 9 presents a screen shot associated with the simulation results of ΔK_p value using the defuzzifier module for different values of membership functions. As illustrated in this figure, Fuzzy-PI ΔK_p output of the defuzzifier block is generated five clock cycles after changing the membership values. The Fuzzy-PI hardware implementation consumes 6015 logical elements (LE) and 80 DSP blocks (Fig. 10).

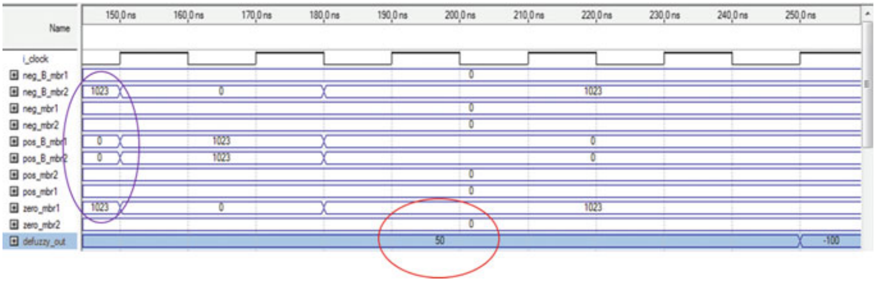


Fig. 9 Quartus II defuzzifier simulation result for signed input

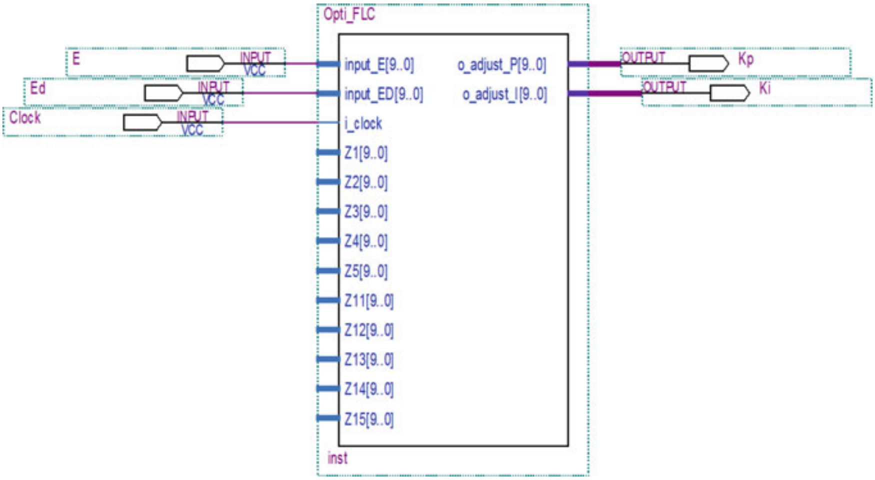


Fig. 10 Fuzzy-PI Quartus design

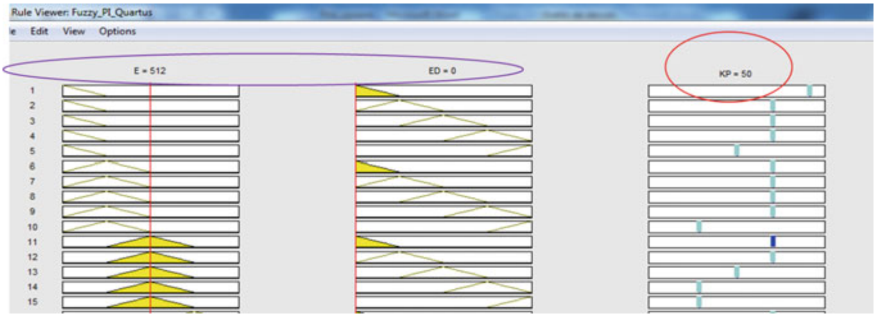


Fig. 11 MATLAB defuzzifier simulation result for signed input

To validate the FPGA implementation accuracy, we compare the VHDL Quartus II design and MATLAB FIS code simulation results. Finally, for an input vector values ($e = 512$ and $de/dt = 0$), the fuzzification results are $Z1 = 1023$ and

$NB2 = 1023$. In conclusion, both the implementation and simulation provide the same output value ($\Delta K_p = 50$), as shown in Fig. 11.

4 Conclusion

In this paper, the design and hardware implementation of a Fuzzy-PI controller for the mobile robot navigation system are presented. We start with the development of Robotino kinematic model, which is followed by the design of a PI controller for the navigation system. The PI controller has shown its inferiority in the accuracy. Moreover, we are forced to continuously change the PI controller parameters to allow the robot to reach the desired position and angle. To overcome these limitations, we design a second fuzzy controller to adjust the PI controller parameters. The Fuzzy-PI controller is able to perform navigation tracking with acceptable accuracy.

The second part of our work is an improvement of the Robotino control platform. Indeed, the current platform lacks several interfaces; memory is limited; and the processor is not fast and efficient enough to compute intelligent algorithms, which require more speed, real-time and embedded aspects. FPGAs seems more promising in our application. VHDL implementation and simulation results, using Quartus II environment, confirm the performances of the preferred approach in terms of speed, circuit resources, and the reliability of outputs variables. In addition, the results of the implementation are compared with the one of the MATLAB simulations for validation.

References

1. Benjemmaa, A., Slim Masmoudi, M., Masmoudi, M., et al.: Optimum architecture of neural networks lane following system. Int. J. Comput. Appl. **42**(12), 41–46 (2012)
2. Huang, H.C.: Sopc-based parallel aco algorithm and its application to optimal motion controller design for intelligent omnidirectional mobile robots. IEEE Trans. Ind. Inf. **9**(4), 1828–1835 (2013)
3. Huang, H.C., Wu, T.F., Yu, C.H., Hsu, H.S.: Intelligent fuzzy motion control of three-wheeled omnidirectional mobile robots for trajectory tracking and stabilization. In: 2012 International Conference on Fuzzy Theory and it's Applications (iFUZZY), pp. 107–112. IEEE (2012)
4. Karray, F.O., De Silva, C.W.: Soft Computing and Intelligent Systems Design: Theory, Tools, and Applications. Pearson Education (2004)
5. Oltean, S., Dulau, M., Puskas, R.: Position control of robotino mobile robot using fuzzy logic. In: 2010 IEEE International Conference on Automation Quality and Testing Robotics (AQTR), vol. 1, pp. 1–6. IEEE (2010)
6. Rossomando, F., Soria, C.: Identification and control of nonlinear dynamics of a mobile robot in discrete time using an adaptive technique based on neural pid. Neural Comput. Appl. 1–13 (2014)
7. Sheikhlar, A., Fakharian, A., Adhami-Mirhosseini, A.: Fuzzy adaptive pi control of omnidirectional mobile robot. In: 2013 13th Iranian Conference on Fuzzy Systems (IFSC), pp. 1–4. IEEE (2013)

8. Takahashi, Y., Seki, H.: Fuzzy logic based regenerative braking control system of electric wheelchair for senior citizen. In: IEEE International Conference on Rehabilitation Robotics, 2009. ICORR 2009, pp. 725–730. IEEE (2009)
9. Tian, D., Wang, S., El Kamel, A.: Fuzzy controlled avoidance for a mobile robot in a transportation optimisation. In: 2011 International Conference on Fluid Power and Mechatronics (FPM), pp. 868–872. IEEE (2011)
10. Treesatayapun, C., Guzman-Carballido, A.: Linearization based on fuzzy rules emulated networks for nonaffine discrete-time systems controller. In: TENCON 2009-2009 IEEE Region 10 Conference, pp. 1–6. IEEE (2009)
11. Weinert, H., Pensky, D.: Mobile robotics in education and student engineering competitions. In: AFRICON, 2011, pp. 1–5. IEEE (2011)

Android-Based Mobile Robotic Platform Performance Testing for Real-Time Navigation

Peter Pásztó, Martin Smol'ák, František Duchoň, Peter Hubinský and L'uboš Chovanec

Abstract This paper discusses the suitability of usage of smartphones running on Android operating system in robot navigation tasks. These phones are equipped with many sensors useful in robotics. Robot navigation tasks are often recommending high-performance devices (computers) to work on. Nowadays smartphones are also equipped with very powerful processors. Therefore a performance test of different devices with Android operating system including their sensorial equipment has been made to determine the usability of these devices for real-time robot navigation.

Keywords Android · Accelerometer · Gyroscope · Compass · Camera · Laser scanner · Mobile robot · Image processing

1 Introduction

Robotics today relies on processing and fusion of data from many kinds of sensors. It is possible because of the existence of high-performance devices and sensors. For robot navigation and localization tasks GPS sensors, ultrasonic sensors, accelerometers, gyroscopes, laser scanners, cameras and image processing algorithms and many other sensors are often used.

Usually, it could be a time-consuming work to set up an application that is able to read and process all the data from the used sensors. Every sensor has its own communication protocol and can be connected to computer (or other supervisor device) in different ways (USB, serial port, Bluetooth, Wi-Fi, etc.). It is a long work to create a robotic platform, which is able to receive all data from all sensors used. Also a high-performance device is usually needed to handle and process the received data.

P. Pásztó (✉) · M. Smol'ák · F. Duchoň · P. Hubinský · L'. Chovanec
Faculty of Electrical Engineering and Information Technology, Institute of Robotics and Cybernetics, Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19 Bratislava, Slovak Republic
e-mail: peter.paszto@stuba.sk; peterpaszto@gmail.com

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_13

The onset of Android devices brought some new opportunities into robotics. These devices are small-sized with a relatively high performance, but the most important is their sensorial equipment. Most of these devices implement an accelerometer, magnetometer, gyroscope, GPS, camera and many other sensors usable in robotics [1–4]. However, the performance and the accuracy of these sensors are often questioned [5].

The aim of this article is to test the performance of different Android devices in terms of the time needed for retrieving and processing data from more sensors and the possibility of further real-time navigation of a four-wheeled mobile robot based on these data.

The mentioned test includes a design of a four-wheeled mobile robot [6] and an Android application, which is reading the data from the sensors of the mobile device and is communicating with the mobile robot (with servo motors and DC motors). The Android application is also obtaining the data from an external laser scanner (Hokuyo URG-04LX) and uses the OpenCV image processing library [7, 8].

The whole sensorial equipment processed is: camera (with OpenCV library), GPS (including Google Maps visualisation), compass, accelerometer, gyroscope, Hokuyou URG-04LX laser scanner connected over USB port, device's orientation computation and Bluetooth communication with the mobile robot's Arduino board.

2 The Four-Wheeled Mobile Robotic Platform

2.1 Hardware of the Robotic Platform

The mobile robotic educational platform consists of a chassis of an electric RC off-road four-wheeled car model with modified control using Arduino board. Arduino is controlling two servo motors and two DC motors of the chassis on the base of instructions coming from a supervisor mobile device running on Android operating system.

The communication between the Arduino and the mobile device is established over Bluetooth. Motors, external sensors and the Arduino board are powered with two 6800 mAh, 12,6–10,8 VDC Li-ion batteries (Table 1).

The developed Android application is able to send control instructions to Arduino in the terms of setting the turning angle of the mobile robot's wheels and their rotation speed. The application is also handling the communication with device's internal and external sensors and their visualisation. Device's external sensors monitoring process is also running on another application thread and is monitoring and processing the Hokuyo URG-04LX laser scanner data [9]. A short description of this laser scanner is shown at Table 2. The scanner is connected to the device through USB port (Fig. 1).

Table 1 Description of internal sensors of Android devices

Sensor type	Description	Unit
Linear accelerometer	Acceleration forces along *x*, *y* and *z* axes excluding gravity	$\frac{m}{s^2}$
Gyroscope	Rates of rotations around *x*, *y* and *z* axes	$\frac{rad}{s}$
Orientation	Azimuth, pitch and roll (angles around *x*, *y* and *z* axes) used for computation of the deviation from north	Degrees
Accelerometer and Magnetometer	Sensors used for device's orientation computation by obtaining a rotation matrix. Sensors are measuring the acceleration forces and geomagnetic field strengths around *x*, *y* and *z* axes	$\frac{m}{s^2}$, μT; rad
Camera	Used for real-time image processing with OpenCV library for Android	

Table 2 Hokuyo URG-04LX parameters short description

Property	Description
Detection	60–4095 mm
Accuracy	Distance 20–1000 mm: ±10 mm Distance 1000–4000 mm: ±1 % of measurement
Resolution	1 mm
Scan angle	240°
Angular resolution	0.36° (360°/1024)

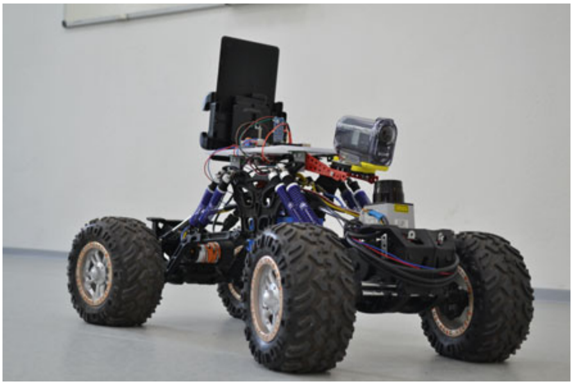


Fig. 1 Four-wheeled mobile robotic platform

2.2 Android Application

The monitored internal sensors (types and descriptions) are shown at Table 1 following the Android developers guide [10]. The communication with mobile device's internal and external sensors and the visualisation processes are running on different

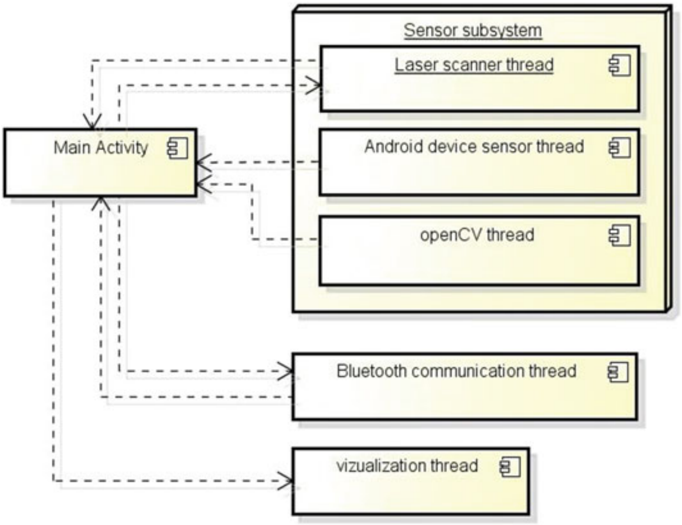


Fig. 2 Android application structure

application threads (Fig. 2), that means every sensor data is retrieved as soon as possible after sensor value change occurrence and is visualised also immediately on the visualisation thread.

3 The Performance Test

The performance test is made on these devices: Samsung I9100 Galaxy S II, Asus Google Nexus 7 and Sony Xperia Z3 compact. Their specifications are at Table 3. [11–13].

The used devices are not rooted and during the performance test there are running all the pre-installed OS implemented background processes. Also the resolutions shown at Table 3 are the default resolutions when programmatically opening

Table 3 Description of devices parameters used in the performance test

Device	API level	CPU	RAM (GB)	Processed image resolution
Samsung I9100 Galaxy S II	Android 4.0.4 API level 15	Dual-core 1.2 GHz Cortex-A9	1	352 × 288
Asus Google Nexus 7	Android 4.4.4 API level 19	Quad-core 1.2 GHz Cortex-A9	1	1024 × 768
Sony Xperia Z3 compact	Android 4.4.4 API level 19	Quad-core 2.5 GHz Krait 400	2	720 × 480

the device's camera. The aim of the performance test is not to compare these devices, but to decide the suitability of different devices with Android OS for robot navigation tasks in real time.

The test is based on the previously described Android application running on the devices. This application is monitoring all the sensors of the devices—on every value change the application writes the actual timestamp into a text file on the SD card. A unique text file is created for every sensor. The test is made twice—while all the sensors actual values visualisation is turned on and off. Every device is fixed to the mobile robot which is driving in an indoor environment during the timestamps recording process. Therefore the measurement on the GPS sensor is excluded from this test.

3.1 Real-Time Navigation Suitability Criterion

The criterion for the suitability of the devices for mobile robot navigation in real time is coming from the needed ability of the mobile robot to react to any obstacle in its way in any time and case. It means that the mobile robot with a given speed must travel a shorter distance between obtaining two samples from sensors than the minimal measuring value of these sensors.

The camera and/or the laser scanner can be used for obstacle detection in our case. The minimal distance of an obstacle that laser scanner can measure (Table 2) is 60 mm. We will use this figure because the minimal measurable distance of obstacle with the camera depends on the size of the obstacle and anyway, the image processing algorithms increase the sampling time of the camera. Also we consider a normal speed of the mobile robot about $1\,\frac{\mathrm{km}}{\mathrm{h}}$, while its maximal speed is about $3\,\frac{\mathrm{km}}{\mathrm{h}}$.

The above results in the criterion that the sampling time of every sensor of the used mobile devices must be under approx. 215 ms.

4 Measurement

Measurements were made to determine the average, maximal and minimal sample time of every sensor on every device during the visualisation turned on and off. The sample time is plotted to a figure for every sensor measured on three devices. The effect of turned off visualisation process on the sampling time for each sensor and each device has been also verified. These results are shown on the following figures.

Measurements are shown in histograms, those count the number of occurrences of each measured sampling time value of the sensors. Such histogram allows calculating the maximal and minimal sampling time that occurred during the measurement (and number of their occurrences) as well as the average sampling time and the most recurrent sampling time.

Internal sensors of the devices those data are acquired and visualised without the need of processing (accelerometer, compass and gyroscope - further called “internal sensors”) were measured with 9000 samples. The rotation of the device is computed by using the gravity and the geomagnetic field sensor by obtaining a rotation and an inclination matrix [10, 14] (further called “software sensor” or “rotation sensor”). The measurement was made by obtaining 3000 samples. Measurements on devices cameras were made by obtaining 500 samples and on the external laser scanner by obtaining 200 samples.

Results are shown from Figs. 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 and 14 and from Tables 4, 5 and 6.

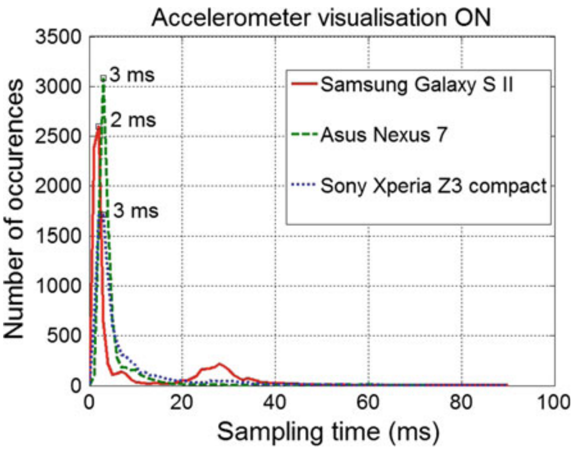


Fig. 3 Sampling time histogram of accelerometer when visualisation is turned on

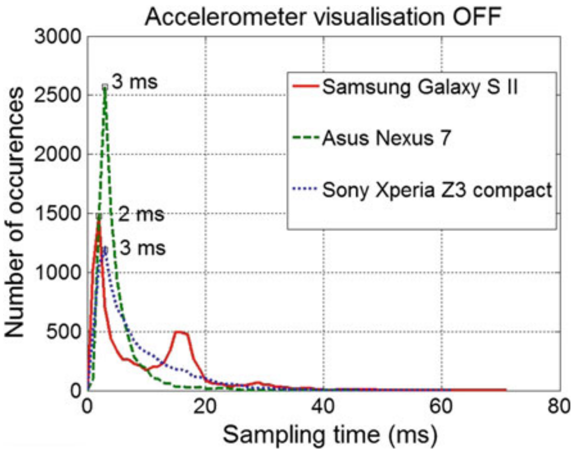


Fig. 4 Sampling time histogram of accelerometer when visualisation is turned off

Fig. 5 Sampling time histogram of compass when visualisation is turned on

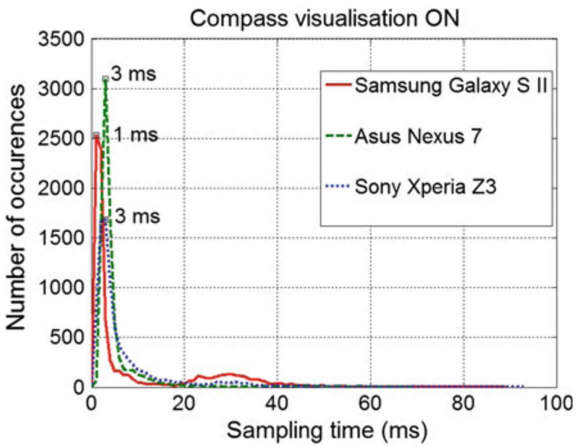


Fig. 6 Sampling time histogram of compass when visualisation is turned off

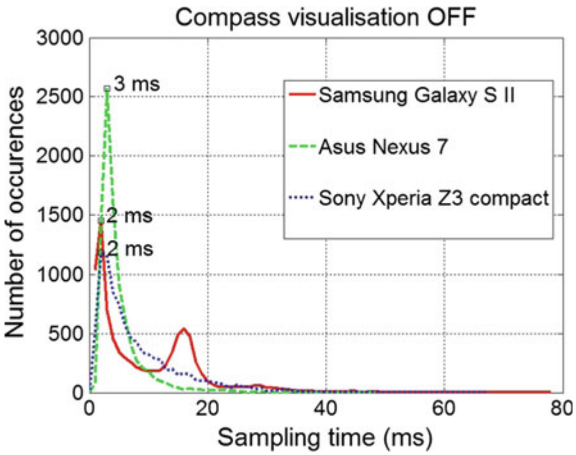


Fig. 7 Sampling time of gyroscope when visualisation is turned on

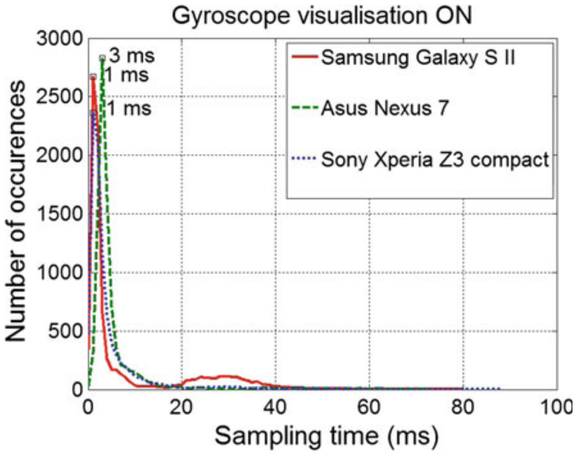


Fig. 8 Sampling time of gyroscope when visualisation is turned off

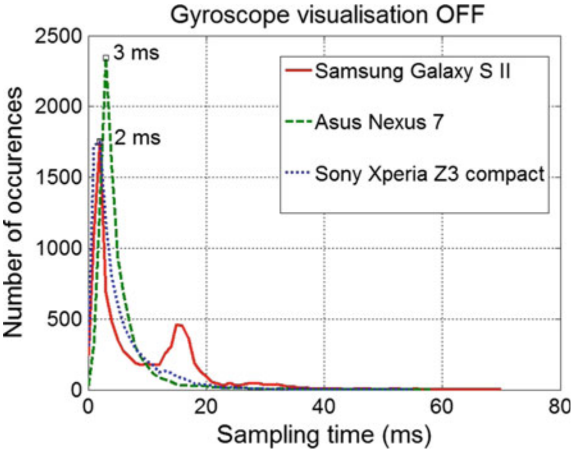


Fig. 9 Sampling time histogram of rotation calculation when visualisation is turned on

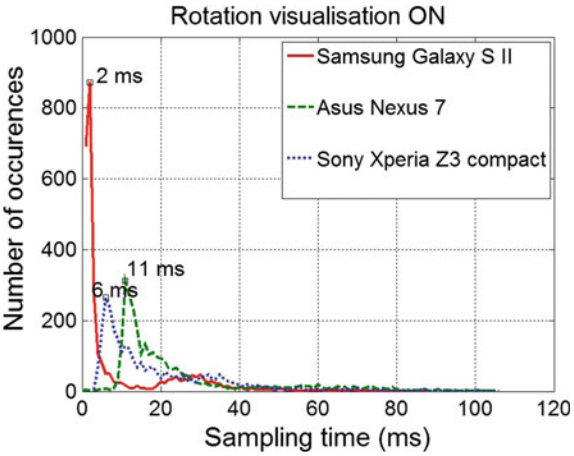


Fig. 10 Sampling time histogram of rotation calculation when visualisation is turned off

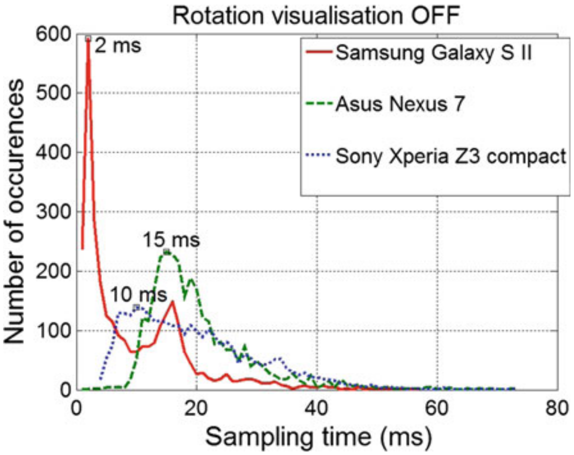


Fig. 11 Sampling time histogram of image processing when visualisation is turned on

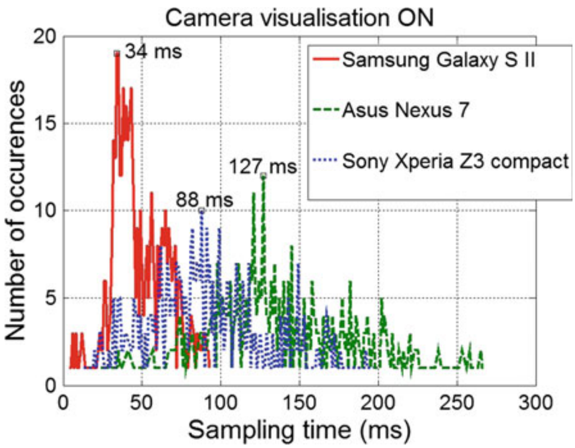


Fig. 12 Sampling time histogram of image processing when visualisation is turned off

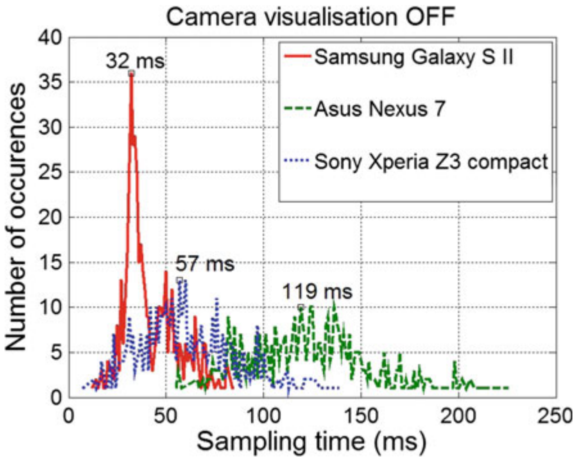
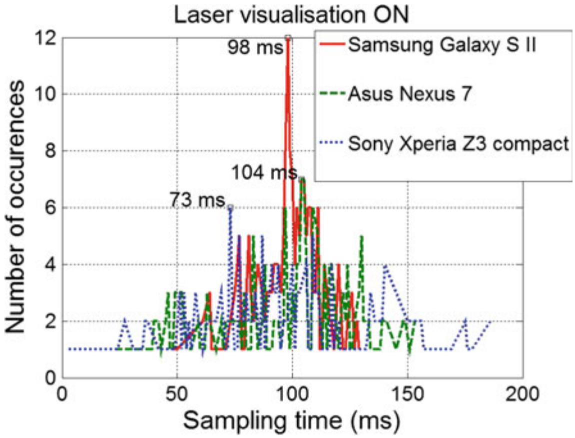


Fig. 13 Sampling time histogram of laser scanner data processing when visualisation is turned on



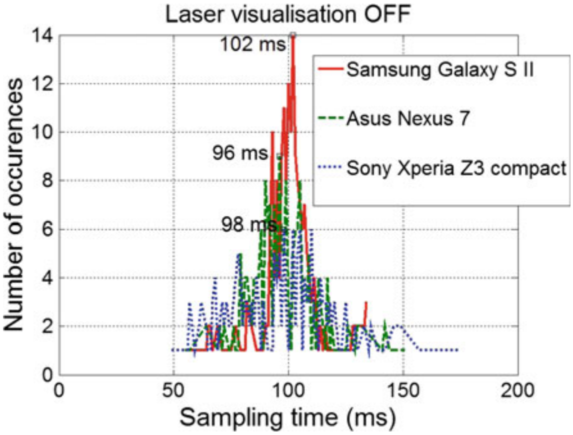


Fig. 14 Sampling time histogram of laser scanner data processing when visualisation is turned off

Table 4 Performance test measurement results for Samsung Galaxy S II

Sensor	Samsung Galaxy S II					
	Sampling time (ms)					
	Number of samples	Visualisation	Average	Min	Max	Most recurrent
Accelerometer	9000	On	10.1432	0	90	2
		Off	10.2653	0	71	2
Compass	9000	On	10.2173	0	89	1
		Off	10.2869	1	78	2
Gyroscope	9000	On	9.0041	0	80	1
		Off	9.0453	0	70	2
Rotation	3000	On	9.897	1	89	2
		Off	9.6893	1	57	2
Camera	500	On	48.598	4	93	34
		Off	42.046	12	85	32
Laser scanner	200	On	97.63	48	129	98
		Off	98.935	55	134	102

The next figures show the impact of turned off visualisation on the sampling time of each sensor and for each device. These data come from the data measured before, but are shown in groups for each sensor when visualisation is turned on and off.

Table 5 Performance test measurement results for Asus Nexus 7

Sensor	Asus Nexus 7					
	Sampling time (ms)					
	Number of samples	Visualisation	Average	Min	Max	Most recurrent
Accelerometer	9000	On	5.9411	0	73	3
		Off	5.1178	0	46	3
Compass	9000	On	5.9438	0	69	3
		Off	5.1503	1	49	3
Gyroscope	9000	On	5.9297	0	73	3
		Off	5.1380	0	58	3
Rotation	3000	On	24.7420	0	105	11
		Off	20.6227	1	73	15
Camera	500	On	143.004	24	267	127
		Off	123.914	55	226	119
Laser scanner	200	On	96.3950	23	153	104
		Off	98.6617	55	151	96

Table 6 Performance test measurement results for Sony Xperia Z3 compact

Sensor	Sony Xperia Z3 compact					
	Sampling Time (ms)					
	Number of samples	Visualisation	Average	Min	Max	Most recurrent
Accelerometer	9000	On	8.0679	0	90	3
		Off	8.1522	0	62	3
Compass	9000	On	8.1007	0	93	3
		Off	8.1764	0	67	2
Gyroscope	9000	On	4.8882	0	88	1
		Off	4.9247	0	55	2
Rotation	3000	On	19.5890	3	106	6
		Off	19.6677	4	73	10
Camera	500	On	91.9800	13	197	88
		Off	61.6960	7	139	57
Laser scanner	200	On	95.4550	3	186	73
		Off	98.4000	49	175	98

5 Results

The figures show that the most recurrent sampling time of internal sensors is about 1–3 ms depending on the device. But from the view of real-time navigation the most recurrent sampling time is not the most important quantifier, because a real-time system has to guarantee a response within a strict time constrain (deadline). Real-time systems are further classified by the consequence of missing a deadline into hard, firm and soft real-time systems [15].

If a real-time system must not miss a deadline, the maximal time response to an event should be considered. In this case the maximal measured sampling time of the internal sensors (accelerometer, compass and gyroscope) was 93 ms (Fig. 5 compass of Sony Xperia Z3 compact, with visualisation turned on).

For the rotation sensor, the maximal measured time response is 106 ms (Fig. 9 Sony Xperia Z3 compact with visualisation turned on), for camera 267 ms (Fig. 11 Asus Nexus 7 with visualisation turned on) and for the laser scanner 186 ms (Fig. 13 Sony Xperia Z3 compact with visualisation turned on).

It may be confusing that turned off visualisation on Figs. 15, 16, 17, 18, 19 and 20 often leads to increasing the most recurrent sampling times of the sensors. But the overall performance of the devices is important—the maximal time responses are almost always noticeably decreased (the proof could also be seen higher—all the maximal measured time responses were with the visualisation turned on). With visualisation off, the maximal time response of internal sensors was 78 ms, rotation sensor 73 ms, camera 226 ms and laser scanner 175 ms.

Another confusion could come from the illusion of higher performance of Samsung Galaxy S II with dual core processor in comparison with other quad-core processors. This effect is caused by the significantly lower camera resolution that this device has to process. Image processing is usually the most processor demanding task and therefore this device has more performance available to process data for other sensors than the devices processing a higher resolution image.

The aim of this paper is to determine the suitability of devices running on Android operating system for mobile robot navigation tasks. This can be done by computing the distance travelled by the mobile robotic platform with its maximal speed during the longest sampling time measured (Table 7).

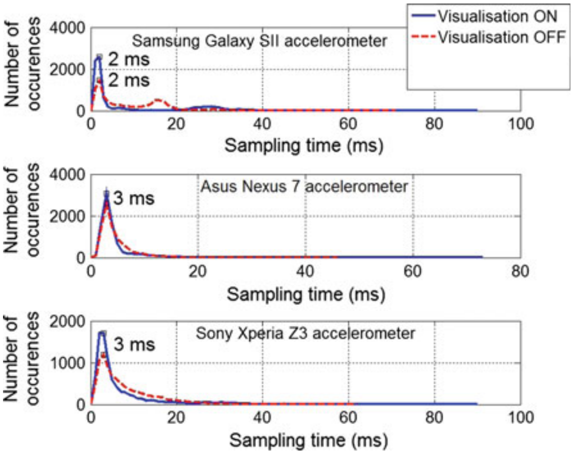


Fig. 15 The impact of visualisation turned on/off on sampling time of accelerometer

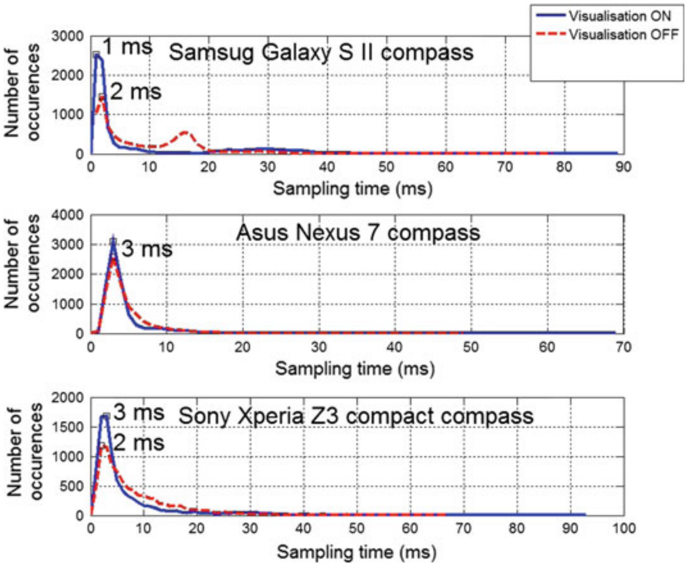
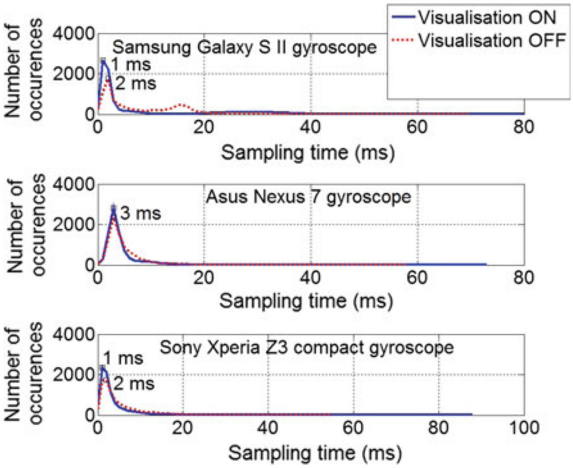


Fig. 16 The impact of visualisation turned on/off on sampling time of compass

Fig. 17 The impact of visualisation turned on/off on sampling time of gyroscope



The measured maximal speed of the mobile robotic platform is approximately $v_{max} = 0.9\,\frac{\text{m}}{\text{s}}$. If we consider a laser scanner usage for avoiding the collision with an obstacle, the maximal distance that robot can travel during the maximal sampling time of this scanner is:

$$s_{travelled} = v_{max} \cdot t_{max} \tag{1}$$

$$s_{travelled} = 0.9\,\frac{\text{m}}{\text{s}} \cdot 0.186\,\text{s} \tag{2}$$

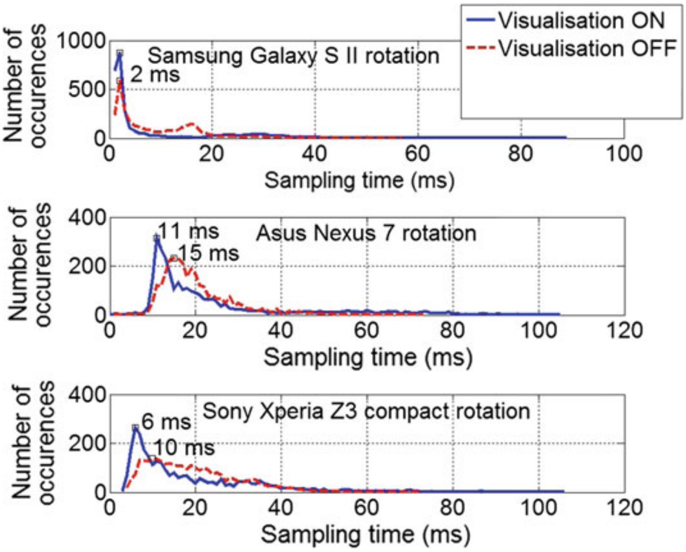


Fig. 18 The impact of visualisation turned on/off on rotation computation time

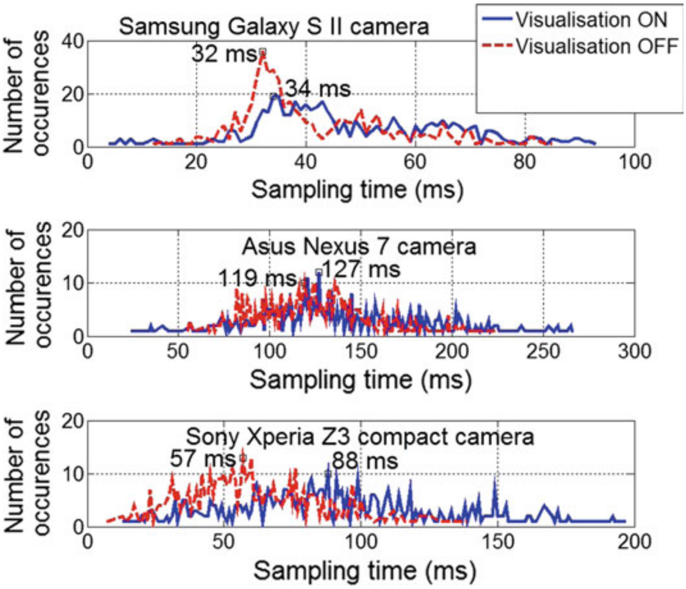


Fig. 19 The impact of visualisation turned on/off on sampling time of camera

Fig. 20 The impact of visualisation turned on/off on sampling time of laser scanner

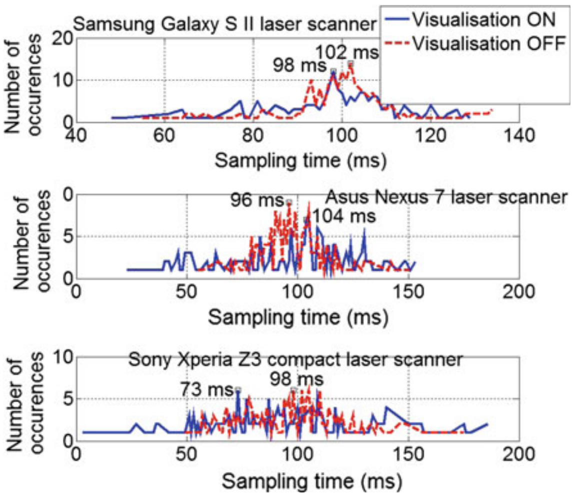


Table 7 The theoretical distance that mobile robot could travel during the sampling time of each sensor

Sensor	Travelled distance visualisation on (m)	Travelled distance visualisation off (m)
Internal sensors	0.0837	0.0702
Rotation	0.0954	0.0657
Camera	0.2403	0.2034
Laser scanner	0.1674	0.1575

$$s_{travelled} = 0.1674\,\text{m} \tag{3}$$

The following table shows the theoretical distance that robot travels during the sample time of each sensor with visualisation turned on and off what is computed the same way as for laser scanner.

6 Conclusions

The experiments and results show that usual devices running on Android operating system with their sensorial equipment and the Android operating system itself is usable in mobile robot navigation tasks. The results show that the experimental mobile robot navigated with visual system travels approximately 24 cm during frame sample obtaining and visualisation process of the camera. However, this distance should increase if more sophisticated image processing algorithms would be used (for example objects and obstacles detection), but on the other side a normal camera can sense larger distances (couple of meters) where the obstacles could be recognised. Therefore the distance of about 20–30 cm from the mobile robot could be a safety distance where mobile robot could stop in one sample before an obstacle to avoid collision. Also a relatively high maximal speed of mobile robot could be reduced to meet the safety distance criteria of the camera's sampling time.

In addition to camera a laser scanner can also be used for obstacle detection. The approximately 17 cm travelled distance during a sample time is also enough to stop the mobile robot in front of an obstacle, because the measuring range of the laser scanner is 4095 mm. The travelled distance of other sensors during their sampling time is in range between 7 and 10 cm. These sensors can be used for example on dangerous tilts detection, collision detection, azimuth computation, etc.

From the point of view of the defined criterion in Sect. 3.1 all the sensors except the camera meet the condition for the suitability of mobile devices for real-time mobile robot navigation. It means that ordinary devices with Android OS can be used for real-time robot navigation tasks with the mentioned sensors and also the camera and image processing algorithms can be implemented to assist during the navigation process in an different application thread. The change in sampling time of the camera by implementation of different image processing algorithms will not rapidly affect the sampling time of other sensors.

Acknowledgments This work was supported by MS SR under contract APVV-0539-11, KEGA 003STU-4/2014 and APVV-0504-12.

References

1. Burian, F., Kocmanová, P., Žalud, L.: Robot mapping with range camera, CCD cameras and thermal imagers. In: Proceedings of 19th International Conference on Methods and Models in Automation and Robotics, MMAR 2014, pp. 200–205 (2014). ISBN: 978-1-4799-5082-9
2. Lipták, T., Duchoň, F., Kelemenová, T., Puškár, M., Kelemen, M., Kurylo, P., Prada, E.: Analysis of uncertainty of tilt measurement with accelerometer. Applied Mechanics and Materials, vol. 611, pp. 548–556 (2014). ISSN: 1660-9336
3. Sukop, M., Hajduk, M., Vagaš, M.: Transferring image processing algorithms on application with multi arm robot and mobile phone. In: Robotics in Alpe-Adria-Danube Region: 23rd International Conference: September 3–5, 2014, Smolenice Castle, Slovakia, pp. 1–6. STU, Bratislava (2014). ISBN: 978-80-227-4219-1
4. Židek, K., Dovica, M., Líška, O.: Angle Measuring by Mems Accelerometers. J. Autom. Mobile Robot. Intell. Syst. **6**(4), 3–6 (2012). ISSN: 1897-8649. http://www.jamris.org/04_2012/saveas.php?QUEST=JAMRIS_No04_2012_P_03-06.pdf
5. Maia, C., Nogueira, L.M., Pinho, L.M.: Evaluating android OS for embedded real-time systems. In: 6th International Workshop on Operating Systems Platforms for Embedded Real-Time Applications (OSPERT 2010), pp. 63–70. Brussels, Belgium (2010)
6. Miková, Ľ., Kelemen, M., Koniar, D.: Mathematical model of four wheeled mobile robot and its experimental verification. Appl. Mech. Mater. **611**, 130–136 (2014). ISSN: 1660-9336
7. Bradski, G., Kaehler, A.: Learning OpenCV, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472 (2008). ISBN: 978-0-596-51613-0

8. Hargaš, L., Hrianka, M., Koniar, D.: Image processing and analysis: a practical approach. Text book. University of Žilina, Žilina (2008). 1 elektronický optický disk (CD-ROM). ISBN: 978-80-8070-962-4
9. Hokuyo Automatic Co., Ltd.: Scanning laser range finder URG-04LX-UG01 (Simple-URG) specifications. Osaka, Japan (2009)
10. Google Inc.: Android developers—sensors overview. http://developer.android.com/guide/topics/sensors/sensors_overview.html (2007)
11. GSMArena, Arena Com Ltd.: Samsung I9100 Galaxy S II. http://www.gsmarena.com/samsung_i9100_galaxy_s_ii-3621.php (2006)
12. GSMArena, Arena Com Ltd.: Asus Google Nexus 7 (2013). http://www.gsmarena.com/asus_google_nexus_7_(2013)-5600.php (2006)
13. GSMArena, Arena Com Ltd.: Sony Xperia Z3 Compact. http://www.gsmarena.com/sony_xperia_z3_compact-6538.php (2006)
14. Google Inc.: Android Developers—SensorManager. http://developer.android.com/reference/android/hardware/SensorManager.html#getRotationMatrix(float[],float[],float[],float[]) (2007)
15. Kuo, M.M., Lee, B.H.: Real-Time Digital Signal Processing: Implementations, Applications, and Experiments with the Tms320c55x. John Wiley and Sons Inc., New York (2001). ISBN 0470841370

Implementation and Control of a Bistable Two-Wheeled Inverted Pendulum-Type Mobile Robot

Kwangyik Jung, Byung Kook Kim and Hyun Myung

Abstract In the past decade, there have been a lot of researches on various models and controllers for two-wheeled inverted pendulum-type mobile robots. Two-wheeled inverted pendulum-type mobile robots are suitable for the service robots working both in the indoor and outdoor environments. In this paper, a bistable two-wheeled mobile robot which can tilt up and down (B-TMR) is designed and implemented. The B-TMR remedies the weak points of two-wheeled inverted pendulum-type mobile robots. The robot body is designed to maintain balancing state and tilted state. Cost efficiency is improved by using a single accelerometer with a Butterworth filter in order to measure inclined angle. To control five motion modes (Mode 1—Balancing; Mode 2—Tilt Up; Mode 3—Tilt Down; Mode 4—Tilted Move; Mode 5 - Balancing Move), a mode-selective controller is proposed. Through various experiments, the proposed hardware and controller are validated.

Keywords Inverted pendulum · Accelerometer · Butterworth filter · PID control

1 Introduction

A service robot is defined as a robot which operates automatically to provide useful services to humans and facilities, excluding manufacturing operations. According to the World Robotics 2013: Service Robots [1] published by IFR (International

K. Jung (✉) · H. Myung
Department of Civil and Environmental Engineering, KAIST,
Daejeon 305-701, Korea
e-mail: ankh88324@kaist.ac.kr

H. Myung
e-mail: hmyung@kaist.ac.kr

B.K. Kim
Department of Electrical Engineering, KAIST, Daejeon 305-701, Korea
e-mail: bkkim@ee.kaist.ac.kr

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_14

Federation of Robotics), the global robotics market grew from US$4.4 billion in 2003 to US$9.4 billion in 2010. The service robot's market share has gone up from 14.3 % (US$63.4 million) in 2003 to 39.3 % (US$369.6 million) in 2010. They expected that the scale of the service robot's market grows fairly within the next 20 years.

As a service robot which is closely related with daily life, the two-wheeled inverted pendulum-type mobile robot has many advantages. Because the control of the two-wheeled inverted pendulum-type mobile robot is much easier than that of the legged robot, it can be adapted to various purposes. The robot can spin in place using two wheels and drive stably at the ramp or in an uneven bumpy ground. Also, the two-wheeled inverted pendulum-type mobile robot is suitable for indoor environments because it can pass narrow corridor and corner without hesitation.

The general two-wheeled inverted pendulum-type mobile robot simultaneously drives and steers using only two motors. Because of that it persistently consumes energy to maintain tilt angle. There are many studies about how to improve weak points by learning the robot's body or changing hardware. A method uses four different sized wheels to keep sitting status and pulls the wheels to be in balancing status [2]. The other method attaches the two additional arms for rest [3].

The research direction for the control of the two-wheeled inverted pendulum-type mobile robot is divided into linear and nonlinear controllers. The study began in 1965 and lots of methods are proposed such as a bang-bang controller, an LQR controller [4], a 3DOF dynamic equation-based controller [5], and so on.

In this paper, we propose a novel two-wheeled inverted pendulum-type mobile robot called 'B-TMR' which is cost effective and energy efficient. The robot body is designed to be 23° inclined underside and only one accelerometer is used to calculate tilt angle. Due to the B-TMR's unique hardware, it has 2 states: tilted state and balancing state. A mode-selective controller is designed to change states and a variety of experiments are performed to evaluate overall systems and controller.

The remainder of this paper is organized as follows: Section 2 explains details on how to design the robot's hardware. In Sect. 3, according to the shape of robot, the robot's movements are separated and a novel controller is suggested to take each action. Experimental results are described to confirm our method in Sect. 4 and conclusion and future work are discussed in Sect. 5.

2 Design of the B-TMR Hardware

2.1 Robot Body

The B-TMR is a more practical and simple robot compared with the existing shapes [2, 3, 6] which are proposed to reduce the power consumption while the two-wheeled inverted pendulum-type robot maintains balancing. Robot body which has a tilted state is designed to minimize a battery loss and maximize the robot speed. Because the 23° inclined underneath robot frame is made by 3D printer, it is light and sturdy. Figure 1 represents a real hardware and Fig. 2 shows details of the structure.

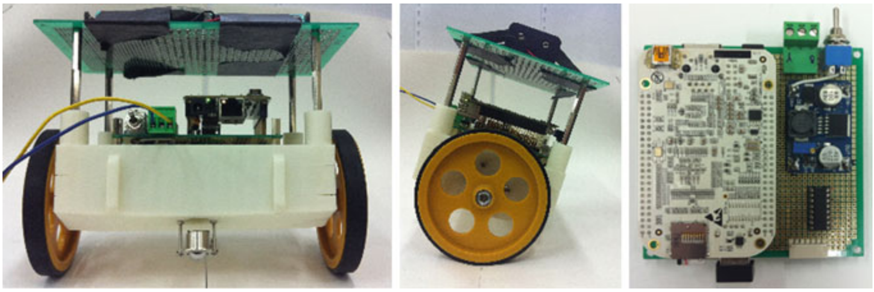


Fig. 1 B-TMR actual hardware and developed interface board

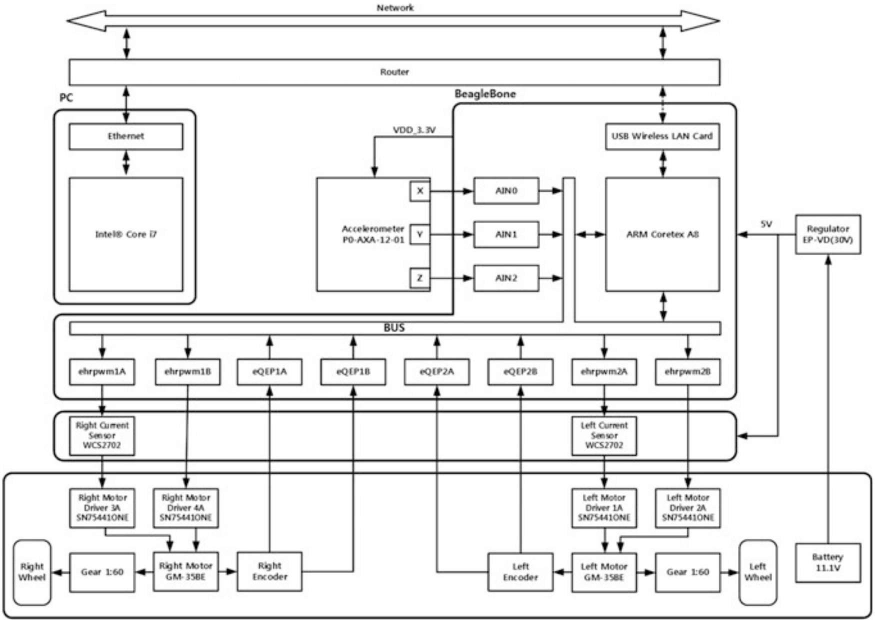


Fig. 2 B-TMR overall structure

2.2 *Embedded Board BeagleBone*

A BeagleBone which has Texas Instrument's AM3359 microcontroller unit is used for B-TMR's brain. The ARM processor is suitable for small systems like low power-based mobile robots and it is easy to develop and debug. The BeagleBone is a credit-card-sized Linux computer and it is an open source type board. The detailed specifications are shown in Table 1. To make own device driver and user application program in cross-development environment, lots of MCU registers are

Table 1 BeagleBone specification

	Feature
Processor	AM3359 500 MHz USB-powered 700 MHz DC powered
Memory	256 DDR2 400 MHz
Ethernet	10/100 RJ45
SD/MMC connector	Micro SD, 3.3 V

directly approached. And Wireless LAN card is used for communication between PC and BeagleBone by considering B-TMR's wide maneuverability.

2.3 Motor Driver and Accelerometer

For this paper own interface board was developed with lots of components. Motor driver SN75441ONE which has a built-in H-bridge circuit is used in order to supply

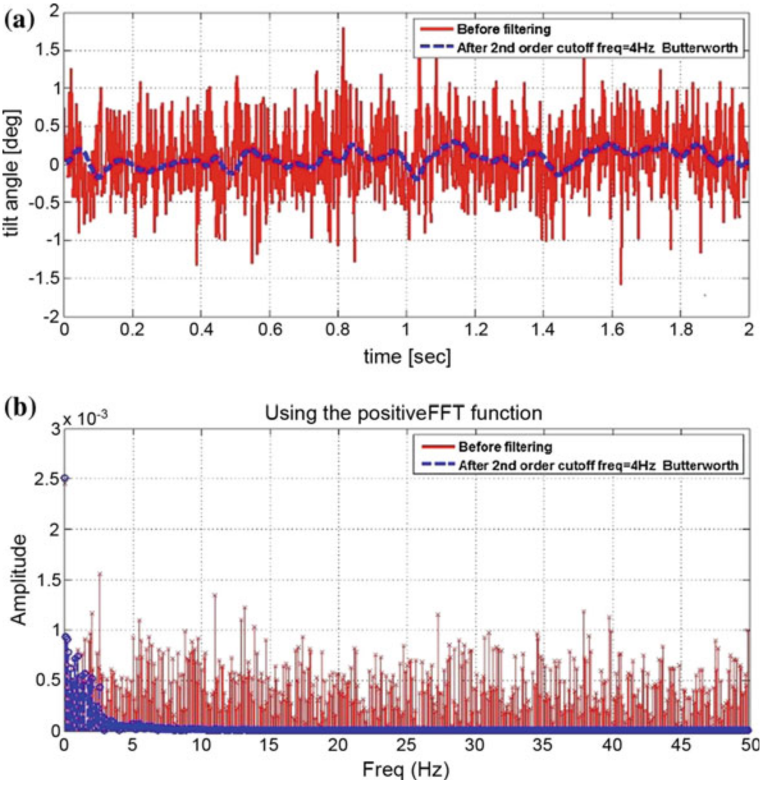


Fig. 3 Butterworth filtering results at 0° (**a**). Fast Fourier Transform about raw and filtered data (**b**)

Table 2 Filtering result analysis

	Min [rad]	Max [rad]	Mean [rad]	Median [rad]	Std [rad]
Raw data	−0.02769 (−1.5814°)	0.0316 (1.8105°)	0.001225 (0.0702°)	0.0009 (0.0516°)	0.008402 (0.4814°)
Filtered data	−0.0033 (−0.1891°)	0.0052 (0.2979°)	0.001256 (0.0720°)	0.00125 (0.0718°)	0.001855 (0.1063°)

PWM to right-and-left GM-35BE motors and a 3-axis accelerometer P0-AXA-12-01 is chosen to calculate the robot's tilt angle. While the existing two-wheeled inverted pendulum-type robots use expensive sensors, in this paper a single accelerometer is used with a Butterworth filter [7] to improve the cost efficiency. We observe the impulse noise at the accelerometer output signals and confirm that high-frequency signals should be removed. Because the second-order Butterworth filter has an even surface at pass-band and a slope of 20 dB/decade degrees at each pole, it can remove above the cutoff frequency signals exactly. The filtering results are represented at Fig. 3 and Table 2.

In comparison with the sensor filtering results of [3], we confirm that using an accelerometer with the 4 Hz cutoff frequency second-order Butterworth filtering is sufficient to get useful data.

3 Mode-Selective Controller and Gain Tuning

3.1 Mode-Selective Controller to Control Five Motion Modes

According to the B-TMR's hardware features, a new method for status changing is needed between balancing state and tilted state. First of all, available actions are broken down into five modes (Mode 1—Balancing; Mode 2—Tilt Up; Mode 3—Tilt Down; Mode 4—Tilted Move; Mode 5—Balancing Move). Every action is performed in a single program and each mode has own reference values and control gains. The mode-selective controller is composed of three subcontrollers (Fig. 4).

3.2 PID Gain Tuning

There are many methods for tuning PID controller's gain. In this paper, the Ziegler–Nichols tuning method [8] is used. Determining the PID gain according to a simple formula is the Ziegler–Nichols method's storing point. Table 3 shows the PID gains for five motion modes.

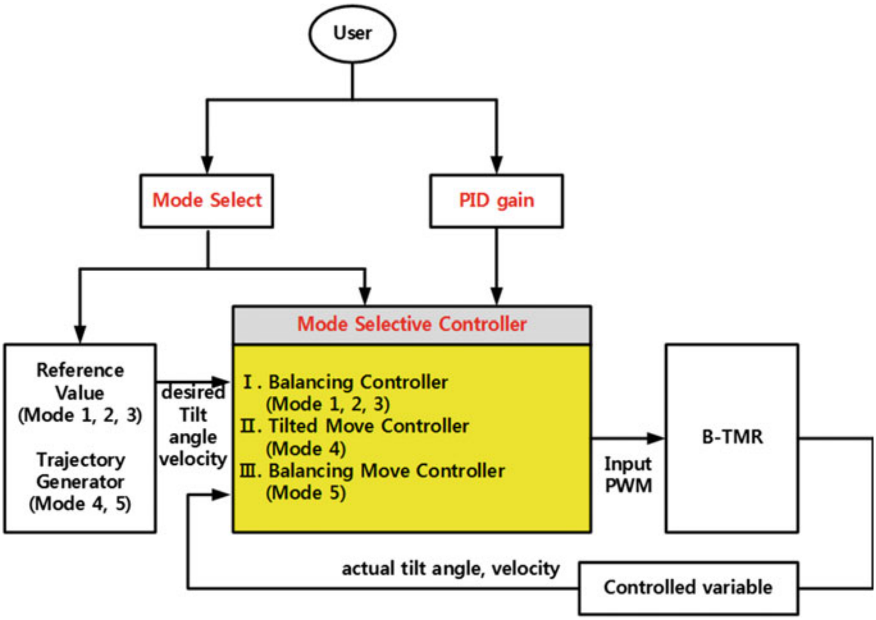

User
Mode Select
PID gain
Mode Selective Controller
Reference Value (Mode 1, 2, 3)
Trajectory Generator (Mode 4, 5)
desired Tilt angle velocity
I. Balancing Controller (Mode 1, 2, 3)
II. Tilted Move Controller (Mode 4)
III. Balancing Move Controller (Mode 5)
Input PWM
B-TMR
actual tilt angle, velocity
Controlled variable

Fig. 4 Mode-selective controller

Table 3 PID gains for experiments

Mode	K_P	K_I	K_D
1—Balancing	Tilt angle = 1.515	Tilt angle = 0.75	Tilt angle = 0.00075
2—Tilt Up	Tilt angle = 2.35	Tilt angle = 0	Tilt angle = 0
3—Tilt Down	Tilt angle = 2.35	Tilt angle = 0	Tilt angle = 0
4—Tilted Move	Right motor = 14 Left motor = 15	Right motor = 2 Left motor = 2	Right motor = 0 Left motor = 0
5—Balancing Move	Tilt angle = 1.515 Linear velocity = 14.5 Angular velocity = 38	Tilt angle = 0.75 Linear velocity = 2 Angular velocity = 12	Tilt angle = 0.00075 Linear velocity = 0 Angular velocity = 0

4 Experiments

4.1 *Mode 1—Balancing*

In Mode 1—Balancing, the B-TMR should retain zero tilt angle. The balancing controller carries out PID control about tilt angle and Fig. 5 shows the actual experimental tilt angle data. The line shows the accelerometer's raw data and the

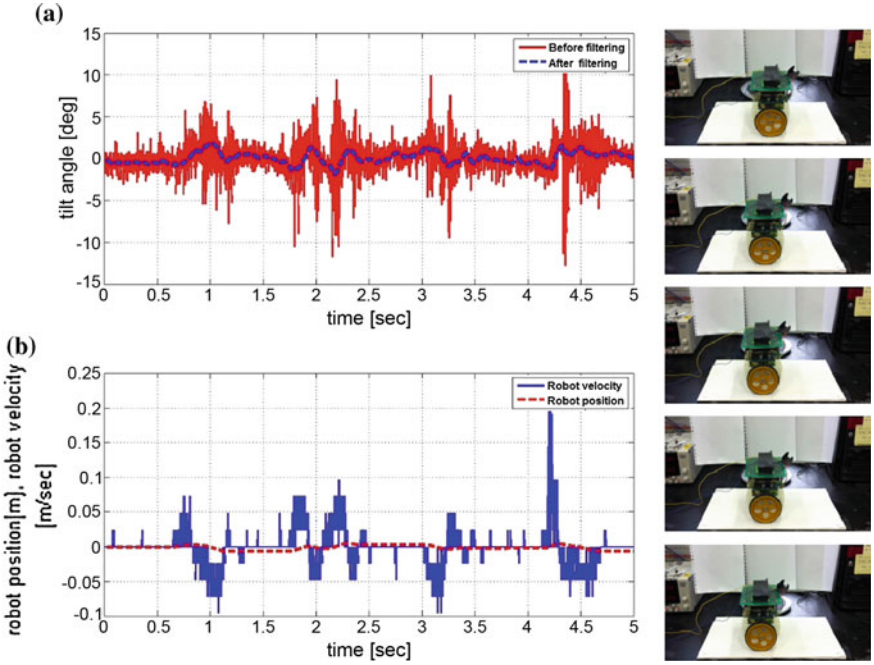


Fig. 5 Balancing tilt angle (**a**), robot velocity and position (**b**)

dotted line represents the Butterworth filtering results. While the robot maintains balancing, it keeps the initial position. Range values are smaller than the experimental data from the compared paper.

4.1.1 Stop Balancing

One of the B-TMR's objective is supposed to reduce the energy consumption while the B-TMR maintains balancing. To calculate two motors' power consumption, current sensors are connected between a motor driver and a motor. The average amount of power is 0.48 W at each motor in balancing state. From the data, we ascertain that the B-TMR cuts down 0.48 W in tilted state without moving (Fig. 6 and Table 4).

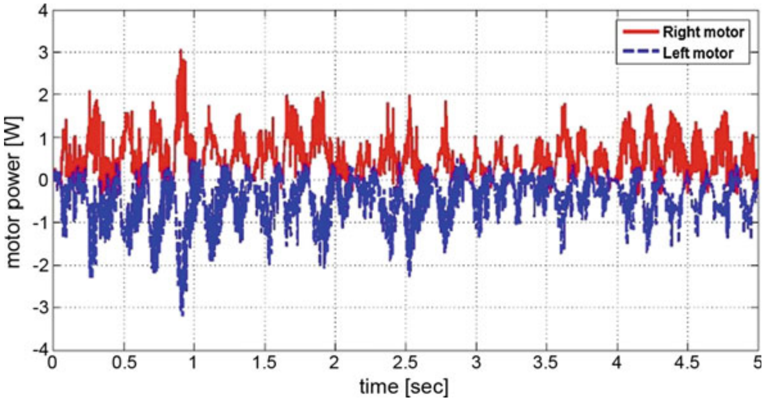


Fig. 6 Power consumption in *right* and *left* motor

Table 4 Comparison of experimental data

Regulated under	Comparison 1 [9]	Comparison 2 [3]	Experiment
Tilt angle	1.1°	3°	1.914°
Position	0.04 m	0.35 m	0.0066 m

4.2 Mode 2—Tilt up and Mode 3—Tilt Down

From the initial-tilted stop status, the B-TMR uses tilt angle's PID control to make the robot body stand. After erecting, the robot maintains balancing. Even though tilt angle oscillates about 15° at first, instantly attains stable balancing state (Fig. 7).

From the balancing initial status, the B-TMR uses tilt angle's PID control for direction selective tilt down. From the Fig. 8, the B-TMR maintains balancing before 1.2 s and the robot performs tilt down for falling a rear ball caster to the ground (minus tilt angle). After 1.2 s, the robot continues uniform motion in tilted state.

4.3 Mode 4—Tilted Move

We make linear and angular velocity profiles to make the B-TMR drive as fast as possible in 90° corner in tilted state. According to the B-TMR's hardware, the robot can use maximum speed of motor without considering Balancing. Figure 9 shows the experimental results.

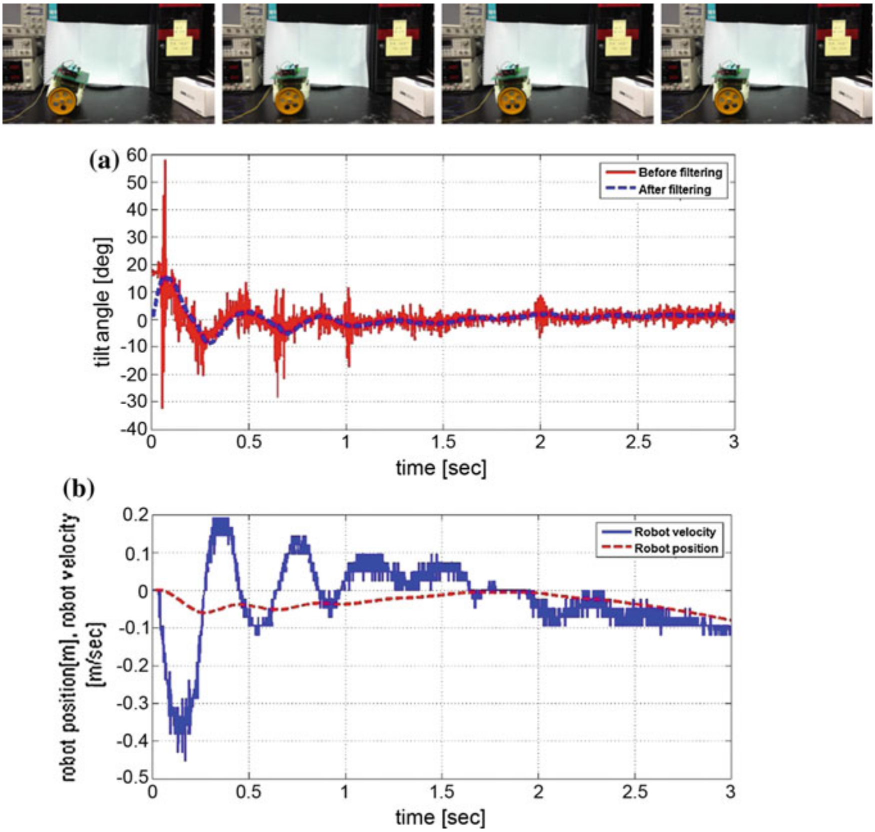


Fig. 7 Tilt Up tilt angle (**a**), robot velocity and position (**b**)

4.4 *Mode 5—Balancing Move*

We produce tilt angle, linear/angular velocity profiles that are considered to simultaneously tilt up and drive. The common two-wheeled inverted pendulum-type mobile robot inclined its body to the opposite direction from driving way for acceleration. However, the B-TMR does not need preprocessing for acceleration, it can drive fast in an uncomplicated way. In other words, the B-TMR stands up before driving from the initial tilted state and maintains ZMP dynamic stability while balancing move. After the drive, the robot returns to tilted state by tilt down controller (Fig. 10).

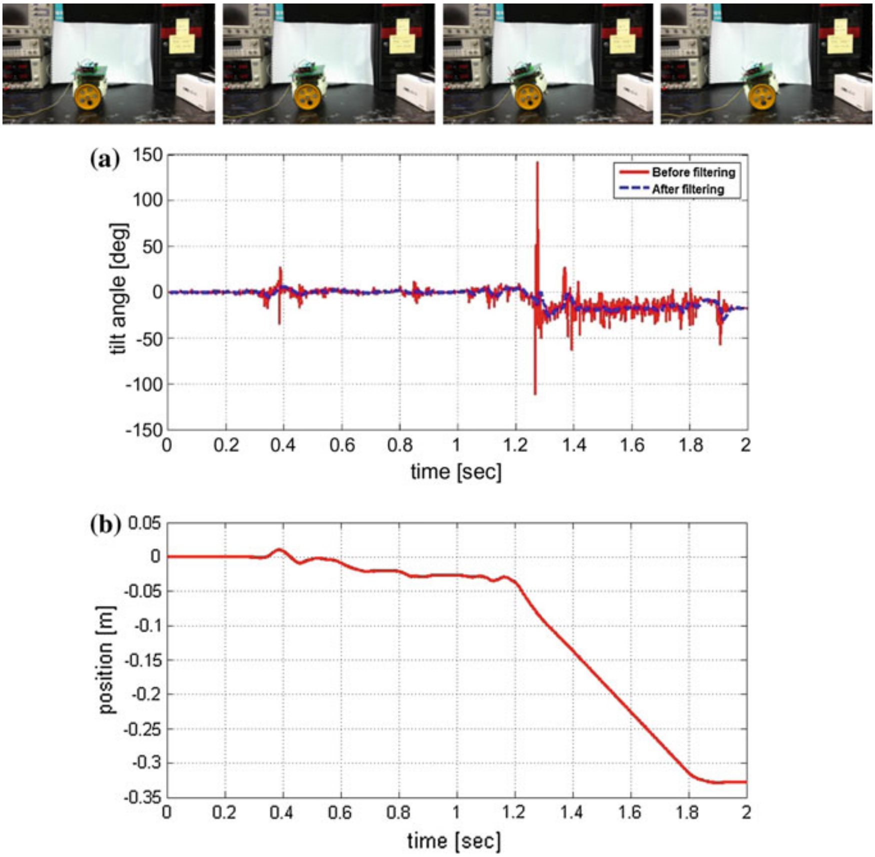


Fig. 8 Tilt Down tilt angle (**a**), robot position data (**b**)

5 Conclusion and Future Work

We proposed a new type of hardware to overcome existing two-wheeled inverted pendulum-type mobile robot's weak points like continuing energy loss while maintaining balancing, limitation of two wheels' velocity, and too much expensive sensors. The B-TMR's body considered the tilted state to save energy and fast move. In spite of a single accelerometer with the Butterworth filter, we got high-quality tilt angle data in comparison with the previous research. According to

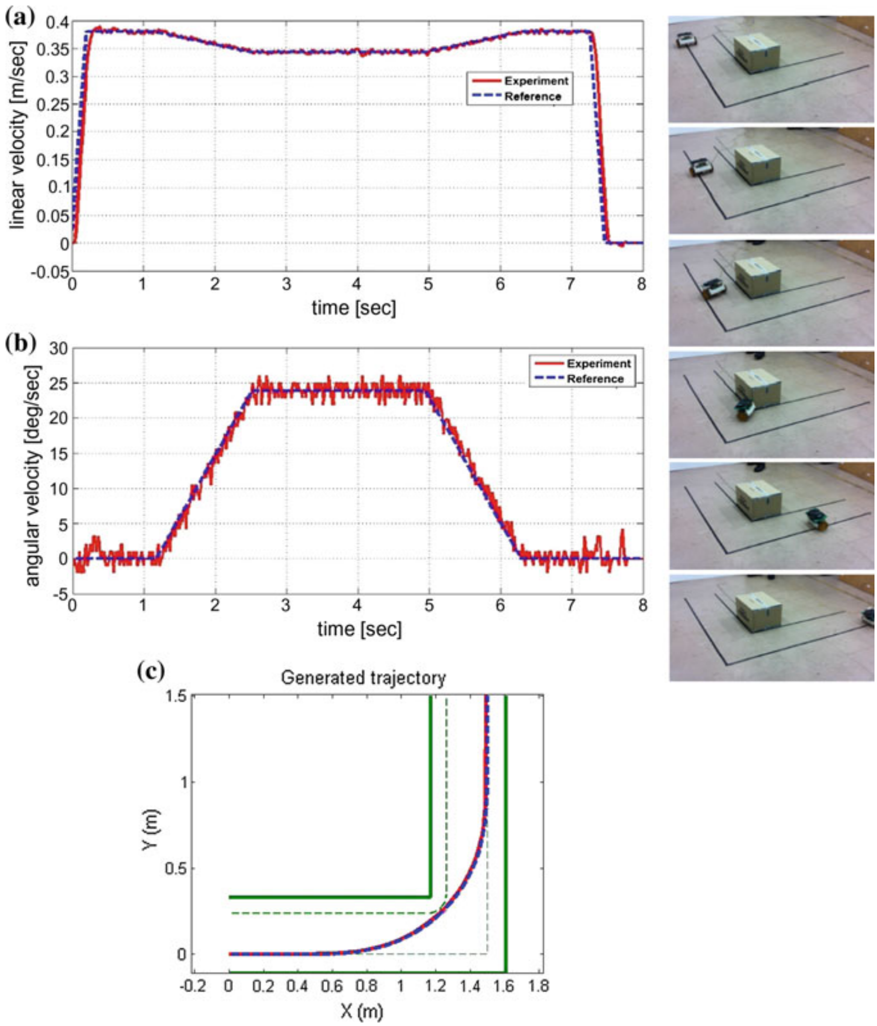


Fig. 9 Tilted move linear velocity (**a**), angular velocity (**b**), overall trajectory without collision at 90° corner (**c**)

the B-TMR's hardware features, the mode-selective controller was developed and various experiments were conducted to validate overall systems. The result of experiments showed that the robot followed reference values or trajectories properly, the mode-selective controller was in control of the robot hardware.

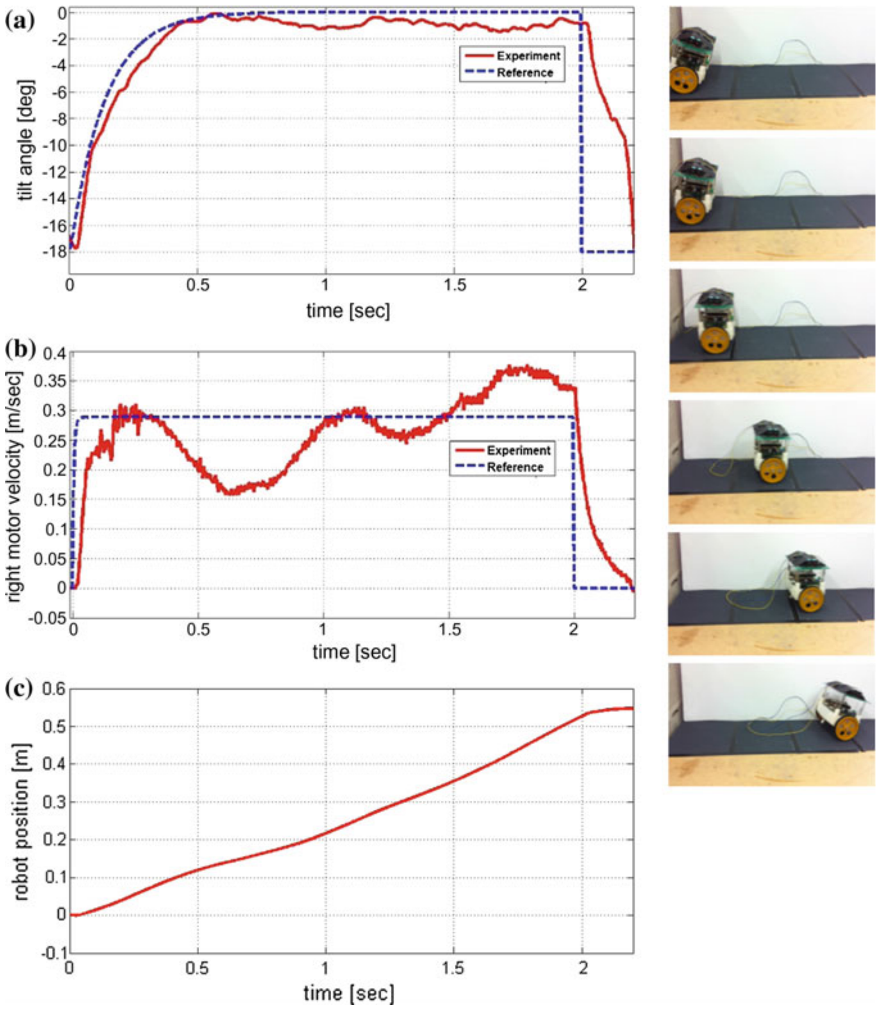


Fig. 10 Balancing Move tilt angle (**a**), robot linear velocity (**b**), accumulated moving distance (**c**)

For future works, we will improve the control method. In this work, we just used PID controller, but the two-wheeled inverted pendulum-type mobile robot has very narrow linearization area about tilt angle. Because of the reason, nonlinear controller is much more proper than linear controller.

Acknowledgments The student was supported by Ministry of Land, Infrastructure and Transport (MOLIT) as U-City Master and Doctor Course Grant Program.

References

1. IFR (International Federation of Robotics): World Robotics 2013 Service Robots (2014)
2. Jeong, S.H., Takahashi, T.: Stable and quick standing-sitting motion of I-PENTAR by whole-body motion with force control. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 199–204 (2008)
3. Feng, T., Liu, T.,Wang, X., Xu, Z., Zhang, M., Han, S.-C.: Modeling and implementation of two-wheel self-balancing robot equipped with supporting arms. In: 6th IEEE Conference on Industrial Electronics and Applications. pp. 713–718 (2011)
4. Kim, Y., Kim, S.H., Kwak, Y.K.: Dynamic analysis of a nonholonomic two-wheeled inverted pendulum robot. J. Intell. Robot. Syst. 25–46 (2005)
5. Ha, Y.S., Yuta, S.: Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot. Robot. Auton. Syst. 65–80 (1996)
6. Miao, S., Cao, Q.: Modeling of self-tilt-up motion for a two-wheeled inverted pendulum. Ind. Robot 76–85 (2011)
7. Sedra, A.S., Brackett, P.O.: Filter Theory and Design: active and Passive.Matrix Publisher (1978)
8. Valério, D., da Costa, J.S.: Tuning of fractional PID controllers with Ziegler–Nichols-type rules. Sig. Process. 2771–2784 (2006)
9. Thao, N.G.M., Nghia, D.H., Phuc, N.H.: A pid back stepping controller for two-wheeled self-balancing robot. Int. Forum Strateg. Technol. 76–81 (2010)

Autonomous Control of a Drone in the Context of Situated Robotics

Sofia Fasce, Diego E. Avila, Emiliano Lorusso, Gustavo Pereira and Jorge Ierache

Abstract This article shows the advances on previous outcomes with a quadcopter in the context of Situated Robotics. Thus, external vision is used to capture the performance environment made up by four checkpoints over which the drone must autonomously fly. The implementation is discussed and the outcomes of the trials performed with the drone guided through two testing models—one external vision-centered and the other one hybrid—are shown.

1 Introduction

The use of air vehicles for different tasks is of utmost importance for Robotics and also for Autonomous Robotics. The number of parameters and variables to take into consideration when moving through the environment is bigger than that of any other type of vehicle, as well as its freedom degrees. A situated robotics environment is proposed to be developed using a drone from the Parrot company, called AR.Drone2.0 [1], on the initial basis of previous works in which a minidrone [2] was used. The AR.Drone2.0 has the necessary tools to develop and experiment as well as for improvements in hardware compared with the previously used drone [2]. Within the parameters offered by the drone, the following can be found: pitch angles, roll and yaw angles, altitude and speed on the "x" and "y" axes. The development of the performed trials was done on language C++, using the SDK provided by Parrot [3] to communicate with the robot, OpenCV [4] and Linux environment (Ubuntu 12.04).

S. Fasce · D.E. Avila · E. Lorusso · G. Pereira · J. Ierache (✉)
Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica,
Facultad de Informática Ciencias de la Comunicación y Técnicas Especiales,
Universidad de Morón, Buenos Aires, Argentina
e-mail: jierache@yahoo.com.ar

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_15

2 Problem

In the field of situated robotics, different applications and competences can be found, such as robot football which belongs to physical categories which perceive the field environment from a suspended camera, for example, RoboCup F-180 category [5]. This paper proposes the creation of a situated robotics environment using air vehicles, taking into account their autonomous control from environment information obtained from the suspended camera and the use of different sensors available in the robot. Our aim was to study the flight dynamics of a quadcopter in situated robotics using the PID controller (proportional–integral–derivative) algorithm and to analyze the possible options for improving its flight.

3 Proposed Solution

In this work, the initial experience with the vision algorithm developed in [2] was taken; a suspended camera (Logitech C270, 3 Mpx, PAL) was used to determine robot position and destination point or checkpoint. Moreover, an altitude control was added so that the robot could fly at a constant height with necessary adaptations and improvements for robot control using SDK to communicate with the AR. Drone2.0 shown in Fig. 1. The red circle over the robot was used to identify it during video frame analysis and allow its position detection.

During flight, the angle and distance the robot must cover to fly over the checkpoint and stay suspended in the air for two seconds was estimated. Once these variables are calculated, the tilt and roll parameters are adapted using the PID

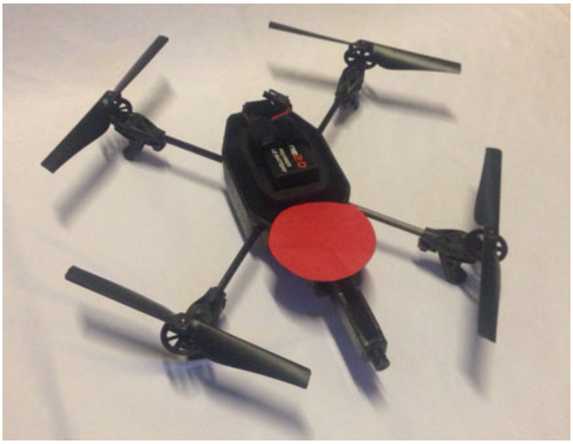


Fig. 1 ARDrone2.0

algorithm. We apply PID [6] to both robot pitch and roll to precisely determine the angle and speed when reaching checkpoint. In our case, we achieved an optimal flight using the following values for each gain: for proportional gain (**Kp**), the value 0.0005; for integral gain (**Ki**), the value 0.000000076; and for derivative gain (**Kd**), the value 0.000425, which corresponds to the 85 % of the proportional gain value. These values were empirically obtained so that they were adjusted with each test. Eighty flights were done to obtain these outcomes. The value optimization allows tilt reductions when reaching the checkpoints and arrival time reductions. The algorithm estimates the error based on the real value of the variable to calculate, that is, the difference between the desired and obtained value. Then, it estimates the values for the derivative, proportional, and integral variables

```
change = error – last_error
P_value = Kp * error
I_value = Integrator * Ki
D_value = Kd * change
```

Lastly, it stores the last error, sums the integrator, and the sum among the different values previously calculated

```
Integrator = Integrator + error
last_error = error
return P_value + I_value + D_value
```

These values are estimated for each processing cycle which corresponds to each video frame (13 fps—frames per second). Thus, and since robot speed is calculated from the video analysis, delay occurs, which should be considered in flight dynamics, making the drone tilt according to video frames from the drone's flight. This delay is included in the algorithm mainly when calculating the derivative gain. This derivative is especially important since it correlates to robot speed and allows the robot to stop at the checkpoint without oscillations. To compare drone behavior to vision-estimated speed, called vision-centered model, we decided to perform the same tests using the speed provided by the robot and not calculated through video analysis, forming a model we call hybrid, since these data do not include delay from video frame and have a much higher precision. Alternatively, a standardization of the resulting values for the integral gain was made, allowing robot speed maximization in order to reach the checkpoint as soon as possible. The method `integratorIncrease` allows the standardization of the `Integrator` attribute outcome so as not to exceed the set maximum and minimum values. The following shows the implementation of the mentioned method in pseudocode:

```
if distance < 0 and Integrator > 0:
     Integrator = max(0, Integrator + 14 * distance)
else if distance > 0 and Integrator < 0:
     Integrator = min(0, Integrator + 14 * distance)
else:
    Integrator += distance
endif
if Integrator > Integrator_max:
     Integrator = Integrator_max
else if Integrator < Integrator_min:
     Integrator = Integrator_min
Endif
```

The `distance` variable shows the distance the robot needs to cover to reach destination. The `Integrator` variable is the algorithm integral component, and it is the one to be increased by the `integratorIncrease` function. This function does not modify algorithm behavior and thus, no changes were made to it. The software developed is in charge of controlling the drone and analyzing the video, creates two execution strings which share part of the memory, making it possible to send messages between them. Each has a different behavior: the first deals with suspended camera video analysis, and the second interprets data obtained from the first string and sends necessary signals to the drone to reach checkpoint. In Fig. 2 a general diagram of the developed software is shown. Both execution strings, representing Video Analyzer and ARDrone Control, and data shared, through which information is sent, can be observed. It can also be seen how these strings accordingly communicate with the video camera, the graphic interface and the drone, and a block diagram explaining the general software functioning.

Alternatively, an interface was developed to visualize the camera video analysis and the resulting values after applying PID algorithm to the robot's pitch and roll variables. Figure 3 shows the suspended camera video analysis carried out by the developed software. (a) shows the values sent to the robot. In (b) the values for roll, pitch, yaw, altitude, and desired rotation angle can be observed. In (c) the values for speed in "x", "y", and "z" are shown. The square drawn (d) represents the camera coverage area according to the robot flight altitude. Each cross drawn (e) represents the identified checkpoints. Lastly (f) represents the robot and its current position.

Figure 4 shows the graphs of the values obtained using the PID algorithm. These are divided into three groups: pitch, roll, and yaw values. The three variable groups are observed. The first graph shows the values for proportional gain (red), integral grain (green), and derivative gain (blue) for rolling. The second graph displays

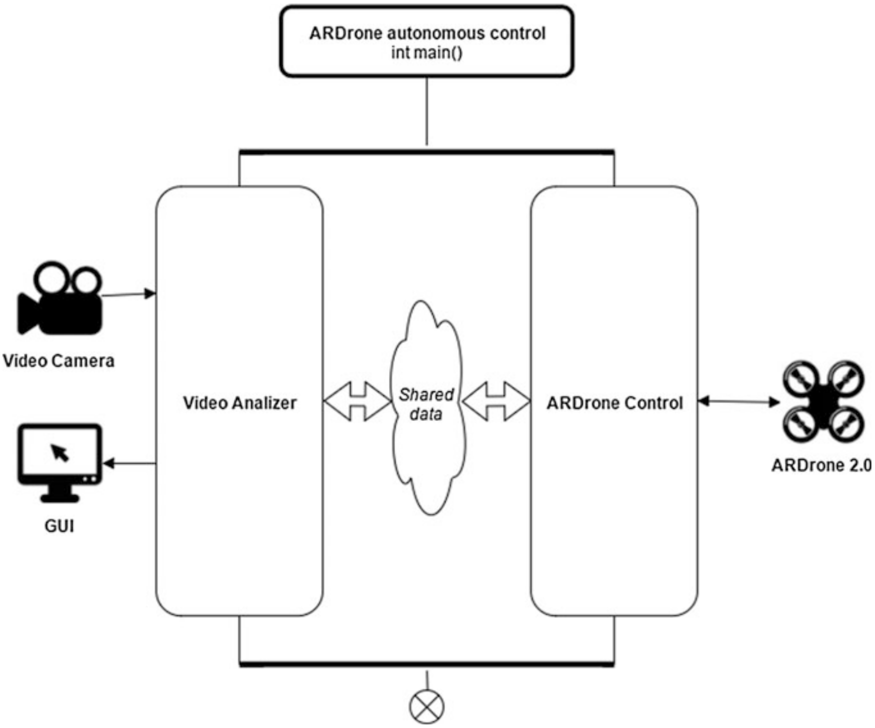
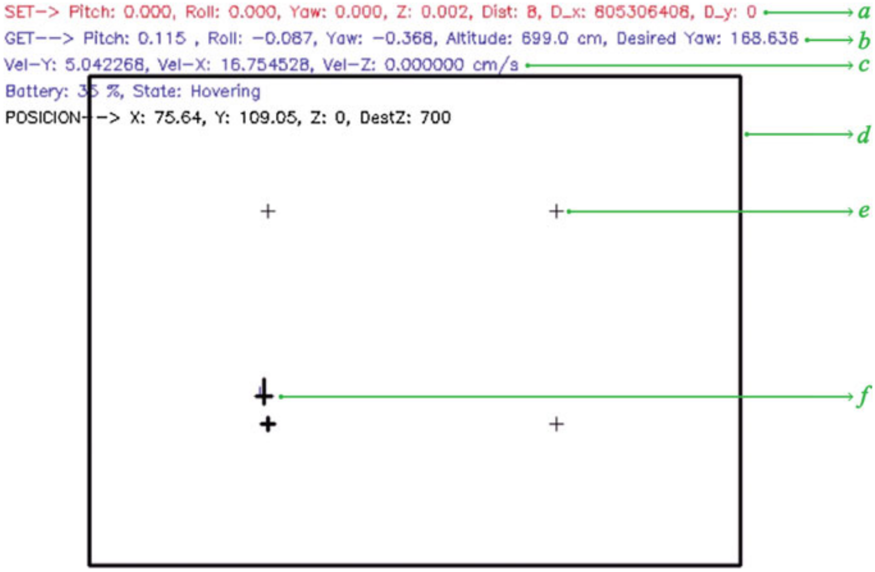


Fig. 2 General diagram of the developed software



```
SET-> Pitch: 0.000, Roll: 0.000, Yaw: 0.000, Z: 0.002, Dist: 8, D_x: 805306408, D_y: 0
GET--> Pitch: 0.115 , Roll: -0.087, Yaw: -0.368, Altitude: 699.0 cm, Desired Yaw: 168.636
Vel-Y: 5.042268, Vel-X: 16.754528, Vel-Z: 0.000000 cm/s
Battery: 35 %, State: Hovering
POSICION--> X: 75.64, Y: 109.05, Z: 0, DestZ: 700
```

Fig. 3 Analysis of the video from the suspended camera

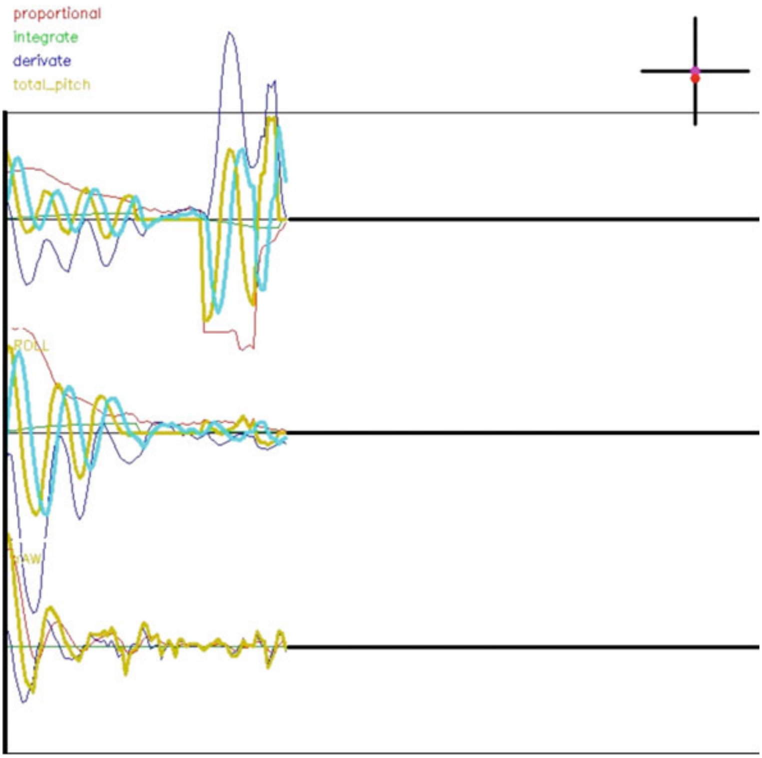


Fig. 4 Graphs of the values obtained by using the PID algorithm

pitch. The third graph indicates yaw. Lastly, the graph thick lines represent the pitch and roll values sent (yellow) and received (light-blue).

4 Achieved Outcomes

Several tests were performed within a circuit with four color patches used as checkpoints. The robot had to fly over all the checkpoints and stay suspended in the air for 2 s at a 10-cm radius, following the A => B => C => D route. Figure 5 shows the robot following the pertinent route. The drone following the determined route is observed. Figure 5a shows the upper view from the suspended camera; Fig. 5b shows the lateral view. Checkpoint A is represented by the orange patch; checkpoint B is represented by the green patch; checkpoint C is represented by the blue patch; and checkpoint D is represented by the yellow patch. The drone starts from the center of the figure, follows the route and lands over the yellow patch.

Two sets of trials were performed; each of them included four tests. The first set of trials consisted in the basic implementation of the PID algorithm, using robot

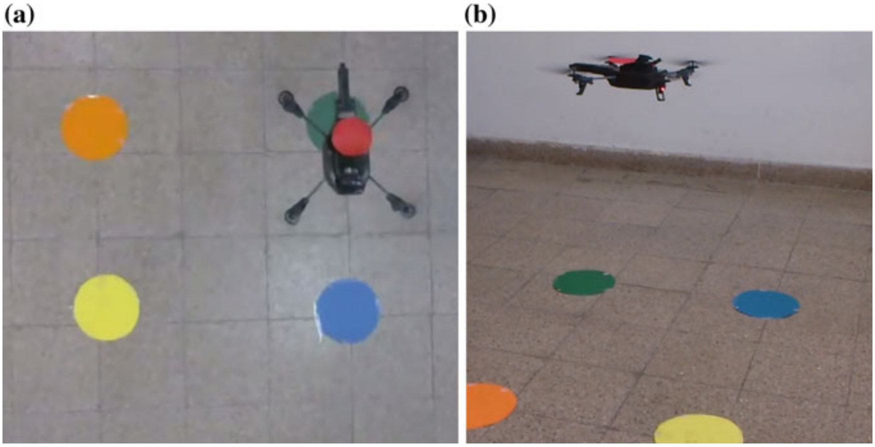


Fig. 5 Robot following the determined route. **a** View from the suspended camera. **b** Lateral view

Table 1 First set of trials for the route when using speed calculations from the video (vision-centered model)

Set of trials	Checkpoint A (s)	Checkpoint B (s)	Checkpoint C (s)	Checkpoint D (s)	Overall (s)
1a	10	7	10.9	10.7	38.6
1b	7.5	8.6	9.4	12	37.5
1c	7.4	6	10.3	10.9	34.6
1d	10.6	7.9	8.9	4.9	32.3
Mean time	8.88	7.38	9.88	9.63	35.75

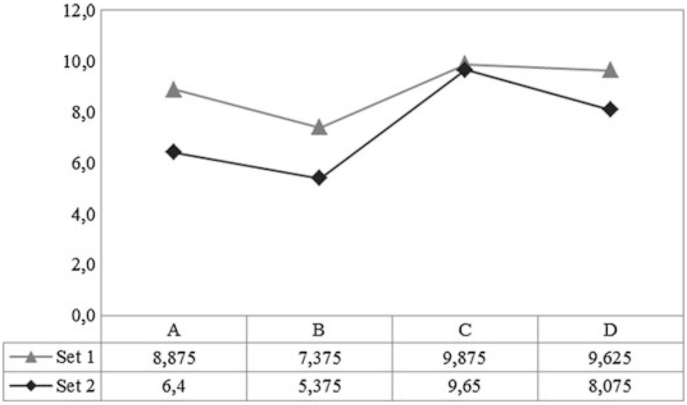
speed from the suspended camera video analysis, which we call vision-centered trial model. Table 1 shows the outcomes of the first set of trials for the route when using speed calculations from the video. Times for each checkpoint, overall time of each route and mean time for each checkpoint are shown.

The second set of trials corresponds to the route followed using the PID algorithm with `integratorIncrease` function and using the speed provided by the robot sensors, which is more precise and does not show considerable delay; this is the model we call hybrid Table 2 shows the outcomes of the second set of trials corresponding to the same route when using speed data from drone sensors. Times for each checkpoint, overall time of each route, and mean time for each checkpoint are shown.

In Fig. 6 appears the comparison between the average for each checkpoint in each set of trials. It shows the comparison between the vision-centered trial model corresponding to the first set of trials based on speed calculations by means of video analysis, and the second set of trials, corresponding to the hybrid model based on using the speed provided by the drone and the position from the vision system. The

Table 2 Second set of trials for the route when using speed data from drone sensors (hybrid model)

Set of trials	Checkpoint A (s)	Checkpoint B (s)	Checkpoint C (s)	Checkpoint D (s)	Overall (s)
2a	5.7	4.6	8.3	4.7	23.3
2b	5.4	8	10.9	10	34.3
2c	7.1	5.3	12.2	11.9	36.5
2d	7.4	3.6	7.2	5.7	23.9
Mean time	6.4	5.38	9.65	8.08	29.5



	A	B	C	D
▲ Set 1	8,875	7,375	9,875	9,625
◆ Set 2	6,4	5,375	9,65	8,075

Fig. 6 Comparison between the average for each checkpoint in each set of trials

second set is noticeably better relating to time needed to reach the different checkpoints. The depicted values represent seconds.

In data analysis, we can observe that the difference between the first set of trials, with an overall mean time of 35.75 s, and the second set of trials, with an overall mean time of 29.5 s is 20 %. We consider that the difference between sets is acceptable given the trial conditions. Video analysis for robot speed calculation, when compared to the data obtained from the quadcopter sensors, does not noticeably affect the vehicle flight. Thus, as the second set of trials shows, an improvement in flight time is achieved by combining data from the suspended camera and speed date from drone sensors.

5 Conclusions

When using air vehicles in situated and autonomous robotics, it is imperative to use (a) a flight dynamic allowing some fluidity when moving through the environment, and (b) the largest quantity of data regarding vehicle and environment status. These two essential characteristics are obtained from both the suspended camera video analysis and the use of all the sensors the robot has.

According to the experimental work, for the development of flight patterns an average difference of twenty percent in the drone's flight duration is shown between a hybrid model when applying the drone's speed and determining its position through the vision system and a vision-centered model, from which not only the position but also the speed are determined.

6 Future Research Lines

The intention here is to venture together with the situated robotics into the intervention of the human being in the control by means of brain–machine interfaces (BMI) [7, 8]. By having this connection, it is possible to control the vehicle through biosignals and perform actions such as choosing the checkpoints to fly over, supervising the robot flight, as well as controlling an independent flight.

References

1. ARDrone 2.0 de Parrot. http://ardrone2.parrot.com/
2. Ávila, D., Lorusso, E., Fasce, S., Ierache, J.: Robótica situada aplicada al control de vehículos aéreos. CACIC 2014. http://hdl.handle.net/10915/42255
3. ARDrone 2.0 SDK. http://developer.parrot.com/ardrone.html
4. OpenCV. http://opencv.org/
5. Robocup Small Size League. http://wiki.robocup.org/wiki/Small_Size_League
6. Zhong, J.: PID Controller Tuning: A Short Tutorial. Purdue University (2006)
7. Ierache, J., Pereira, G., Iriibarren, J.: Navigation Control of a Robot from a remote location via the Internet using Brain-Machine Interface; Robot Intelligence Technology and Applications 2: Series Advances in Intelligent and Soft Computing of Springer, pp 297–310. Springer (2014)
8. Ierache, J., Pereira, G., Iribarren, J., Sattolo, I.: Robot Control on the Basis of Bio-electrical Signals, Robot Intelligence Technology and Applications, Series Advances in Intelligent and Soft Computing of Springer, pp. 337–346, vol. 208 (2013)

Fast *MAV* Control by Control/Status OO-Messages on Shared-Memory Middleware

Dimitri Joukoff, Vladimir Estivill-Castro, René Hexel and Carl Lusty

Abstract We describe how control/status OO-messages on shared-memory middleware can provide better performing control of a micro-air vehicle (*MAV*). To illustrate this, we provide a new hardware abstraction for a controller application that is completely analogous to the popular `ardrone_autonomy` (*AA*) package that enables the Parrot AR Drone 2.0 quadcopter to be flown using commands over Wi-Fi. For fairness of comparison, we use the OO-messages on shared-memory middleware implementation `gusimplewhiteboard` in parallel with the `ROS` *AA* in the same code-base. We demonstrate the performance improvements associated with using `gusimplewhiteboard` messaging in place of `ROS messages` and `services`. We explain how further performance improvements can be achieved by fully implementing the Time Triggered Architecture (*TTA*) of the `gusimplewhiteboard` and its associated tools (`clfsm` & LLFSMs).

Keywords Robotic middleware · `ROS messages` and `services` · PULL versus PUSH Control/status messages · Time triggered architecture

1 Introduction

Software architectures for robotics provide levels of abstraction that enable the construction of complex robotic systems in a modular ensemble of high-level structures. Moreover, there is the need to integrate components that capture and interpret

D. Joukoff · V. Estivill-Castro · R. Hexel · C. Lusty (✉)
Griffith University, Nathan, QLD 4111, Australia
e-mail: c.lusty@griffith.edu.au; carl.lusty@griffithuni.edu.au
URL: http://mipal.net.au

D. Joukoff
e-mail: d.joukoff@griffith.edu.au; Dimitri.Joukoff@griffithuni.edu.au

V. Estivill-Castro
e-mail: v.estivill-castro@griffith.edu.au

R. Hexel
e-mail: r.hexel@griffith.edu.au

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_16

information from several types of sensor as well as modules that integrate this information. Since the hardware itself can integrate different elements and effectors, layers of abstraction are required for higher-level information processing modules and decision making modules so they are decoupled from the specifics of the underlying hardware. One clear illustration of this are micro-air vehicles (*MAV*s), where a very rapid feedback loop control ensures stability in the air. To enable sophisticated behaviour, several layers of abstraction interconnected through middleware allow the creation of manoeuvres (rotate, hover, etc.) that form the building blocks for intelligent capabilities (task-planning, reasoning, decision-making, negotiation) at a higher level of abstraction. Based on this, even higher-level software layers then carry out command and control of whole "missions". The need for this abstraction and structure of the robotics architecture is so ubiquitous that several environments exist to this end [4]. Perhaps the most popular at this stage is `ROS` [12], which has been ranked more comprehensive than others [3]. It is important that each layer shall provide the required interface and abstraction with minimum performance penalty. Unfortunately, all the proposals mentioned above tend to follow the PUSH-approach to message passing that creates significant coupling [6] between the sender module and the receiver module [5]. The PUSH-approach consists of a publisher/subscriber model where the receiver designates a call-back function to the middleware, expecting to handle messages deposited by the publisher. However, the `gusimplewhiteboard` middleware [7] provides an alternative (under the PULL—approach) to facilitate communication between software modules on a single robot as well as to share information between groups of robots [5].

We demonstrate the advantages of the `gusimplewhiteboard` on platforms where timely mission control can be critical. Flying robots such as *MAV*s are a challenging platform because of their need for fast and reliable execution of manoeuvres and mission control. Thus, in this paper, we use a quad-copter to assess and analyse the contrast between these alternatives. We selected the Parrot AR Drone 2.0 as the test platform as it has constituted itself as a popular low cost, ready-to-fly product, with an existing API that allows it to be controlled from a computer [1, 2, 11]. We will contrast the `gusimplewhiteboard` PULL-based approach with the `ROS` PUSH-approach in exactly the same missions. In this context, the `gusimplewhiteboard` outperforms `ROS` by three orders of magnitude when it comes to response time. The remarkable difference in performance between the PULL} and the PUSH alternatives raises the question why the vast majority of practitioners are still opting for the overly optimistic PUSH-approach. Therefore, this paper also offers, a detailed explanation of the methodology used to gather the data together with an analysis of the results. We believe that these successful flights and data demonstrate the benefits of the paradigm represented by the `gusimplewhiteboard`. Since many researchers are familiar with `ROS`, we will explain the relevant steps so that the results here are reproducible.

2 Preliminaries: The gusimplewhiteboard

The gusimplewhiteboard[1] [7], is a middleware based on the principle of the Time-Triggered Architecture (*TTA*) [9]. It is a scalable multi-platform framework that enables lockless inter-process communication (IPC) through shared memory. Usage examples include the Aldebaran Nao platform in different contexts including the RoboCup competition, ROS, and the *Gazebo* simulator [8] for mobile robotic manipulators like the Kuka. Further applications include Lego's Mindstorm NXT for teaching robotics, the Giraff robot for Human/Robot Interaction, and settings where the controller is a Raspberry Pi.

The gusimplewhiteboard is the combination of a set of user defined whiteboard classes that act as communication channels (analogous to ROS:topics) and a static instance, in shared memory, of each of the channel classes in the set. The same data structure can be used for two slots, constituting the status and control channels. To allow lock-free atomic messaging and change tracking, the gusimplewhiteboard is typically configured to store the last four generations of each message in a ring buffer. The configuration of message-slots and sizes here happens at compile time for improved performance. The dimensions of the gusimplewhiteboard are defined such that the shared memory space is able to be cached on many platforms.

Producers and consumers use local instances of whiteboard classes to communicate. When a producer wishes to send a message, they perform a copy operation that updates the static instance of the object in the gusimplewhiteboard with the contents of the producer's local instance. When a consumer needs to retrieve (PULL) a message, the reverse happens, i.e. copy the current instance of the object in the gusimplewhiteboard into the consumer's local memory (if required, an older generation can be requested explicitly). When operating in a distributed system, a UDP-based sharing algorithm that also makes use of the *TTA* is used to broadcast data to other agents. There is no native equivalent to ROS::Services in the gusimplewhiteboard. Instead, synchronisation, the corner stone of the *TTA*, is used to perform the necessary operations. The designer of the system can structure the semantic organisation of the whiteboard classes that are used on the gusimplewhiteboard. Thus, it can service a number of different agents and be used to perform different tasks. Each whiteboard class is independent of the others. Agents can also have more than one, independent gusimplewhiteboard operating at the same time. For example, one gusimplewhiteboard can transact higher-level messages between behaviours while another operates at a hardware level reading sensors and controlling actuators.

As stated above, the gusimplewhiteboard embodies the principles of the *TTA*. Thus, other than the physical limitations of the host system, there is no inherent message rate limit, unlike the PUSH-approach where the receiver subscribes a call-back-function and the sender is a publisher (hence, system resources must be

[1] Available at http://mipal.net.au/downloads.php.

available to enact the call-back and synchronisation mechanisms must be used to handle the threads generated for each invocation of the call-back).

2.1 Whiteboard Classes

At a glance, gusimplewhiteboard messages appear similar to ROS messages. Both, gusimplewhiteboard and ROS transmit a data structure of a conceptual abstraction or entity. Recall that in ROS, the data structure for a message is analogous to a struct that enumerates properties/fields as name and type. However, because the gusimplewhiteboard is object-oriented, messages are objects with methods. As with ROS, the data structure can have two semantic meanings: *control* or *status*. For example, in ROS the AR Drone geometry_msgs/Twist can be posted as the **control** command containing the velocities to aim for, or, in the other direction (not used in ardrone_autonomy), as a **status** message containing the drone's position and altitude measurements. We use the ardrone_autonomy's Cmd_Vel hardware abstraction [10] as an example and show its analogous gusimplewhiteboard implementation and illustrate how the same data structure could be both the payload of a control and a status message. Figures 1 and 2 show data structures analogous to ROS, but defined in C. The gu_util.h header provides the definition for the PROPERTY macro so the getters and setters for C++ are generated. Nesting and containment of data structures is simply nesting of types: Fig. 2 uses the type defined in Fig. 1. While it is not necessary to discriminate between a status and a control message, we added the Boolean property, requestCmd, for data recording in our later experiments.

For actual whiteboard objects, we create subclasses of the C base structs. Thus, the class in Fig. 4 uses the payload from Fig. 3. While ROS requires a ardrone/cmdvel topic for overloading geometry_msgs/Twist, the strong typing of C++ allows association of the ARDrone_Cmd_Vel directly as a status or control message, and only requires that separate shared memory slots be used. All gusimplewhiteboard slots are assigned in a Types List (TSL) file [6] that is

Fig. 1 wb_ardrone_vector_31d defines a simple data structure

```
#ifndef _WBARDRONEVECTOR31D_
#define _WBARDRONEVECTOR31D_

#include <gu_util.h>

struct wb_ardrone_vector_31d {
  PROPERTY(double, x)    /** X Axis */
  PROPERTY(double, y)    /** Y Axis */
  PROPERTY(double, z)    /** Z Axis */
};
#endif // _WBARDRONEVECTOR31D_
```

Fig. 2 An analogous data structure to ROS's geometry_msgs/Twist

```
#ifndef _WBARDRONEGEOMETRYTWIST_
#define _WBARDRONEGEOMETRYTWIST_

#include "wb_ardrone_vector_31d.h"

#include <gu_util.h>

struct wb_ardrone_geometry_twist {
  /** Linear Component*/
  PROPERTY(wb_ardrone_vector_31d, linear)
  /** Angular Component*/
  PROPERTY(wb_ardrone_vector_31d, angular)
};
#endif // _WBARDRONEGEOMETRYTWIST_
```

Fig. 3 wb_ardrone_cmd_vel is the next level nesting base class

```
#ifndef _WBARDRONECMDVEL_
#define _WBARDRONECMDVEL_

#include "wb_ardrone_geometry_twist.h"

#include <gu_util.h>

struct wb_ardrone_cmd_vel {
  PROPERTY(bool,requestCmd)
  PROPERTY(wb_ardrone_geometry_twist, cmd)

};
#endif // _WBARDRONECMDVEL_
```

Fig. 4 The ARDrone_Cmd_Vel class

```
#ifndef ARDrone_Cmd_Vel_DEFINED
#define ARDrone_Cmd_Vel_DEFINED

#include <cstdlib>
#include <stdlib.h>
#include <gu_util.h>
#include "gusimplewhiteboard.h"
#include "wb_ardrone_cmd_vel.h"

namespace guWhiteboard {
  class ARDrone_Cmd_Vel: public wb_ardrone_cmd_vel {
    public:
      ARDrone_Cmd_Vel() : wb_ardrone_cmd_vel() {}
      ARDrone_Cmd_Vel(const std::string &name)
         : wb_ardrone_cmd_vel(std::string(name)) {}
  };
};
#endif // ARDrone_Cmd_Vel_DEFINED
```

pre-processed to provide handler and access methods to the shared memory, producing a standard API for posting and retrieving messages.

3 Design Improvements and the ARDrone_Whiteboard

To compare the PULL with the PUSH-approach in a fast-command control environment, we have implemented a hardware abstraction of the Parrot AR Drone 2.0 completely analogous to the original ROS version. ARDrone_Whiteboard implements the equivalent gusimplewhiteboard classes for all control and status messages provided by ardrone_autonomy, including 'legacy' messages for version 1 as well as their version 1.4 replacements. Messages with very large payloads (e.g. Odometry and IMU), are split into their components to fit within the gusimplewhiteboard width of 512 bytes chosen for ARDrone_Whiteboard. In fact, this replicates most version 1 to version 1.4 ardrone_autonomy performance improvement changes. Note that while ROS offers ROS::Params, which enable some configuration at run-time, the configuration of message-slots and sizes here happens at compile time for improved performance.

Since the gusimplewhiteboard does not provide a blocking handshaking functionality like ROS::Services, the ardrone_autonomy services needed to be redesigned as status messages. That is, in an application on top of ROS that uses ardrone_autonomy, if we want to obtain the value of a sensor, we invoke a ROS::Services and our program will be blocked there until the ROS middleware comes back with an answer. This introduces lots of issues of module coupling and timing already discussed in the literature [6]. The explicit advantage of the PULL-approach is that there are no blocking calls. To retrieve a sensor value, just PULL the most recent status message (that the sensor posts at the sensors publishing rate). Consumers of status messages can inspect the message generation counter if they want to or need to check for updates; however, typically idempotent messages are used. Thus, in ARDrone_Whiteboard no call-backs are used. Instead, the teleop_twist update_teleop() function checks (PULLS) new messages within its run loop. We will demonstrate that this has the effect of greatly reducing the latency between issuing commands and their enactment. This way, we are creating a deterministic, predictable schedule (as opposed to the non-deterministic enacting and resource consuming management of the same code, but encapsulated in a call-back). We will see that this results in a much lower variance than ROS when relaying messages. To ensure that all other aspects of the context remain the same, ARDrone_Whiteboard builds under the same catkin workspace as ardrone_autonomy with an analogous CMakeList.txt, linking to the existing ARDroneLib. This setting allows missions and higher layers of behaviour control on the drone to operate unchanged, and to interchangeably fly with the ROS PUSH approach represented by ardrone_autonomy or the gusimplewhiteboard PULL approach using ARDrone_Whiteboard. The only feature not included is ardrone_autonomy's video functions.

The advantages of PULL can be further illustrated by inspecting the architecture of ardrone_autonomy. For example, the ROS ardrone/mag topic is arguably redundant since the normalised data it posts can be derived from the data already being sent via the ardrone/navdata topic. This creates overhead. The object-oriented nature of gusimplewhiteboard messages enables a cleaner design, by placing an instance method that handles navdata and encapsulates the normalisation code only when required, rather than for each posting. Contrasting ardrone_autonomy versions 1 and 1.4 shows the attempts to overcome some of ROS' inherent limitations; in particular the transmission delays resulting from large messages. In version 1.4, clients have access to a collection of messages to choose portions from the large 'legacy' topics. These new messages are generally shorter, and thus faster to transmit. However, producing smaller messages from the original legacy blob has the same design issues as the `ardrone/navdata` topic example above. From a TTA point of view, the 'legacy' messages are the most reliable and efficient status messages as they only require the header to be computed a small number of times and are sent as atomic units. The family of topics that are replacing them in version 1.4 require more computational overhead to ensure packets being used have the correct header details (particularly on non-real-time operating systems), due to transport time variances. With the `gusimplewhiteboard`, transporting a large message such as `ardrone/navdata` does not incur a significant performance penalty, eliminating the above ROS problems which is demonstrated in the experiments we describe next.

4 Performance

4.1 First Experiment

In the first experiment, the high-level behaviour consists of a sequence of commands sent to the drone to perform a dance routine that was originally manually choreographed to background music. The routine is a dead-reckoning sequence, thus, there is no feedback from sensors. As already explained, common elements are the Wi-Fi relay at 30 Hz, the `telop_twist update_teleop()` method runs at 40 Hz, and the controlling application runs in a separate process. This first experiment compares the PULL-approach with the PUSH-approach ensuring that measurements are accurate across processes, while being completely equivalent between the ROS or the `gusimplewhiteboard` implementations. Thus, the monitoring code has exactly the same number of data collection points for each configuration, and the same monitoring source code at each test point. Moreover, monitoring in this first experiment introduces very little additional load. The tests were designed such that timing variances would be consistent across tests (i.e. no other active applications would be running or launched during the tests) and performance calculations would be based on relative values. The tests were implemented in a manner that allowed them to be turned on and off when required through compiler directives. Since the `gusimplewhiteboard` shared memory access time is significantly faster than ROS and did not require additional threads, in this first experiment, the timing data was collected in the `gusimplewhiteboard` itself. This enabled an external process to collect the data and log it. The application issues three types of control messages, `Cmd_Vel`, `Take-off`, and `Land`. A sample point is a point where we collect the absolute system time in microseconds using the POSIX gettimeofday() function. Such sample points were chosen at analogous positions in the processing paths of ROS and the `gusimplewhiteboard`. We designated the sample points at the following locations (as depicted in Figs. 5 and 6):

A The client applications issues the control message,
B Time the command was received in *AA*[2] in the ROS callback or in the new preamble section of update_teleop in the case of the gusimplewhiteboard,
C the command was actuated in *AA*; the message data was processed,
D update_teleop received the command; the method proper,
E the command is forwarded to ARDroneLib,
F the update_teleop reaches its end, and
G the issuing of the command is completed in the client.

Figure 7 shows the sample points inside the callback CmdVelCallback for collecting Cmd_Vel data under ROS, corresponding to sample positions **B** and **C**. By comparison, Fig. 8 shows the analogous gusimplewhiteboard implementation within update_teleop(). Each collection has a sequence number incorporated into the design of the wb_ardrone_performance class. Then, the frequency of the gusimplewhiteboarď s publish/subscribe functionality (*deprecated*) was increased to 1 kHz (far above what it normally operates at), to allow a data logger to catch messages in a separate process. With the PUSH approach, subscribers are not able to keep up with this data rate and sometimes the elapsed time between

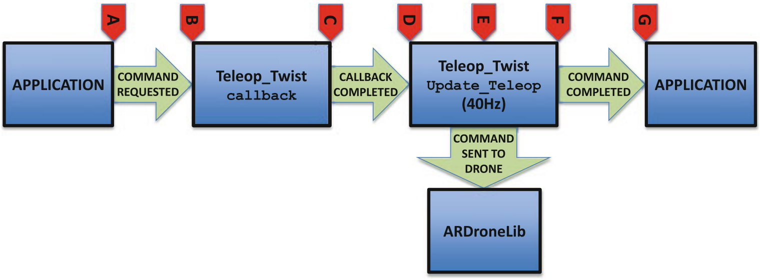


Fig. 5 Position of sample points in the PUSH-approach. represented by ROS

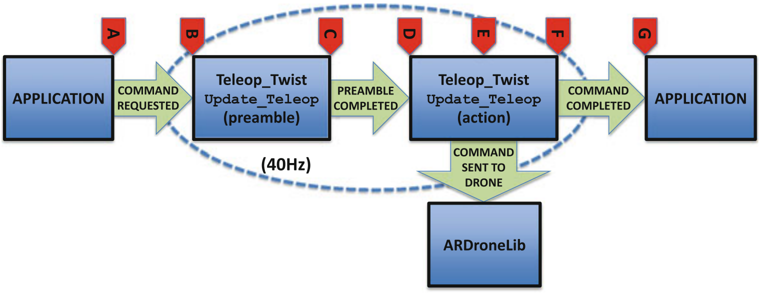


Fig. 6 gusimplewhiteboard sample point positions in the PULL-approach

[2] We used version 1.4 http://wiki.ros.org/ardrone_autonomy.

```
void CmdVelCallback(const geometry_msgs::TwistConstPtr &msg)
{
#ifdef WHITEBOARD_PERFORMANCE_TEST
  //Indicates how many times this callback has been accessed.
  cmdVelCBSeq++;

  // Message being received, event B.
  guWhiteboard::ARDrone_Performance_Cmd_Vel perfCmdVel;
  perfCmdVel.set_eventStrB("CmdVelCallback");
  perfCmdVel.set_bTime(get_utime());
  perfCmdVel.set_bSeq(cmdVelCBSeq);
  wbPerfCmdVel.set(perfCmdVel);
#endif // WHITEBOARD_PERFORMANCE_TEST

  vp_os_mutex_lock(&twist_lock);
  // Main 4DOF
  cmd_vel.linear.x  = std::max(std::min(-msg->linear.x, 1.0), -1.0);
  cmd_vel.linear.y  = std::max(std::min(-msg->linear.y, 1.0), -1.0);
  cmd_vel.linear.z  = std::max(std::min(msg->linear.z, 1.0), -1.0);
  cmd_vel.angular.z = std::max(std::min(-msg->angular.z, 1.0), -1.0);
  // These 2DOF just change the auto hover behaviour
  // No bound() required
  cmd_vel.angular.x = msg->angular.x;
  cmd_vel.angular.y = msg->angular.y;
  vp_os_mutex_unlock(&twist_lock);

#ifdef WHITEBOARD_PERFORMANCE_TEST
  has_cmdvel = true;
  // Request actuated, event C.
  perfCmdVel = wbPerfCmdVel.get();
  perfCmdVel.set_eventStrC("Callback_done");
  perfCmdVel.set_cTime(get_utime());
  perfCmdVel.set_cSeq(cmdVelCBSeq);
  wbPerfCmdVel.set(perfCmdVel);
#endif // WHITEBOARD_PERFORMANCE_TEST
}
```

Fig. 7 Cmd_Vel data collection under ROS

some sample points is very small, in particular, between **B** and **C**. Redundancy was designed into data collection structures were by the time stamps were persistent through the whole command execution cycle in ARDrone_Whiteboard. However, this is to be expected as what really costs time is the enacting and returning from such a call-back. The sequence number enabled us to track the overall start-to-end (round trip) execution times as well as detect any message losses.

Data was collected from two flights of a Parrot AR Drone 2.0 that consisted of 152 Cmd_Vel commands each. The first flight used the PUSH-approach, while the second flight used the PULL-approach. The computer used here for all experiments was a standard 2014 Retina Macbook Pro 15”, running Ubuntu 14.04.02, ROS Indigo Desktop, and *AA* 1.4.0, and Wicd 1.7.2.4 Network Manager. All software used for testing was compiled with debug symbols enabled (and used during performance testing).

```
    cmdVel = wbCmdVel.get();
    if (cmdVel.requestCmd()) {
#ifdef WHITEBOARD_PERFORMANCE_TEST
      //Indicates how many CmdVel messages have been received.
      cmdVelSeq++;

      // Message being received, event B.
      guWhiteboard::ARDrone_Performance_Cmd_Vel perfCmdVel;
      perfCmdVel.set_eventStrB("WB_CmdVel_Received");
      perfCmdVel.set_bTime(get_utime());
      perfCmdVel.set_bSeq(cmdVelSeq);
      wbPerfCmdVel.set(perfCmdVel);

      /** Further below, 'has_cmdvel' ensures performance messages are only
       *   posted when a cmdvel command is being processed.
       *
       *   'update_teleop' treats velocity settings as a state, not an event.
       *   When measuring performance, we only want to send data when
       *   events occur so as to match received packets with output
       *   packets.
       */
      has_cmdvel = true;
#endif // WHITEBOARD_PERFORMANCE_TEST

      cmdVel.set_requestCmd(false);
      // Main 4DOF
      cmd_vel.linear.x  = std::max(std::min(-cmdVel.cmd().linear().x(), 1.0), -1.0);
      cmd_vel.linear.y  = std::max(std::min(-cmdVel.cmd().linear().y(), 1.0), -1.0);
      cmd_vel.linear.z  = std::max(std::min(cmdVel.cmd().linear().z(), 1.0), -1.0);
      cmd_vel.angular.z = std::max(std::min(-cmdVel.cmd().angular().z(), 1.0), -1.0);
      // These 2DOF just change the auto hover behaviour
      // No bound() required
      cmd_vel.angular.x = cmdVel.cmd().angular().x();
      cmd_vel.angular.y = cmdVel.cmd().angular().y();
      wbCmdVel.set(cmdVel);


#ifdef WHITEBOARD_PERFORMANCE_TEST
      // Message actuated, event C.
      perfCmdVel = wbPerfCmdVel.get();
      perfCmdVel.set_eventStrC("WB_CmdVel_Actuated");
      perfCmdVel.set_cTime(get_utime());
      perfCmdVel.set_cSeq(cmdVelSeq);
      wbPerfCmdVel.set(perfCmdVel);
#endif // WHITEBOARD_PERFORMANCE_TEST

    }
```

Fig. 8 Cmd_Vel data collection under gusimplewhiteboard

4.1.1 Data Analysis

For evaluation, we focused on the following measurements:

B to F : Total `ARDrone_Whiteboard` Message Processing Time.

D to F in the PUSH-approach (ROS) compared to **B to F** in the PULL-approach: Additional `Update_Teleop` Processing Time Due to inclusion of Preamble (but in the PULL-approach we are doing the work of the callback **B to C**.

A to G : Client-Side Message Processing Time (Command Request time)

A to E : Delay from Command Request (in Client) to it being sent to `ARDroneLib`

A to B : Command Request to `ARDrone_Whiteboard` commencing response.

In this experiment, we do not have large volumes of data. Nevertheless, important trends are clearly visible (refer to the histograms in Figs. 9 and 10). A summary of the data collected for the `Cmd_Vel` command's performance is shown in Table 1. The overhead of the system call to retrieve the time at each data collection point is included in the results. Therefore, in the "Total `ardrone_whiteboard` Message processing time" measurement for the `gusimplewhiteboard` and "Client Side Message processing time" for both systems, the reported times are affected by more invocations that happen in between. Nevertheless, the data show that the PULL-approach outperforms the PUSH-approach throughout. The most important

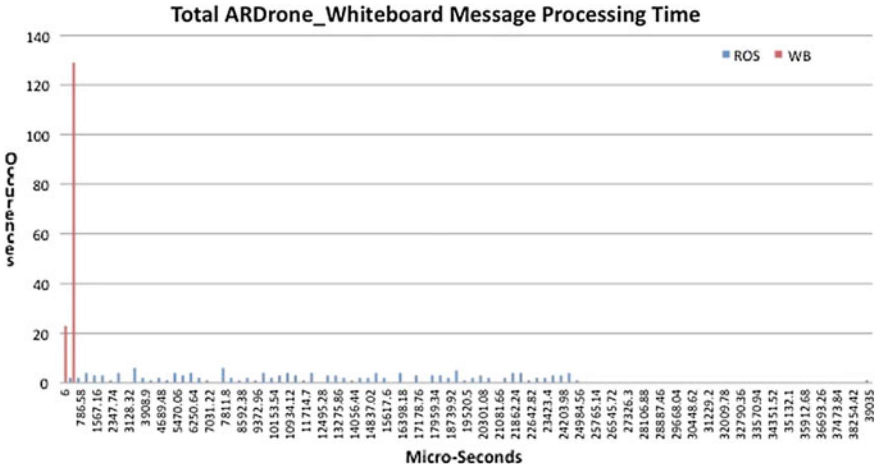


Fig. 9 Histogram comparing total message processing time

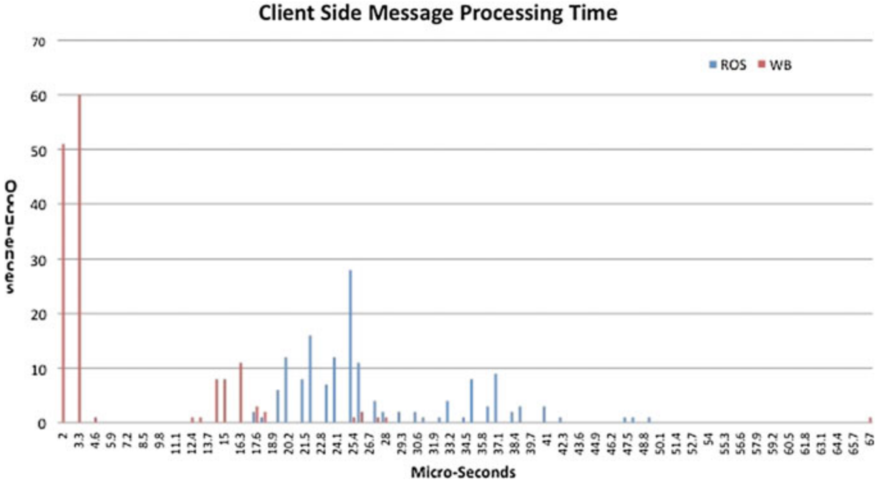


Fig. 10 Histogram comparing total delay at the client

Table 1 Measurements (in μSec) of relay performance for the control `Cmd_Vel` message

	ROS				gusimplewhiteboard			
Segment	Mean	StdDev	Min	Max	Mean	StdDev	Min	Max
A–B	17,115	1,764	13,858	20,174	12,664	7,445	101	25,083
A–D	29,581	7,964	15,990	55,276	12,674	7,445	106	25,089
A–E	29,582	7,963	15,991	55,278	12,675	7,445	107	25,090
A–F	29,583	7,963	15,992	55,278	12,676	7,445	108	25,090
B–F	12,468	7,672	111	39,035	11	8	6	52
C–D	12,461	7,673	106	39,030	6	6	3	38
A–G	26.94	6.81	17	49	6.61	8.18	2	67
D–F	2.12	1.36	1	13	1.74	2.01	1	12
B–C	4.97	4.85	2	20	4.30	4.73	1	20
D–E	1.55	1.31	1	12	1.22	2.00	0	12
E–F	0.57	0.50	0	1	0.53	0.50	0	1
D–F/B–F	2.12	1.36	1	13	11.93	8.88	6	52

measure is the amount of time that `ardrone_whiteboard` takes to process a received message ("Total ardrone_whiteboard Message processing time"), as it is a direct comparison of the PUSH vs PULL paradigms. We note that the original design explicitly sets the `ROS::Rate` parameter in *AA* to 50 Hz in an attempt to regulate the timing of this interface. The `gusimplewhiteboard` does not have this problem, and the results demonstrate those advantages.

Another important result, though not as dramatic, is the comparison of the impact this has on the client's time, posting the command message ("Client Side Message processing time"). Here, the `gusimplewhiteboard` is again considerably faster (comparing the time it takes to for the `gusimplewhiteboard` handler to copy the `ARDrone_Cmd_Vel` object into the shared memory with using `ROS` to publish the message and then call `ROS spinOnce()`). This result could be critical when considering postings of large numbers (or sizes) of messages or when the system being controlled demands a high response frequency. The "Delay from issuing the Command to it being sent to the drone" is very high for both systems, due to the lack of synchronisation between the client and `update_teleop()`. Under `ROS`, in several cases the delay exceeded the 25ms `update_teleop()` period and under `gusimplewhiteboard`, several samples approached the 25ms barrier, but never exceeded it. This kind of performance is expected in an ad hoc open loop system. What is significant about this result, is the greater `gusimplewhiteboard` improvement potential in a tuned feedback control loop scenario, only constrained by the "Total `ardrone_whiteboard` Message processing time" when compared to `ROS`.

4.1.2 Reliability

As *AA*'s `teleop_twist update_teleop()` loop runs at 40 Hz, and the test application was sending its message at intervals generally greater than one second, it would be expected that all messages would be successfully delivered. But, even at this slow posting rate, there were several occasions where `update_teleop()` did not pick up the message, using `ROS`, until its next iteration. Whilst not significant in this use case, it shows that with systems that have a higher frequency response it would be `ROS` that would falter first, and Table 1 supports this observation. A careful analysis of the output shows that all the data in both systems was correctly received. This was confirmed by matching the message counter in the samples. We observed that the monitoring code of the `gusimplewhiteboard` although running at 1KHz, missed some time recordings indicating that the `ARDroneDriver::Run() loop`, which runs at 50 Hz, would miss even more messages, or deliver them late when a queue was in place. This prompted our second experiment.

Table 2 List of commands used in an iteration

Seq #	Command	Seq #	Command
1	hoverDrone(0.05,"P01");	22	hoverDrone(0.05,"P38");
2	hoverDrone(0.05,"P11");	23	moveDown(0.05,"P39");
3	moveUp(0.05,"P12");	24	hoverDrone(0.05,"P3A");
4	hoverDrone(0.05,"P13");	25	hoverDrone(0.06,"P3B");
5	moveDown(0.05,"P14");	26	moveBackward(0.05,"P41");
6	hoverDrone(0.05,"P15");	27	hoverDrone(0.06,"P42");
7	moveRight(0.05,"P21");	28	hoverDrone(0.04,"P43");
8	hoverDrone(0.05,"P22");	29	moveUp(0.05,"P44");
9	hoverDrone(0.05,"P23");	30	hoverDrone(0.05,"P45");
10	moveUp(0.05,"P24");	31	moveDown(0.05,"P46");
11	hoverDrone(0.05,"P25");	32	hoverDrone(0.05,"P47");
12	moveDown(0.05,"P26");	33	hoverDrone(0.06,"P48");
13	hoverDrone(0.04,"P27");	34	moveBackward(0.05,"P51");
14	hoverDrone(0.04,"P28");	35	hoverDrone(0.05,"P52");
15	moveForward(0.03,"P31");	36	moveUp(0.05,"P53");
16	hoverDrone(0.05,"P32");	37	hoverDrone(0.05,"P54");
17	hoverDrone(0.05,"P33");	38	moveBackward(0.05,"P55");
18	moveLeft(0.04,"P34");	39	hoverDrone(0.05,"P56");
19	hoverDrone(0.06,"P35");	40	moveUp(0.05,"P57");
20	hoverDrone(0.05,"P36");	41	hoverDrone(0.05,"P58");
21	moveUp(0.05,"P37");	42	moveUp(0.05,"P61");
		43	hoverDrone(0.06,"P62");

Command parameters are:
1st Delay in seconds using **ros::Duration().sleep()**
2nd Message to print (which was suppressed for the large dataset run)

4.2 Second Experiment

To create larger datasets while accounting for the drone's battery discharge rate during flight, we increased the application command transmission rate in the second experiment by an order of magnitude, from 300 ms down to 30 ms intervals. Table 2 lists the command sequence used over 5,046 iterations +(2.14 sec per iteration) for an approximate flight time near to +(3) hours. A small number of commands were within two times the `update_teleop()` polling time.

Another refinement of this experiment was to remove external probe effects from the code, by using a pre-allocated `C++` vector instead of the whiteboard for recording measurements, thus leaving the `gusimplewhiteboard` code to only perform control/status functions. `ARDrone_Whiteboard` and the application were modified accordingly. In order to increase clock accuracy and reduce overhead, we used `clock_gettime(CLOCK_MONOTONIC_RAW)` in place of `gettimeofday()`. The data collection points are the same as in the first experiment (time stamp and sequence number only, the latter in place of the unused `cmd_vel.angular.x` parameter). Table 3 confirms the contrasting performance of the `gusimplewhiteboard` over `ROS` but now with 216,978 samples, which allows us to show 95 % confidence intervals. Moreover, to show the remarkable difference in performance by the two architectures we summarise the data of the 216,439 common samples in box-plots in Fig. 11. All the values of **B**–**F** and **C**–**D** for the `gusimplewhiteboard` are so small they basicly disappear from the figure. While the performance results were in line with in experiment 1, (in fact, the **A**–**B** mean for the sweep for the `gusimplewhiteboard` was 12.497 ms, extremely close to the expected 12.5 ms), under `ROS`, messages were actually getting lost inside `ARDrone_Whiteboard` (Table 4 on the next page), a situation that would require extremely complex end-to-end handshaking to recover from.

Table 3 Summary of measurements of relay performance for the control `Cmd_Vel` message for second experiment (216,978 samples)

	ROS				gusimplewhiteboard			
segment	Mean 95 %c.i	StdDev	Min	Max	Mean 95 %c.i.	StdDev	Min	Max
A–F	22,731 ± 7	9,254	164	49,280	12,503 ± 7	7,251	5	26,966
B–F	12,554 ± 14	7,240	3	27,550	6 ± 0.005	3	1	476
C–D	12,461 ± 15	7,673	106	39,030	6 ± 0.011	6	3	38
A–G	31 ± 0.37	18.99	5	4,902	1 ± 0.003	1	0.31	556

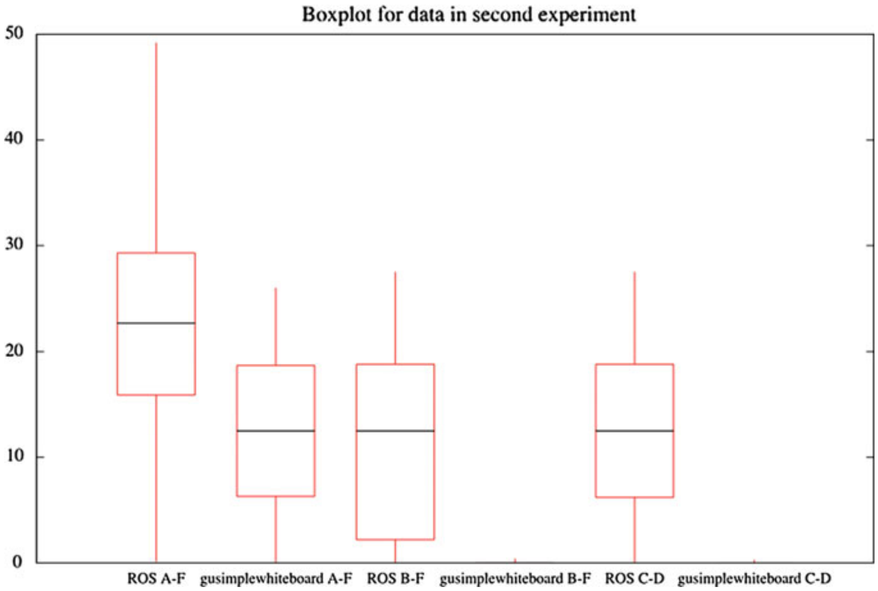


Fig. 11 The visual companions of the performance difference between ROS and gusimplewhiteboard for the second experiment

Table 4 Summary of Lost End-to-End Messages (misses) in ROS. All misses occurred in the segment between **C** and **D**

Command type	Occurrences
12	9
13	4
14	522
17	5
27	8
Total misses	539

The total number of samples is 216,978. While the ROS callback is able to detect the missed message it can not do so in time to be processed by Update_Teleop()

5 Conclusion

We have successfully used the gusimplewhiteboard to reproduce the hardware abstraction of the *AA* package in our ardrone_whiteboard implementation. The new abstraction introduces predictable performance. Experiment one shows that a slow posting rate does not lose messages in the PUSH approach, but introduces a significant delay. This latency may significantly decrease the stability of a *MAV*. However, in our second experiment, we increase the frequency at which commands are generated in the application, and in this case the PUSH approach loses messages, which means that, potentially critical command events are lost. We have demonstrated that the PULL approach represented by the `gusimplewhiteboard` is significantly faster, more reliable, and better suited for minimising overhead by middleware than the PUSH approach represented by `ROS::Messages` and `ROS::Services`, particularly in cases that require higher frequencies of operation. We hope this case study will contribute to the understanding of the PULL approach (and its shared memory implementation in the `gusimplewhiteboard`) and its importance when it comes to predictable, timing-critical messaging. In future work we expect to show that even greater performance is possible as the PULL approach could easily be synchronised with the 30 Hz WiFi relay cycle. We would like to emphasise that we view the PULL approach as complementary to the PUSH approach and expect that the results here inform the community to consider using the PULL approach in critical systems rather than the current uniformity of using the PUSH approach in robotic middleware.

For further work we will apply this approach to micro-air vehicle real-time control.

Acknowledgments The authors wish to thank Dr. Jun Jo who made equipment and infrastructure available.

References

1. Bernardini, S., Fox, M., Long, D.: Planning the behaviour of low-cost quadcopters for surveillance missions. In: Chien, S., Do, M.B., Fern, A., Ruml, W. (eds.) Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling, ICAPS 2014. AAAI, 21st–26th June 2014
2. Bogert, K.D., Solaimanpour, S., Doshi, P.: Aerial robotic simulations for evaluation of multi-agent planning in GaTAC. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. pp. 1919–1920. AAMAS '15, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2015)
3. Chitic, S.G., Ponge, J., Simonin, O.: Are middlewares ready for multi-robots systems? In: Brugali, D., Broenink, J.F., Kroeger, T., MacDonald, B.A. (eds.) Simulation, Modeling, and Programming for Autonomous Robots—4th International Conference, SIMPAR 2014, Bergamo, Italy, 20–23 Oct 2014. Proceedings. Lecture Notes in Computer Science, vol. 8810, pp. 279–290. Springer (2014)
4. Elkady, A., Sobh, T.: Robotics middleware: a comprehensive literature survey and attribute-based bibliography. J. Robot. (2012). article ID 959013
5. Estivill-Castro, V., Hexel, R.: Logic labelled finite-state machines and control/status pull technology for model-driven engineering of robotic behaviours. In: 26th International Conference on Software & Systems Engineering and their Applications. AFIS—French Association for Systems Engineering & TELECOM ParisTech, Paris, 27th–29th May 2015
6. Estivill-Castro, V., Hexel, R.: Simple, not simplistic—the middleware of behaviour models. In: ENASE 10th International Conference on Evaluation of Novel Approaches to Software Engineering, pp. 189–196. INSTCC, Barcelona, Spain, 29th–30th April 2015
7. Estivill-Castro, V., Hexel, R., Lusty, C.: High performance relaying of C++11 objects across processes and logic-labeled finite-state machines. In: Brugali, D., Broenink, J.F., Kroeger, T., MacDonald, B.A. (eds.) Simulation, Modeling, and Programming for Autonomous Robots—

4th International Conference, SIMPAR 2014. Lecture Notes in Computer Science, vol. 8810, pp. 182–194. Springer, Bergamo, Italy, 20th–23rd Oct 2014

8. Koenig, N.P., Howard, A.: Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: IROS IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 2149–2154. Sendai, Japan (2004)
9. Kopetz, H.: Real-Time Systems: Design Principles for Distributed Embedded Applications, 2nd edn. Springer Science+Business Media (2011)
10. Monajjemi, M.: Ardrone autonomy: Ros driver for parrot ar-drone 1.0 and 2.0 quadrocopters (2012). https://github.com/AutonomyLab/ardrone_autonomy
11. Pestana, J., Sanchez-Lopez, J., Saripalli, S., Campoy, P.: Computer vision based general object following for gps-denied multirotor unmanned vehicles. In: American Control Conference (ACC) 2014, pp. 1886–1891 (2014)
12. 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)

Framework and Modeling of a Multi-robot Simulator for Hospital Logistics

Seohyun Jeon and Jaeyeon Lee

Abstract An autonomous mobile robot has opened a new era of hospital logistics. Mobile robots will substitute human courier by transporting goods within the hospital day and night. Multiple robots will be deployed at a hospital and coordinating these robots will increase the efficiency of delivery. This paper introduces the framework of the simulator which enables testing coordination algorithms of multiple robots. The simulator models hospital environment with the hospital-specific delivery demand as well as the specification of robots used for the delivery. The outcome of the simulator is the performance of applied task allocation algorithms, which includes overall delivery time and waiting time of delivery tasks.

1 Introduction

There are numerous materials and samples to transport in a hospital, such as blood, laboratory samples, sterile goods, linens, medicine, or clinical wastes. They are delivered from wards, operating room, clinical laboratory, pharmacy, laundry, kitchen, and central storage. Conventionally, the transportation mostly relies on human couriers. They work in a physically harsh environment since they frequently turnaround nearly tens of kilometers a day pushing a heavy cart. The working hours for these tasks are up to 195 h per week [1]. To reduce such human effort, systems are developed, such as dumbwaiter, pneumatic tube, and auto track conveyor. However, these systems need to be considered from the beginning of the construction of the building for implementation since they are not flexible to change the infrastructure. Also, the maintenance costs are expensive. On the other hand, a mobile robot for hospital logistics has been developed by Aethon Inc. and Swisslog Inc [2]. It is

S. Jeon (✉) · J. Lee
Human Robot Interaction Research Team Electronics and Telecommunications
Research Institute (ETRI), Daejeon 305-700, South Korea
e-mail: happyseohyun@etri.re.kr

J. Lee
e-mail: leejy@etri.re.kr

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_17

able to travel anywhere inside a hospital without additional construction effort for guiding robots since it autonomously generates the optimal path from sensory data that saves time and avoids collision with the environment [3]. Delivery with a mobile robot has advantage in a hospital environment since it does not require preinstalled infrastructure, works 24 h a day and saves costs and human labor. A research shows that 6 robots save 2.8 full-time equivalent employees [4, 5]. The delivery log can be reported which helps tracking materials and managing the inventory. Nowadays, more hospitals are opening their places for the robot courier such as UCSF medical center in which 25 Tugs are already in service [6, 7]. When a fleet of mobile robots are used in a place, a coordinator is needed to increase the productivity which assigns a delivery task to an appropriate robot. To begin with, types of delivery tasks inside the hospital are analyzed and the scheduling algorithm is introduced according to the delivery type. The delivery type is preprogrammed in the simulator to develop the scheduling algorithm. This paper will analyze the delivery type and discuss the framework of the simulator to coordinate multiple robots in hospital logistics.

2 Types of Delivery Tasks and Robots for Hospital Logistics

2.1 Hospital Delivery

Many kinds of supplies are delivered inside a hospital. Samples for laboratory examination arrive frequently, drug medicines are delivered upon request, and sterilized surgical instruments are distributed in the morning to the operating room. Meals are delivered three times a day, linens are delivered to wards twice a day, at dawn and in the afternoon, and medical wastes are collected from time to time. These logistics can be categorized as follows:

Scheduled service There are prescheduled timely services for delivery. Cleaned linens and prepared meals are included in this type. Punctuality is an important matter in this type of service.

On-demand service Nurses and staffs frequently order on-demand delivery service. Laboratory samples occur from everywhere in the hospital and are delivered to the clinical laboratory. Similarly, medicines are delivered from pharmacy to wards upon request.

Roundabout service Used linens and finished food tray are collected when robots are available. Also, medical wastes are collected in this manner. In this type of service, punctuality is not important and the robot routes several positions periodically.

The difference between scheduled and roundabout service is the punctuality. Scheduled service should meet on time, whereas roundabout service does not require strict delivery time but transports whenever the robot is available. Scheduled and roundabout services are preprogrammed in the coordinating system, but on-demand service is called by request.

Table 1 Type of a delivery robot

Weight	Container	Security	Contamination
Heavy	Cart	Open access	Contaminated
Light	Drawer	Authorized access	Sterilized

2.2 Delivery Robot

The type of robot used for hospital delivery is not homogeneous. It can be categorized into several types based on the purpose and the usage: Weight, container, security, and contamination (Table 1). The weight can be divided into heavy and light. A robot carrying meals and linens should be able to carry heavy load. Light and agile robot is suitable for carrying small materials such as medicine, blood and other laboratory samples. Also, the container type can be divided into cart and drawer. A cart-type robot can pile materials upto its payload and tow additional cart behind. This type usually carries heavy goods such as linen and meals. A drawer-type robot is a robot mounted with multiple drawers and moves only itself without towing a cart. This type of robot carries small goods or secured materials such as medicine. Accordingly, some of the robot need security lock when delivering drugs and medicine. Only authorized person is accessible to the contents. On the other hand, people might not be interested in the contents that are carried by a robot, such as linens or wastes which does not require security lock. Also, separated robots should be used when delivering sterilized or contaminated goods. Since different types of robots are used for dedicated purpose in hospital logistics, types of robot should be taken into account when scheduling robots (Table 1).

3 Simulator

The simulator consists of three parts as in Fig. 1: *Task scheduler, core, and reporter*. The *task scheduler* contains the user-defined scheduled and roundabout service, the list of requesting time, material, and positions for delivery. For on-demand service, the user enters the set of positions for pickup and drop-off at the simulator on-line, and these are coordinated by *task allocator*. The user can implement several task allocation algorithms in the *task allocator* and test on the simulator. The results of tested allocation algorithms can be seen in the *reporter*.

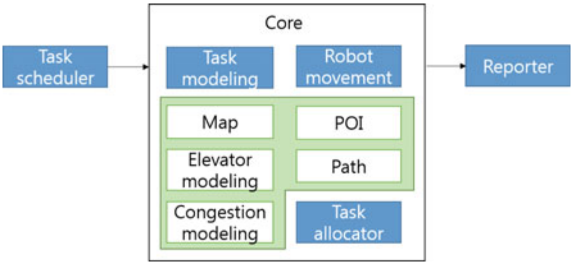
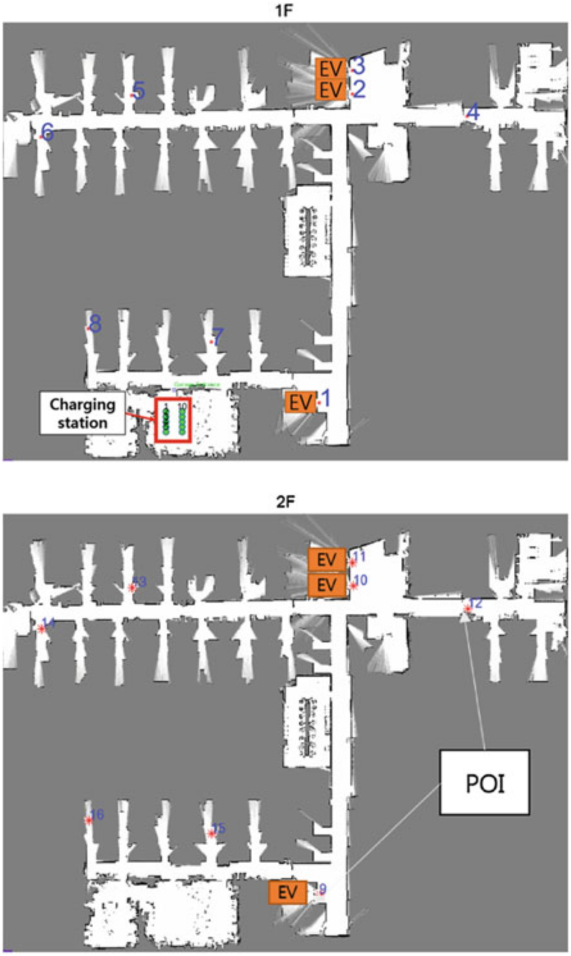


Fig. 1 The framework of the hospital logistics simulator

Fig. 2 The simulator layout showing two floors



The hospital environment is modeled in the *core*. Based on the 2D binary map, the position of frequently delivered spots is marked as the point of interest (POI) as can be seen in Fig. 2. POIs can be nursing units, wards, storage, laboratory, kitchen, and laundry. POIs are marked on the map and expressed as a topology that consists of node and edges. Edge of the topology denotes the distance and the level of congestion of the pathway which varies according to the time. A robot moves between POIs along with the generated path considering the congestion level. Since there exists multi-floor transportation, the location and status of the elevator are also simulated considering the stochastic distribution. The task allocator monitors the position and status of all robots, and decides which robot should be assigned for the requested delivery task. The selected robot will regenerate the path corresponding to the newly assigned task. The task allocator will consider the efficiency and appropriateness of the allocation. This will be discussed in the next section, about the scheduling algorithm. The *core* runs and calls the delivery task from the *task scheduler* and moves and manages status of robots by simulating the 24 h of hospital logistics concisely.

After simulation is finished, the reporter shows the result of the applied task allocation algorithm. It includes the actually triggered time of the delivery task, waiting time, task completion time, and also, the operation rate of multiple robots. With the same scheduling algorithm, different results can be derived when simulating with different numbers of robot. Also, various scheduling algorithms can be tested in the fixed environment and the performance can be compared.

4 Scheduling Algorithm

After types of delivery task and robot are defined, these properties should be considered in the scheduling algorithm. Coordinating such heterogeneous delivery robot should follow the constraints that certain robot should not be cross used, i.e., sterilized and contaminated goods, and that a robot should not receive more task when the battery is low. To increase the delivery productivity of using multiple robots, a robot can response to more than one task. Assume a robot is conducting a set of task, that is, pickup (PU) from P1 position to drop-off (DO) at D1 position. If a new delivery task is requested from P2 to D2, and all robots are out for delivery, the new task should be added to current working robot. Then, the possible combination of the added assignment for a robot can be seen as Fig. 3. Delivery is a matter of time: how fast it picks the goods and delivers to the destination. The schedulingalgorithm

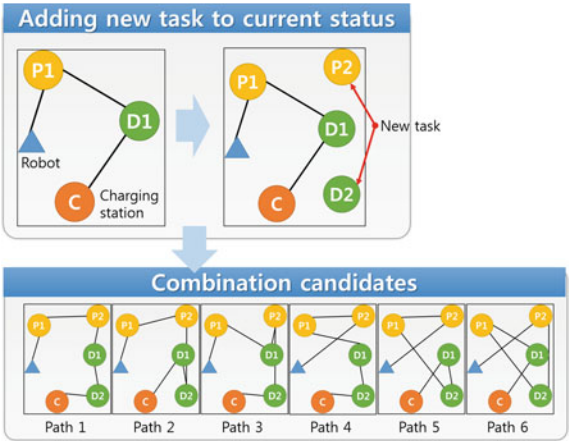


Fig. 3 Current task of a robot is P1(PU)→D1(DO). If a new delivery task, P2→D2, is requested, 6 path candidates can be generated by the constraints that PU and DO cannot be reversed. Considering the total delivery time and the waiting time at pickup position, the path with minimum cost will be selected for a robot. Similarly, this selection will be performed by all available robots, and the robot suggesting overall minimum cost will be selected for the task

decides which robot should be assigned for the task to respond the requested delivery as soon as possible while considering the total delivery time.

Allocating a task to multiple robots for hospital logistics is based on finding a closest robot to the pickup positions that minimizes the waiting and delivery time, and this problem is similar to the vehicle routing problem with multiple couriers (mVRP) [8]. The difference is that couriers in the hospital logistics are heterogeneous. Also, the former has to respond to real-time dynamic delivery requests which occur arbitrarily whereas the latter is about finding the optimal path off-line.

5 Conclusion and Future Work

Starting with a robot cleaner through warehouse logistics, an autonomous mobile robot has found its new application for hospital logistics. Once the robotic technology is mature enough to be deployed at a hospital, the management issue arises that how many robots are needed to cover the hospital logistics and how to coordinate these robots. The number of required robots varies according to the hospital specific environment such as the scale of the hospital, the amount and kinds of delivery goods and their rates, and travel velocity and distance of a robot. Such parameters are implemented in the proposed simulator and enable to measure the performance regarding the number of robots and allocation algorithms. The performance is the simulated result which includes the total delivery time, average waiting time, and operation rate of robots. The user can decide the number of robots after reviewing the result. Based on the developed hospital logistics simulator, the several allocation algorithms will be tested and compared in the future work.

Acknowledgments This work was supported by the Robot R&D program of MOTIE/KEIT. [10051155, The Development of Robot Based Logistics Systems Applicable to Hospital-wide Environment]

References

1. Bardram, J.E., Bossen, C.: Mobility work: the spatial dimension of collaboration at a hospital. Comput. Support. Coop. Work (CSCW) **14**(2), 131–160 (2005)
2. Niechwiadowicz, K., Zahoor, K.: Robot based logistics system for hospitals-survey. In: IDT Workshop on interesting results in computer science and engineering (2008)
3. Calderon, C.A.A., Mohan, E.R., Ng, B.S.: Development of a hospital mobile platform for logistics tasks. In: Digital Communications and Networks, pp. 289–294 (2015)
4. Qureshi, M.Q., Syed, R.S.: The impact of robotics on employment and motivation of employees in the service sector, with special reference to health care. Safety Health Work **5**(4), 198–202 (2014)
5. Rossetti, M.D., Kumar, A., Felder, R.A.: Mobile robot simulation of clinical laboratory deliveries. In: Proceedings of the 30th Conference on Winter Simulation, pp. 1415–1422 (1998)

6. Ljungblad, S., Kotrbova, J., Jacobsson, M., Cramer, H., Niechwiadowicz, K.: Hospital robot at work: something alien or an intelligent colleague? In: Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW), pp. 177–186 (2012)
7. http://www.ucsfmissionbayhospitals.org/articles/high-tech-tug-robots-do-heavy-lifting-at-mission-bay.html. Accessed 25 Nov 2015
8. Nallusamy, R., Duraiswamy, K., Dhanalaksmi, R., Parthiban, P.: Optimization of multiple vehicle routing problems using approximation algorithms. Int. J. Eng. Sci. Technol. **1**(3), 129–135 (2009)

Emotion in Robot Decision Making

Rony Novianto and Mary-Anne Williams

Abstract Social robots are expected to behave in a socially acceptable manner. They have to accommodate emotions in their decision-makings when dealing with people in social environments. In this paper, we present a novel emotion mechanism that influences decision making and behaviors through attention. We describe its implementation in a cognitive architecture and demonstrate its capability in a robot companion experiment. Results show that the robot can successfully bias its behaviors in order to make users happy. Our proposed emotion mechanism can be used in social robots to predict emotions and bias behaviors in order to improve their performances.

Keywords Emotion · Subjective bias · Decision making

1 Introduction

Social robots are required to behave in a socially acceptable manner. Behaving socially means that the robots are required to make decisions or select actions not only based on goals, objective functions, or performance functions, but also based on subjective bias or social factors. In other words, these subjective bias or social factors have to be incorporated as part of the goals. For example, social robots are expected to wait for the right situation before delivering bad news or knock at a door before entering a room. These additional actions of waiting and knocking may slow the robots down and decrease their efficiency in achieving their goals of delivering news or entering a room. However, these actions are necessary when dealing with people in social environments. Therefore, social robots have to accommodate subjective bias in their decision making.

R. Novianto (✉) · M.-A. Williams
Centre for Quantum Computation & Intelligent Systems (QCIS),
University of Technology Sydney, Sydney, Australia
e-mail: rony@ronynovianto.com

M.-A. Williams
e-mail: mary-anne.williams@uts.edu.au

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_18

Emotions are one of the major sources of subjective bias and social factors. They have different influences on an individual's behaviors. For example, some people would undertake actions based on the pursuit of goals despite their emotions (e.g., do bungee jumping to receive money despite the fear of height), while others would choose differently in the same situation based on their emotions despite their goals (e.g., refuse bungee jumping because of fear of heights, despite the money offered).

The importance of emotion in decision-making has been supported by several works in cognitive science and psychology [1, 6, 10, 17]. Anderson [1] argues that decision-making incorporates rational evaluations and inferences along with anticipated emotions.

In the literature, various computational models of emotion have been proposed and implemented in robots [2, 3, 7–9, 16]. However, majority of them focus on emotion expressions and recognition (e.g., facial expression, body language, etc.). Little work have been proposed to influence decision-making and behaviors.

In this paper, we propose a novel emotion mechanism that influences decision-making and behaviors through attention. This mechanism is implemented in ASMO cognitive architecture [11], thus we call it ASMO's emotion mechanism. Section 2 first discusses existing computational models of emotion in robots and discusses how they are different to the proposed emotion mechanism. Section 3 then describes the design and implementation of the proposed emotion mechanism in ASMO cognitive architecture. Section 4 describes the experiment of ASMO's emotion mechanism in a robot companion problem and shows that the robot's decision-making and behavior can be influenced to achieve a goal successfully. Finally, Sect. 5 summarizes the work of ASMO's emotion mechanism.

2 Related Work

A recent comparison and summary of computational models of emotions in autonomous agents have been presented in the literature [4, 15]. Various computational models of emotion have also been proposed and implemented in robots [2, 3, 7–9, 16, 18]. However, majority of them focus on emotion expressions and recognition (e.g., facial expression, body language, etc.). Little work have been proposed to influence decision-making and behaviors. In the following, we discuss related computational models of emotions that are used to influence decision-making and behaviors.

Rosis et al. [14] proposed a formal model of fear based on Belief-Desire-Intention (BDI) framework. They distinguish two kinds of evaluations that cause emotions, namely cognitive evaluations and intuitive appraisals. A cognitive evaluation is a rational judgment supported by reasons, whereas an intuitive appraisal is an unconscious, automatic and fast nonrational judgment based on associative learning and memory rather than justifiable reasons. Their model describes only fear emotions and it is not implemented in robots.

Conati and Maclaren [4] has proposed a probabilistic approach using Dynamic Decision Networks (DDN). This model contains nodes to represent situations, such as a student's goal, an interaction pattern, an appraisal and an agent's action. It can handle uncertainty in emotions. It can also recognize emotions described by the OCC cognitive appraisal theory [13]. However, it does not allow opposite emotions to elicit simultaneously (e.g., happy and sad at the same time) and it is not implemented in robots.

Sawada et al. [16] proposed a computational model of emotion in robot based on the following internal biological needs: hunger, fullness, pain, comfort, fatigue, and sleepiness. This model can generate six basic emotions (follows Ekman's basic emotion theory [5]) and one neutral emotion. However, it does not accommodate the cognitive perspective of emotions. It does not allow opposite emotions to elicit simultaneously and does not handle uncertainty in emotions.

Breazeal [3] proposed emotion mechanisms called emotive releasers, affective appraisals, emotion elicitors, and emotion activation. Each emotive releaser can be thought of as a simple "cognitive" evaluation with respect to the robot's well-being and its goals. It has an activation level that is determined by evaluating attributes in the environment, such as the presence or absence of a stimulus (and for how long), the context of the stimulus (e.g., toy-related or person-related), or whether the stimulus is desired or not. Each releaser with activation above threshold will be tagged by the affective appraisals based on three dimensions in order to create a somatic marker, namely arousal, valence, and stance. These markers are mapped to some distinct emotions (e.g., anger, fear, etc.) by filtering them through the emotion elicitors. Finally, these distinct emotions compete for activation and they are selected by the emotion activation based on the winner-take-all scheme. In this model, activation levels, thresholds, releasers, and appraisals are designed by hand. Thus, this model is difficult to be used for general tasks and changing environments. It does not allow opposite emotions to elicit simultaneously and does not handle uncertainty in emotions.

This paper presents our proposed emotion mechanism to influence decision-making and behavior through attention. Our proposed emotion mechanism differs from previous work in the following: (i) it allows opposite emotions to elicit simultaneously, (ii) it uses a probabilistic model to handle uncertainty in emotions, (iii) it is a real-time model implemented in a physical robot and (iv) it is embedded in a cognitive architecture that can be used to explain the relationship between attention, emotions, and learning.

3 Proposed Approach

ASMO cognitive architecture is a theory and model that explain how humans make decisions based on what caught their attention [11]. It consists of self-contained, concurrent, modular, and distributed processes (also called modules) that compete for attention to gain access to their required resources. There are three types of processes which determine how they behave in the competition. For simplicity of this paper, we only describe ordinary and supervisor nonreflex processes.

An ordinary nonreflex process is a process that competes for attention by explicitly specifying the required amount of attention measured as *total attention level*. This total attention level is determined by the sum of two attributes in the process, namely *attention value* and *boost value* (1). The attention value attribute captures the degree of attention the process seeks based on the demand of the tasks. The boost value attribute represents the boost or bias associated with the process as a result of learning [12]. By convention, an ordinary process can only inspect and update its own attention value and boost value.

A supervisor nonreflex process is like an ordinary nonreflex process, except that it is allowed to inspect and modify attention values and/or boost values of other nonreflex processes.

$$\underbrace{TAL_t(p)}_{\text{Total}} = \underbrace{BV_t(p)}_{\text{Learned}} + \underbrace{AV_t(p)}_{\text{Determined}} \tag{1}$$

where:
$TAL_t(p)$ is the total attention level of process p at time t
$BV_t(p)$ is the boost value of process p at time t
$AV_t(p)$ is the attention value of process p at time t.

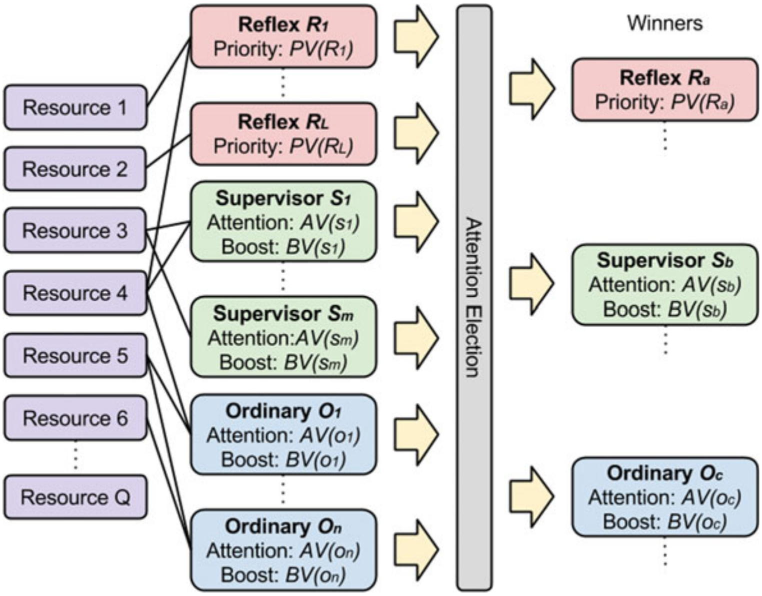


Fig. 1 Attention competition

In every attention competition (see Fig. 1), nonreflex processes are first ranked based on their total attention levels. The higher the total attention level, the higher the ranking. They are then selected from the highest to the lowest ranking depending on their required resources and available resources. Processes that require the same resources at the same time (i.e., have conflict in using resources) will compete for attention. Then, the process with the highest total attention level among them will win, be selected, and gain access to their required resources. In contrast, processes that have no conflict in using resources will eventually be selected despite their low rankings.

Our ASMO's emotion mechanism is proposed to influence decision-making and behaviors by modifying the attention values of nonreflex processes. It is implemented as a supervisor process in order to be allowed to inspect and modify these processes. Modifying these processes' attention values will change their total attention levels, which will then change their chances of winning in an attention competition. As a result, the selection of processes is effected.

ASMO's emotion mechanism determines the attention value of nonreflex process *np* based on the following two steps:

1. **Predict Feeling**
   ASMO's emotion mechanism uses multiple causal Bayesian networks to determine how likely feelings will be elicited given a situation and actions proposed by *np*. The probability of a feeling will be elicited, is calculated based on the conditional joint probability of the feeling using Bayes' rule (2). The higher the conditional joint probability, the more likely that the feeling will be elicited.
2. **Determine Expected Desirability of Feelings**
   ASMO's emotion mechanism then uses the predicted feelings to calculate their expected desirability (3). This expected desirability is the subjective bias (called subjective weight) that will be provided to modify the attention value of *np*.

$$P(h|e) = \frac{P(e|h) \times P(h)}{P(e)} \tag{2}$$

$$S_t(np) = \frac{1}{N_a(np)N_f} \sum_{i=0}^{N_a(np)} \sum_{j=0}^{N_f} P(f_j|a_i(np), e)D(f_j) \tag{3}$$

where:

$P(h|e)$ is the posterior probability. It is the probability of hypothesis h given or after evidence e is observed

$P(e|h)$ is the likelihood. It indicates how likely evidence e is associated with hypothesis h

$P(h)$ is the prior probability. It is the probability of hypothesis h before any evidence is observed

$P(e)$ is the marginal likelihood. It indicates how likely evidence e is observed regardless of the hypothesis

$N_f \leq N_{fn}$
N_f is the number of feelings that developers want the system to experience
N_{fn} is the total number of feeling nodes
N_a is the number of actions proposed by process *np*
S_t is the subjective weight provided to process *np* at time *t*
a_i is the ith action proposed by process *np*
e is the set of nodes that represents the situation (i.e., $e \leftarrow e_1, e_2 ... e_n$)
$P(f|a, e)$ is the conditional joint probability of feeling *f* given action *a* and a set of nodes *e* that represents the situation
$D(f)$ is the desirability of experiencing feeling *f*.

The following example describes how ASMO's emotion mechanism works. Imagine that a robot is desired to have emotions in a soccer scenario. ASMO's emotion mechanism can represent and have a model of possible feelings that may occur in the scenario, such as happy, sad, anger, and surprise. However, developers may consider only happy and sad feelings to be relevant in playing soccer. Thus, they may want the robot to have only these two feelings (i.e., $N_f = 2$), rather than all the possible feelings. Suppose that a nonreflex process called 'chase_ball' proposes an action to run to a ball and an action to track the ball at the same time (i.e., $N_a = 2$). A subjective weight of this process will then be determined based on the probability of each feeling (i.e., happy and sad) will be elicited given each action (i.e., walk and track).

The casual Bayesian Networks used in ASMO's emotion mechanism contains a set of nodes that connect actions, factors, labels, and dimensions in order. Actions performed by a robot are inferred by two kinds of factors, namely biological and cognitive factors. These factors represent the biological and cognitive perspectives of emotions, respectively. They can determine the reasons of why particular emotions are elicited. Labels are the specific types of emotions such as happy, sad, and fear. Finally, dimensions are the general representations of emotions, such as positive valence, negative valence, and arousal.

The dimension nodes are represented in bivariate dimensions rather than bipolar dimensions. A bivariate dimension suggests that two opposite qualities can occur at the same time. For example, people can feel both happy and sad at the same time during graduation. In this case, a positive valence is not the opposite of a negative valence. Instead, they are represented in separate nodes. Thus, positive valence and negative valence can be independent.

4 Evaluation and Discussion

ASMO's emotion mechanism is experimented in a robot companion problem using a bear-like physical robot called Smokey. The goal of this problem is to effectively accompany or entertain a person (i.e., target user) while simultaneously regulating the user's rest (see Fig. 2). Smokey can go to sleep to encourage the user to rest (since it will not interact with the user when it is sleeping). It can also play either a red ball

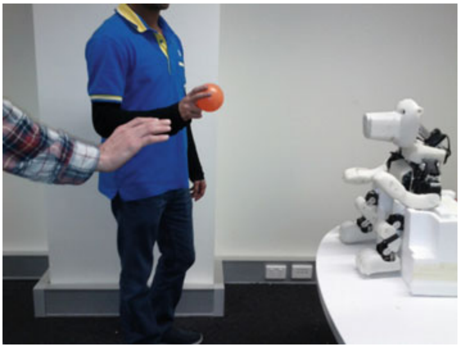


Fig. 2 Smokey robot companion experiment

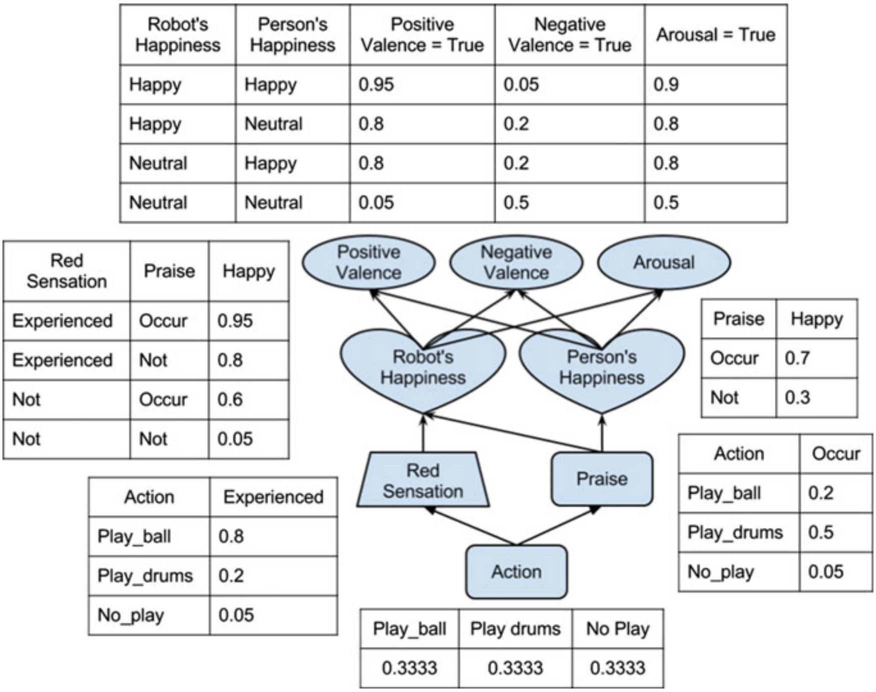
game or drums to accompany the user. When playing, it will pay attention to any motion in the environment from people other than the user.

In addition, Smokey is also desired to behave socially in order to accompany the user in a way that will make him or her happy. It can receive a request from the user through a graphical user interface to either play the ball, play the drums, or go to sleep. It will consider this request, but does not necessarily have to perform this request. As a companion robot, Smokey is designed to have an emotion towards red color and to like being praised. The more frequent Smokey receives the same request from a user, the more the user wants to see Smokey performing the request, the higher Smokey will believe that it will be praised for performing that request.

There are five processes created to handle Smokey's desired behaviors

1. **The attend_motion Ordinary Process**
   The attend_motion process proposes an action when Smokey is not sleeping to look at the fastest motion in the environment. Its attention value is set to the average speed of the motion scaled between 0.0 and 100.0. The faster the motion, the more attention demanded by the process to look at the motion.
2. **The play_ball Ordinary Process**
   The play_ball process proposes an action when Smokey is not sleeping to either track or search for the ball depending on whether the location of the ball is known or not respectively. Its attention value is set to a constant value of either 60.0 when the user preferred Smokey to play the ball more than to do other tasks, or 50.0 when the user preferred Smokey to do other tasks more than to play the ball.
3. **The play_drums Ordinary Process**
   The play_drums process proposes an action when Smokey is not sleeping to either play, track, or search for the drums depending on whether the location of the drums is known and within reach, known but not within reach or unknown, respectively. Similar to the play_ball process, the attention value of the play_drums process is set to a constant value of either 60.0 when the user preferred Smokey to play the drums more than to do other tasks, or 50.0 when the user preferred Smokey to do other tasks more than to play the drums.

4. **The go_sleep Ordinary Process**
   The go_sleep process proposes an action to go to sleep and wake up in every defined period. Its attention value is linearly increased. It maintains its attention value when either its attention value reaches 100.0 or it wins an attention competition. It then resets back its attention value to 0.0 after Smokey has enough sleep (i.e., predefined time).
5. **The emotion_mechanism Supervisor Process**
   ASMO's emotion mechanism is implemented in a supervisor process called *emotion_mechanism*. This process proposes an action to update the attention values of the play_ball, play_drums and go_sleep processes by the amount of their subjective weights. It first uses a causal Bayesian network (see Fig. 3) to predict Smokey's happy feeling and the user's happy feeling (2). It will then determine the subjective weights of the play_ball, play_drums and go_sleep processes according to the expected desirability of those happy feelings (3). Its attention value is set to an arbitrary value. This attention value does not hold any significant meaning because the emotion_mechanism process does not require any resource, so it does not need to compete for attention against other processes.



Robot's Happiness	Person's Happiness	Positive Valence = True	Negative Valence = True	Arousal = True
Happy	Happy	0.95	0.05	0.9
Happy	Neutral	0.8	0.2	0.8
Neutral	Happy	0.8	0.2	0.8
Neutral	Neutral	0.05	0.5	0.5

Red Sensation	Praise	Happy
Experienced	Occur	0.95
Experienced	Not	0.8
Not	Occur	0.6
Not	Not	0.05

Praise	Happy
Occur	0.7
Not	0.3

Action	Experienced
Play_ball	0.8
Play_drums	0.2
No_play	0.05

Action	Occur
Play_ball	0.2
Play_drums	0.5
No_play	0.05

Play_ball	Play drums	No Play
0.3333	0.3333	0.3333

Fig. 3 Causal Bayesian network used in smokey

The emotions toward the red color and praise are modeled as the following nodes in a casual Bayesian Network (see Fig. 3):

1. **Action Node**
   The *action* node represents Smokey's actions that may affect its emotions.
2. **Red Sensation and Praise Nodes**
   The *red sensation* and *praise* nodes represent the biological and cognitive factors that cause emotions respectively.
3. **Robot's Happiness and User's Happiness Nodes**
   The *robot's happiness* and *user's happiness* nodes represent the happiness of Smokey and the happiness of the user respectively.
4. **Positive Valence, Negative Valence and Arousal Nodes**
   The *positive valence*, *negative valence* and *arousal* nodes represent the dimensions of emotions.

Table 1 shows the predictions of Smokey's happy and the user's happy feelings and also the subjective weights of the play_ball and play_drums processes for different probabilities of receiving praise. The higher the probability of receiving praise, the higher the subjective weight of the play_drums process. Both desirabilities of Smokey's happy and the user's happy are set to 20.0. Symbols used in the table are as described as follows:
$P(pr|pda)$ is the probability of receiving praise given Smokey plays the drums
$P(rh|pba)$ is the probability of Smokey is happy given Smokey plays the ball
$P(uh|pba)$ is the probability of the user is happy given Smokey plays the ball
$P(rh|pda)$ is the probability of Smokey is happy given Smokey plays the drums
$P(uh|pda)$ is the probability of the user is happy given Smokey plays the drums
$S_t(pb)$ is the subjective weight of the play_ball process at time t
$S_t(pd)$ is the subjective weight of the play_drums process at time t.

Figure 4 and 5 show the results of the two experiments without and with ASMO's emotion mechanism, respectively. In the experiment, the user's preferences of Smokey playing the ball and the drums were equal and Smokey received requests

Table 1 The subjective weights of the play_ball and play_drums processes

$P(pr\|pda)$	Play ball			Play drums		
	$P(rh\|pba)$	$P(uh\|pba)$	$S(pb)$	$P(rh\|pda)$	$P(uh\|pda)$	$S(pd)$
0.5	0.696	0.38	10.76	0.435	0.5	9.35
0.55	0.696	0.38	10.76	0.4585	0.52	9.785
0.6	0.696	0.38	10.76	0.482	0.54	10.22
0.65	0.696	0.38	10.76	0.5055	0.56	10.655
0.7	0.696	0.38	10.76	0.529	0.58	11.09
0.75	0.696	0.38	10.76	0.5525	0.6	11.525
0.8	0.696	0.38	10.76	0.576	0.62	11.96

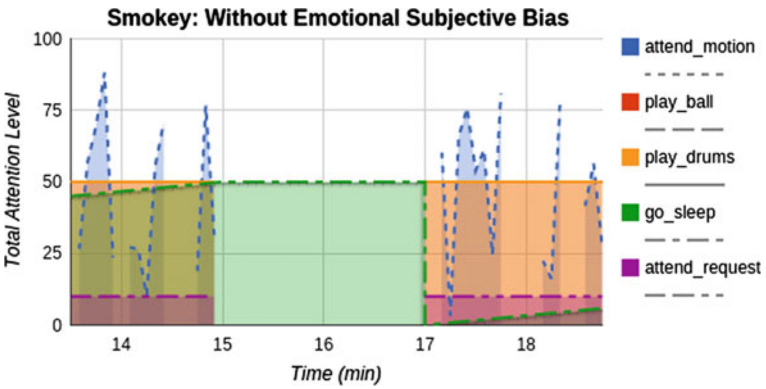


Fig. 4 Without emotional subjective bias

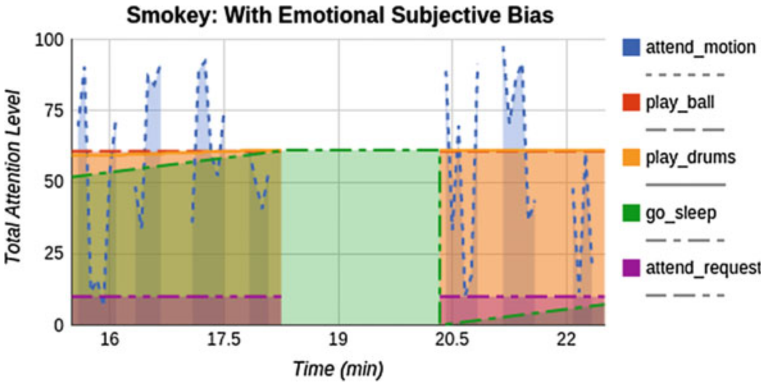


Fig. 5 With emotional subjective bias

to play drums. These requests to play drums increased the probability of receiving praise given that Smokey plays the drums (i.e., $P(pr|pda)$).

Without ASMO's emotion mechanism, both the play_ball and play_drums modules had an equal total attention level, since the user's preference is equal (see Fig. 4). Thus, Smokey relied on a conflict resolution strategy (e.g., random) to choose one among the two modules.

With ASMO's emotion mechanism, the play_ball and play_drums modules' total attention levels were increased by their subjective weights (see Fig. 5). The play_drum module's subjective weight was increased as the probability of receiving praise was increased. Over time, the total attention level of the play_drums module was slightly higher than the total attention level of the play_ball module. Thus, Smokey chose to play drums instead of playing ball when interacting with the user. In addition, Smokey's sleep time was shifted because the go_sleep module took a longer time to win the attention competition against the play_ball and play_drums modules.

5 Conclusion

This paper has demonstrated the capability of ASMO's emotion mechanism to predict emotions and bias behaviors in real-time in a physical robot. This mechanism allows opposite emotions (e.g., positive valence and negative valence) to elicit simultaneously. The final effects of the opposite emotions are determined based on the expected desirability of feelings.

ASMO's emotion mechanism can learn emotions from situations in the environments. It increases the probabilities of nodes given the observations from the environments. For example, it increases the probability of praise as the users requesting Smokey to play drums. This mechanism handles the uncertainty of actions, situations, and feelings using the Bayes rule.

References

1. Anderson, C.J.: The psychology of doing nothing: forms of decision avoidance result from reason and emotion. Psychol. Bull. **129**, 139–167 (2003)
2. Beck, A., Caamero, L., Bard, K.: Towards an affect space for robots to display emotional body language. In: RO-MAN, 2010 IEEE, pp. 464–469 (2010)
3. Breazeal, C.: Emotion and sociable humanoid robots. Int. J. Hum.-Comput. Stud. **59**(1–2), 119–155 (2003)
4. Conati, C., Maclaren, H.: Empirically building and evaluating a probabilistic model of user affect. User Model. User-Adapt. Interact. **19**(3), 267–303 (2009)
5. Ekman, P.: Universals and cultural differences in facial expressions of emotion. In: Nebraska Symposium on Motivation, vol. 19, pp. 207–283 (1971)
6. Hastie, R., Dawes, R.: Rational Choice in an Uncertain World: The Psychology of Judgment and Decision Making. Sage (2009)
7. Kim, H.R., Kwon, D.S.: Computational model of emotion generation for humanrobot interaction based on the cognitive appraisal theory. Intell. Robot. Syst. **60**(2), 263–283 (2010)
8. Kirby, R., Forlizzi, J., Simmons, R.: Affective social robots. Robot. Auton. Syst. **58**(3), 322–332 (2010)
9. Kishi, T., Kojima, T., Endo, N., Destephe, M., Otani, T., Jamone, L., Kryczka, P., Trovato, G., Hashimoto, K., Cosentino, S., Takanishi, A.: Impression survey of the emotion expression humanoid robot with mental model based dynamic emotions. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 1663–1668 (2013)
10. Loewenstein, G., Lerner, J.S.: The role of affect in decision making. In: D.R. et al. (eds.) Handbook of Affective Sciences, pp. 619–642. Oxford University Press, Oxford (2003)
11. Novianto, R.: Flexible attention-based cognitive architecture for robots. Ph.D. Dissertation, University of Technology Sydney, Sydney, Australia (2014)
12. Novianto, R., Williams, M.A., Grdenfors, P., Wightwick, G.: Classical conditioning in social robots. In: Social Robotics, Lecture Notes in Computer Science, vol. 8755, pp. 279–289. Springer International Publishing (2014)
13. Ortony, A., Clore, G.L., Collins, A.: The Cognitive Structure of Emotions. Cambridge University Press (1990)
14. Rosis, F., Castelfranchi, C., Goldie, P., Carofiglio, V.: Cognitive evaluations and intuitive appraisals: can emotion models handle them both? In: Cowie, R., Pelachaud, C., Petta, P. (eds.) Emotion-Oriented Systems. Series in Cognitive Technologies, pp. 459–481. Springer, Berlin (2011)

15. Rumbell, T., Barnden, J., Denham, S., Wennekers, T.: Emotions in autonomous agents: comparative analysis of mechanisms and functions. Auton. Agents Multi-Agent Syst. 1–45 (2011)
16. Sawada, T., Takagi, T., Fujita, M.: Behavior selection and motion modulation in emotionally grounded architecture for qrio sdr-4xii. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings, vol. 3, pp. 2514–2519 (2004)
17. Simon, H.A.: Making management decisions: the role of intuition and emotion. Acad. Manag. Exec. (1987–1989) **1**(1), 57–64 (1987)
18. Zhong, J., Canamero, L.: From continuous affective space to continuous expression space: nonverbal behaviour recognition and generation. In: 2014 Joint IEEE International Conferences on Development and Learning and Epigenetic Robotics (ICDL-Epirob), pp. 75–80 (2014)

Multi-robot Task Allocation Using Clustering Method

Farzam Janati, Farzaneh Abdollahi, Saeed Shiry Ghidary, Masoumeh Jannatifar, Jacky Baltes and Soroush Sadeghnejad

Abstract This paper introduces an approach to solve the task assignment problem for a large number of tasks and robots in an efficient time. This method reduces the size of the state space explored by partitioning the tasks to the number of robotic agents. The proposed method is divided into three stages: first the tasks are partitioned to the number of robots, then robots are being assigned to the clusters optimally, and finally a task assignment algorithm is executed individually at each cluster. Two methods are adopted to solve the task assignment at each cluster, a genetic algorithm and an imitation learning algorithm. To verify the performance of the proposed approach, several numerical simulations are performed. Our empirical evaluation shows that clustering leads to great savings in runtime (up to a factor of 50), while maintaining the quality of the solution.

F. Janati (✉) · M. Jannatifar · S. Sadeghnejad
Robotics Institute, AmirKabir University of Technology,
424 Hafez Ave, Tehran, Iran
e-mail: farzam.janati@aut.ac.ir
URL: http://arc.aut.ac.ir/en/

M. Jannatifar
e-mail: m.jannatifar@aut.ac.ir

S. Sadeghnejad
e-mail: s.sadeghnejad@aut.ac.ir

F. Abdollahi
Department of Electrical Engineering, AmirKabir University of Technology,
424 Hafez Ave, Tehran, Iran
e-mail: f_abdollahi@aut.ac.ir

S.S. Ghidary
Department of Computer Engineering and Information Technology,
AmirKabir University of Technology, 424 Hafez Ave, Tehran, Iran
e-mail: shiry@aut.ac.ir

J. Baltes
University of Manitoba, Winnipeg, MB R3T 2N2, Canada
e-mail: jacky@cs.umanitoba.ca

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_19

Keywords Multi-agent · Task assignment · Clustering · Genetic algorithm · Imitation learning

1 Introduction

Cooperation between groups of robots improves the performance of the mission and enables the agents to accomplish a goal they are not able to do individually. The objective of cooperation is to ensure that every single decision, made by agents, results in optimal decisions for the whole team. A multi-agent system consists of multiple independent agents interacting together. Each agent, in the multi-agent system (MAS), makes decisions and acts autonomously, based on its observations and also shared information of other agents, in order to reach a joint goal in the mission. MAS has some potential advantages over single robot systems, MAS can improve performance, scalability, and robustness of a mission by parallelizing actions [1], Also it can distribute the computational resources through the group of agents in decentralized systems. The fields of application of MAS have a wide range of applications, such as autonomous surveillance, reconnaissance, and exploration missions [2–4].

Task allocation and path planning are necessary for efficient operation of multiple robots. The goal of task allocation is to find a match between agents and tasks that maximizes the overall utility of the team. The optimization criteria vary due to purpose of the mission. Some of them minimizing the time of accomplishing the mission, minimizing the overall path of robots, or minimizing the energy consumption of all robots.

Task allocation for robots can be formulated as a multiple traveling salesman problem (MTSP), which is a challenging problem and considered as a NP-hard combinatorial optimization problem. Therefore there is not a specific method for finding an optimal solution. Many approaches have been developed to overcome the complexity and find optimal solutions. One of the approaches is Mixed Integer Linear Programming (MILP), which is a mathematical programming method [5–7]. Although solutions provided by MILP are acceptable, it is computationally expensive. Another approach is meta-heuristic algorithms such as Genetic Algorithm (GA) [8–10] and particle swarm optimization (PSO) [11]. GA and PSO are generic methods for finding suboptimal solutions. The meta-heuristic approaches obtain solutions quickly, however the quality of solutions might be poor, and also they might easily become intractable for large-sized problems. Another promising solution for solving the task allocation problem is introduced by learning algorithms. These approaches use a set of examples and extract the policy and build a model to determine optimal solution in test conditions. Imitation learning is one category of learning algorithms that benefits from human expert guidance thorough learning process to extract policies. These approaches are also time-consuming in large-scaled problems.

Increasing the number of tasks and robots expands the size of the state space dramatically and affect the performance of approaches. Since this process requires high computational time, this paper describes a method that reduces the size of the state space explored, by partitioning the tasks to the number of robotic agents. Therefore, the problem becomes task allocation for a single agent for each cluster, thus the computational cost will be reduced. Here, a k-means method [12] is used to partition the tasks. K-means aims to partition the input data to specific number of clusters. This paper will focus on large-scale assignment problems involving hundreds of robots and tasks. First, the tasks are partitioned to the number of robots, then robots are being assigned to the clusters optimally, and finally a task assignment algorithm is executed individually and in parallel in the clusters.

The paper has been organized as follows. The second section of this paper will describe the formulation of multiple task assignment problem. In Sect. 3, task allocation using clustering method is described. In Sect. 4, a non-clustering genetic algorithm for multiple task assignment problem is proposed. In Sect. 5, an imitation learning algorithm, maximum margin planning (MMP), and clustering MMP for task allocation problem are explained. Sections 6 and 7 provide numerical simulations and a comparison between the GA, MMP, and the proposed algorithms with clustering method. Finally, Sect. 8 concludes the paper.

2 Problem Formulation

In this section, a mathematical formulation of the multiple task allocation problem is presented. Let N_a be the total number of the robots and A be the agents set with known positions that can be defined as follows:

$$A = \left\{A_1, A_2, \dots, A_{N_a}\right\}. \tag{1}$$

The tasks set T are defined as follows:

$$T = \left\{T_1, T_2, \dots, T_{N_t}\right\}. \tag{2}$$

where N_t is the number of the tasks. The task allocation formulation can be expressed as follows:

$$\min \sum_{i=1}^{N_a} \left(\sum_{j=1}^{N_t} c_{ij} u_{ij} \right). \tag{3}$$

Subject to:

$$\sum_{i=1}^{N_a} u_{ij} = 1, \quad \forall j \in \left\{1, 2, \dots, N_t\right\}. \tag{4}$$

$$\sum_{j=1}^{N_t} u_{ij} \leq L_t, \quad \forall i \in \{1, 2, \ldots, N_a\}. \tag{5}$$

where decision variable $u_{ij} = 1$, if agent i is assigned to task j and 0 otherwise, c is the cost (distance) matrix, and L_t is the maximum number of tasks that can be assigned to each agent. The objective of a task allocation is to find the best conflict-free match between N_a agents and N_t tasks. Task allocation is conflict free if each task is assigned to just one agent. The cost function often represents a path dependent reward and the goal is to minimize the total path of robots. In addition, path independent cost functions often minimize a user-defined function by determining the priority of tasks. The task allocation is done when all of the tasks are assigned to agents.

3 Task Allocation Using Clustering Method

3.1 Clustering

The goal of clustering is to divide an input data into a finite discrete set of structures. Clustering algorithms partition the input data into a specific number of clusters. This paper describes a method that reduces the size of the state space explored, by partitioning the tasks to the number of robotic agents. Hence the main problem is divided into some simple problems, such that the problem of task allocation for multiple agents is divides into several task allocation problems, each for a robotic agent in a cluster. Hence, the computational complexity is reduced by decreasing the size of the state space. There are many methods for data clustering, such as hierarchical methods, partitioning methods, density-based methods, model-based clustering methods, grid-based methods, and soft-computing methods [13]. In this paper a partitioning method called K-means is used [12]. K-means clustering is a simple unsupervised learning algorithm for clustering analysis. The algorithm aims to partition N_t points into N_a groups in order to minimize the total distance between the centroids of each cluster and their corresponding points. The objective function defines as follows:

$$\arg\min_{S} \sum_{i=1}^{N_a} \sum_{x \in S_i} \|x - \mu_i\|^2. \tag{6}$$

where $S = \{S_1, S_2, \ldots, S_{N_a}\}$ is the set of clusters, $x = \{x_1, x_2, \ldots, x_{N_t}\}$ is the set of input data and μ_i is the center in cluster S_i. A set of initial cluster centers is chosen randomly, then, in each iteration, each point is assigned to its nearest cluster center, finally, the cluster centers are calculated again.

3.2 Assigning Robots to Clusters

After partitioning the tasks, the robots should be assigned to clusters in an optimum way, based on the distance from robots to clusters. The distance from a robot to a cluster is defined based on the robot and the nearest task of the cluster to it. The distance matrix is a $N_a \times N_a$ matrix, that N_a is the number of robots in the mission. Now, the optimum way to assign robots to the clusters should be computed. This problem is a linear sum assignment problem (LSAP). In LSAP, each row has to be matched to a different column in such a way that sum of the corresponding entries is minimized.

Many approaches has been developed to solve LSAP problem, the algorithms for LSAP are based on different approaches: first class of methods directly solves the primal problem, second one solves the dual, and third one uses an intermediate approach (primal–dual) [14]. Hungarian method [15] is the first polynomial-time primal–dual algorithm for solving the LSAP. The time complexity of the original Hungarian method is $O(n^4)$. Here, we use the Jonker–Volgenant algorithm [16] to solve this problem, which is a shortest path implementation for the Hungarian algorithm with higher performance. This method is a Dijkstra [17]-like algorithm for shortest path augmentation which is done after three levels of preprocessing: column reduction, reduction transfer, and augmenting row reduction. These steps are the main contribution and the most time-consuming part of this method. The Dijkstra-like algorithm for shortest path augmentation grows an alternating tree for an unassigned cluster, to find alternating paths including possibly augmenting path starting from that cluster [14]. A special implementation of Dijkstra and a total time complexity of $O(n^3)$ makes this method suitable for fast assignment of robots to the clusters.

3.3 Clustering Genetic Algorithm

Genetic algorithm (GA) mimics the process of natural selection and an iteration evolutionary process for finding optimal solutions. The set of feasible solutions known as chromosome in the GA represents as follows. Each chromosome with the length of the number of tasks in the cluster expressed as C, and C_i is the ith element of C representing the ith task on the schedule of the agent. To generate a new solution candidate, GA operators such as selection, crossover, mutation, and replacement are executed. The tournament selection method is adopted as selection operator. Two chromosomes from the population are chosen with the selection method as the input for crossover operator. Partially Matched Crossover (PMX) [18] method is adopted as crossover operator. The PMX method is widely used for permutation chromosomes. Randomly two non-equal points between the zero and length of the chromosome are selected, a sub-string between the two random points of the parent chromosome is donated to the offspring at the same position, and then it affects cross by position-by-position exchange operations. The mutation crossover consists of swap, reversion,

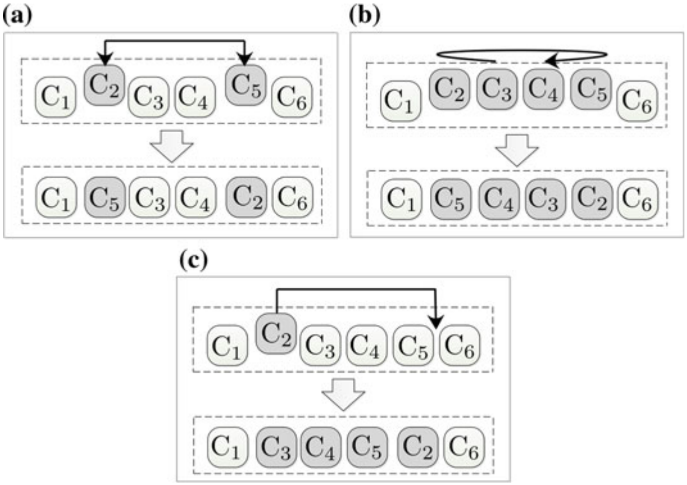


Fig. 1 Mutation operators: **a** swap operator, **b** reversion operator, **c** insertion operator

and insertion. The swap operator exchanges the value of two random points in the chromosome, the reversion operator reverses a swath of the chromosome, and insertion operator transmits the value of a random point to another point in chromosome. Figure 1 explains the mutation operators. The algorithm has been executed several times for different number of tasks to find the iteration number when there is no more change in the cost. A polynomial is fitted to these points to find the termination criteria for further executions of the algorithm with any arbitrary number of tasks.

4 Non-clustering Genetic Algorithm for Multiple Task Allocation Problem

GA is executed to find the suboptimal solution. The set of feasible solutions known as chromosome in the GA represents as follows. Each chromosome expressed as C, which consists of two parts, the first part expressed as CA with the length of N_a, represents the number of tasks for each agent, the ith element of CA expressed as CA_i is the amount of tasks for agent i. The second part expressed as CT with the length of N_t is a permutation of doing tasks for agents. CT is made from concatenation of different sequences, each with length of CA_i, $i = 1, 2, \ldots, N_a$, representing a permutation of allocated tasks to the agent i. Each of these genes has a value representing task number CT_j, $j = 1, 2, \ldots, N_t$. The structure of chromosome is shown in Fig. 2. Two parts of the chromosome are being separated from each other to perform crossover and mutation. Crossover and mutation operators decide to perform action only on one part at each time with a constant probability. GA operators, such as selection, crossover, mutation and replacement are exactly the same with the previous section.

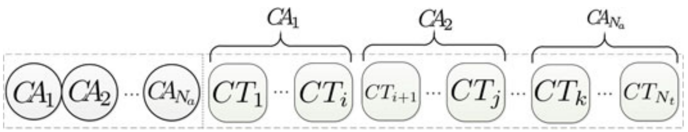


Fig. 2 Structure of the chromosome

Since the sum of elements of the first part is constant and equal to the number of tasks, a little change has been done to the crossover operator to keep the sum of elements constant.

5 Imitation Learning

Learning a mapping between a problem's world states and actions is the most important problem in many robotic applications. This mapping which is also called a policy enables a robot to select an action based upon its current world state. With imitation learning, policy is extracted from a set of demonstrations directly provided by a human or through human guidance [19]. Using imitation learning to extract policies from a set of demonstrated allocations is a powerful approach for reducing the complexity of search spaces for learning by eliminating impossible solutions [20].

In multi-robot task allocation domains, explicitly modeling task features well enough so that they are used in allocation process may be intractable, but a human expert may be able to quickly gain some knowledge about the form of the desired solution [21]. In the next section one of the imitation learning frameworks, Maximum Margin Planning is discussed.

5.1 Maximum Margin Planning for Multiple Task Allocation Problem

Maximum Margin Planning (MMP) is an imitation learning approach for structured prediction over the space of policies [22]. μ represents a particular set of allocations from the space of possible allocations G. $f \in \mathbb{R}^d$ is feature vector for each possible task allocation and F is an accumulation matrix of these features. The product $F\mu$ represents the accumulation of features encountered by following the policy μ [22]. Training data set is specified as $D = \left\{\left(F_i, G_i, \mu_i, l_i\right)\right\}_i^N$. Each training example i consists of a set of tasks, agents, and the expert allocation $\mu_i \in G_i$. Each example includes a loss field l_i, so that $l_i^T \mu$ quantifies how bad a policy μ is compared to the demonstrated policy μ_i. The goal is then to learn a set of weights ω, so that each demonstrated allocation is better than every other possible allocation for the same scenario [21]:

$$\omega^T F_i \mu_i > \omega^T F_i \mu + l_i^T \mu \qquad \forall i, \mu \in G. \tag{7}$$

If constraint (7) holds for all allocations in G, it must hold for the best allocation, thus the only constraint to consider is the tightest one, corresponding [21]:

$$\mu_i^* = \max_{\mu \in G_i} \left[\left(\left(\omega^T F_i + l_i^T \right) \mu \right) \right]. \tag{8}$$

Solving this problem using subgradient method [21]:

$$g_\omega = \lambda \omega + \frac{1}{N} \sum_{i=1}^{N} \left(F_i \mu_i^* - F_i \mu_i \right) = \lambda \omega + \frac{1}{N} \sum_{i=1}^{N} \left(F_i \Delta^\omega \mu_i \right). \tag{9}$$

Subgradient update rule then becomes [21]:

$$\omega_{t+1} = \omega_t - \alpha g_\omega. \tag{10}$$

For learning rate α,intuitively, this gradient update rule increases the reward (in feature space) on the demonstrated allocation and decreases the reward (again in feature space) on the chosen allocation. In order to use demonstrated allocations in task allocation mechanism, a bias term is introduced into task execution order, which uses the learned feature weighting vector w. Therefore the profit used for selecting task for execution is [21]

$$profit = cost + bias. \tag{11}$$

cost is a constant value, added to profit and the bias:

$$bias = w^t f^k. \tag{12}$$

f_k is features vector for task k.

5.2 Clustering Maximum Margin Planning

In order to use clustering advantages in imitation learning for multi-robot task allocation, at first, tasks are clustered with K-means algorithm, and then each robot starts to execute own clusters tasks according to learned policy with maximum margin planning. Tasks execution priority is chosen in order to maximize the total profit, calculated in Eq. (11).

6 Numerical Results for GA

In this section, a comparison including numerical results between clustering GA and the non-clustering GA method for solving multiple task allocation problem is represented. Simulations for both methods are performed in several cases. As explained in Table 1, in each case different number of tasks and agents are included. Coordinates

Table 1 Conditions of numerical solutions

	Case 1	Case 2	Case 3	Case 4	Case 5
Number of tasks	20	50	100	200	200
Number of agents	4	8	10	15	25

Table 2 Numerical results of task assignment for clustering GA

	Case 1	Case 2	Case 3	Case 4	Case 5
Clustering time (s)	0.141	0.156	0.156	0.157	0.157
Assigning time (s)	0.016	0.016	0.016	0.016	0.017
Algorithm time (s)	0.864	2.219	5.365	12.344	8.406
Total cost	1814.5	2414	3951.6	5607	5230.9

Table 3 Numerical results of task assignment for non-clustering GA

	Case 1	Case 2	Case 3	Case 4	Case 5
Algorithm time (s)	1.812	20.531	87.812	226.250	132.078
Total cost	1814.5	2414	3951.6	5607	5230.9

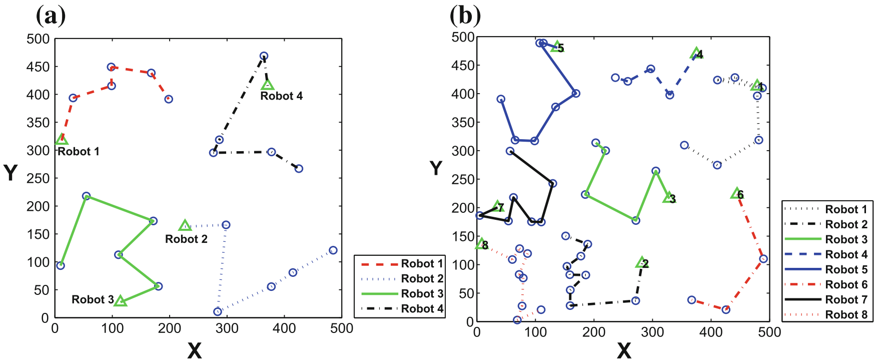


Fig. 3 Final solutions for task allocation with clustering GA for **a** case 1 and **b** case 2; the *circles* indicates the coordinates of tasks and the *triangles* indicates the initial coordinates of robots

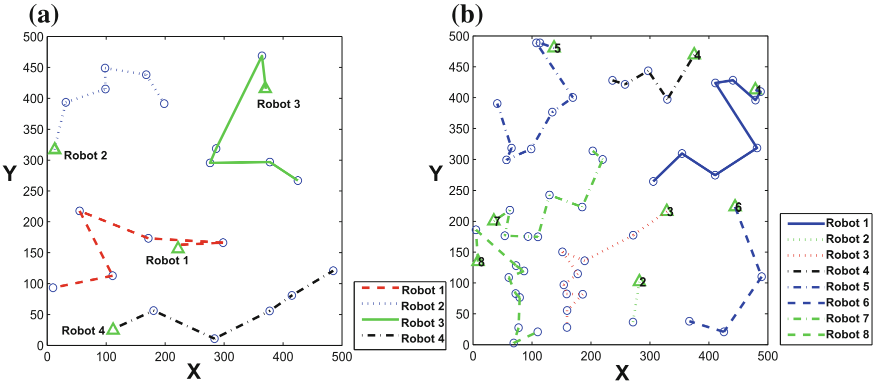


Fig. 4 Final solutions for task allocation with non-clustering GA for **a** case 1 and **b** case 2; the *circles* indicates the coordinates of tasks and the *triangles* indicates the initial coordinates of robots

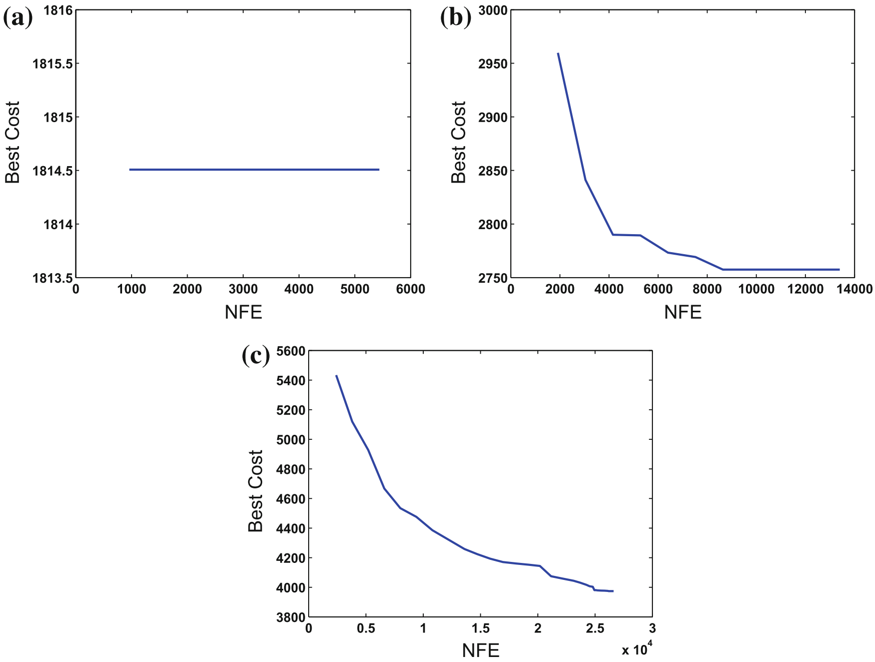


Fig. 5 Total cost of the mission with respect to NFE for clustering GA for **a** case 1, **b** case 2 and **c** case 3

of tasks and agents are generated randomly, the same coordinates are used for both algorithms. The simulations were performed on a laptop computer which has a dual core 2.0 GHz CPU and 6 GB RAM using MATLAB. The MATLAB parallel computing toolbox is used to execute the algorithm for all clusters in parallel. The

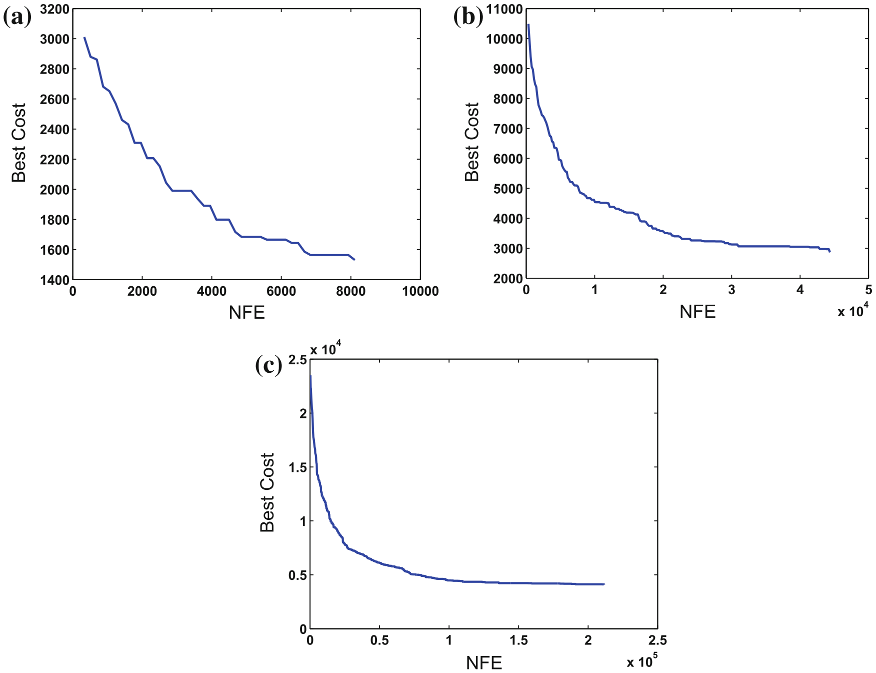


Fig. 6 Total cost of the mission with respect to NFE for non-clustering GA for **a** case 1, **b** case 2 and **c** case 3

termination criterion for the proposed method is explained in Sect. 5. The non-clustering GA method is executed until it reaches the total cost of our method for each case. Total cost is the total distance of the path of all robots. In Table 2, process time of clustering, process time of assigning robots to the clusters, computation time of the algorithm, and the total cost of mission are represented. Table 3 shows the cost and process time of the assignment problem with simple GA. The final solutions of all cases for our proposed method and the GA without clustering are shown in Figs. 3 and 4, also Figs. 5 and 6 represents the total cost of the mission with respect to the number of fitness evaluation (NFE) for all cases. The algorithm process times for clustering GA and non-clustering GA with different number of agents and tasks are illustrated in Figs. 7 and 8. According to the numerical results in all cases, our proposed method produced solutions very fast, whereas the simple GA took a lot of time to reach the same total cost, and this time increases exponentially when the number of tasks increases.

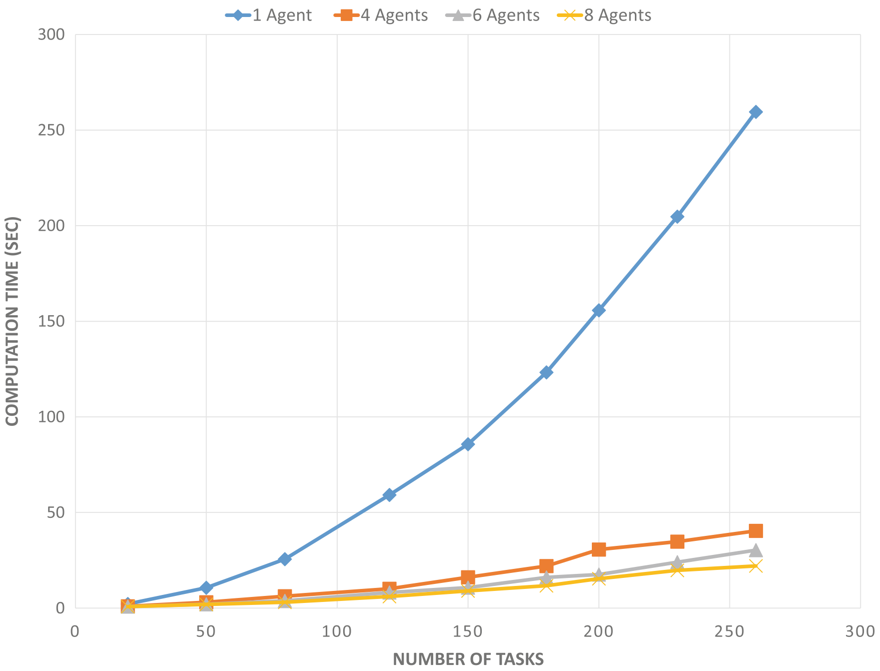


Fig. 7 Computation time for clustering GA with different number of agents and tasks

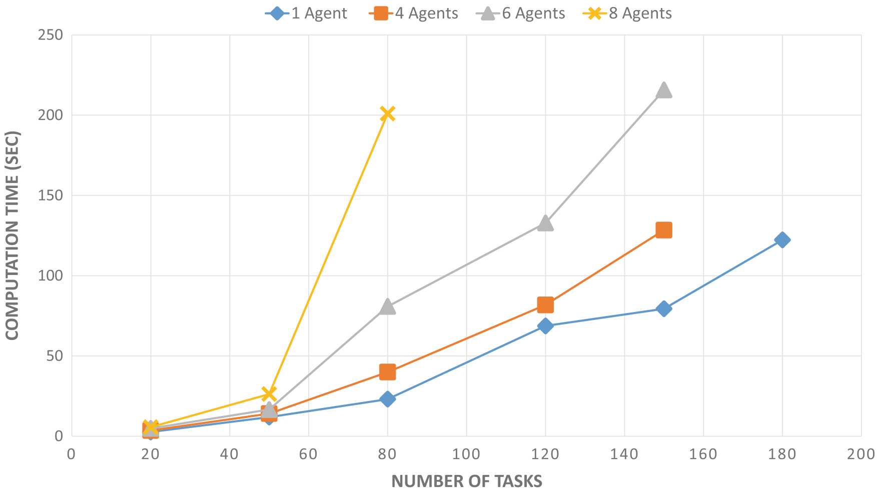


Fig. 8 Computation time for non-clustering GA with different number of agents and tasks

7 Numerical Results for Imitation Learning

In this section, a comparison between MMP and the MMP with clustering for solving multiple task allocation problem is represented. A scenario for task allocation for buildings in fire [21] is represented and will solved with imitation learning. Each task has a feature vector that express information about each individual task in the environment. Here features that are computed per task, consists of 2 elements: (1) Size: building size represented with size of task square in figures (2) Health: current building health represented in percent in figures. Human experts in example demonstrations, represent optimal task allocations, for example in this scenario tasks closer to robot with bigger building size and healthier buildings are prior to others. Using these training set, policy is extracted and used in future test conditions.

Experimental results for all cases in Table 1 are represented. Tables 4 and 5 show the average profit of all robots profit and total time to finish all robots tasks. The

Table 4 Total time to finish the mission

	Case 1	Case 2	Case 3	Case 4	Case 5
Total time of non-clustering MMP	10.122	15.435	34.1094	43.1214	53.078
Total time of clustering MMP	0.3188	0.4188	0.6188	0.8188	0.9324

Table 5 Average profit of all robots

	Case 1	Case 2	Case 3	Case 4	Case 5
Average profit for non-clustering MMP	348.09	352.22	358	362.07	365.87
Average profit for clustering MMP	348	352.01	358.01	361.07	365.23

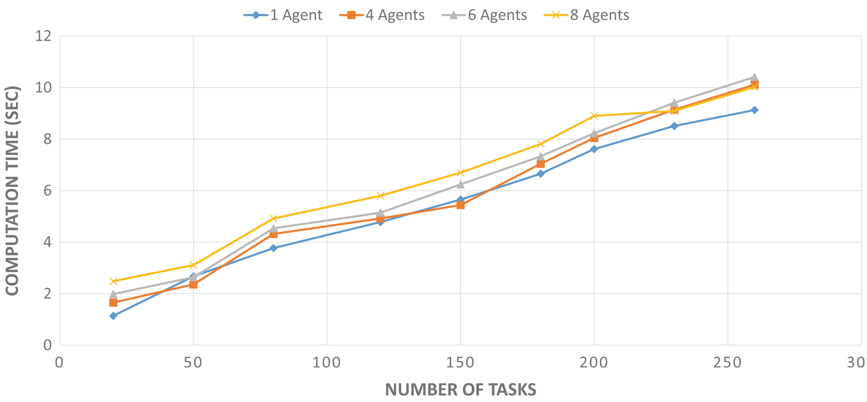


Fig. 9 Computation time for clustering imitation algorithm with different number of agents and tasks; because each robot executes each clusters tasks in parallel with other robots, MMP with clustering overcome the algorithm without clustering in total algorithm time in all cases

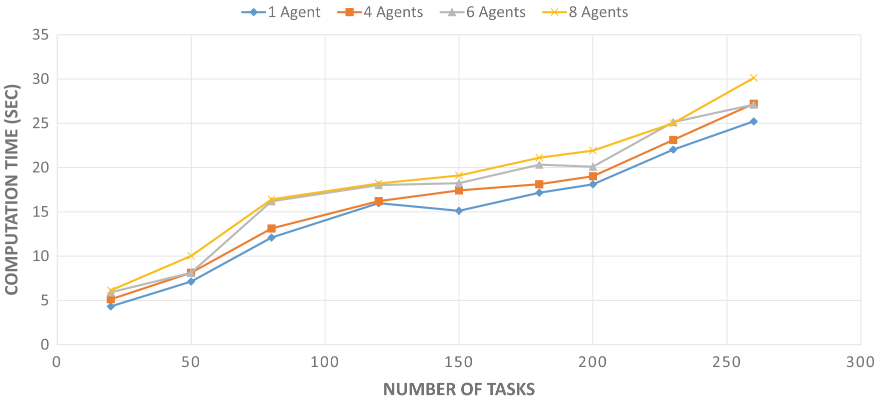


Fig. 10 Computation time for non-clustering imitation algorithm with different number of agents and tasks; approximately in all cases algorithm time increases about 50

algorithm process times for clustering and non-clustering imitation algorithm with different number of agents and tasks are illustrated in Figs. 9 and 10. As the results show, maximum margin planning with clustering overcome the algorithm without clustering in total algorithm time. Total cost and solutions found for both algorithms are nearly the same. Clustering time and assigning time are identical to GA with clustering.

8 Conclusion

In this paper, a task assignment method is presented to deal with large number of tasks and robots. The proposed method is able to assign a large number of tasks to robots in a high efficient time. The effectiveness of our approach is demonstrated by numerical simulations. Especially in large-scaled task allocations, using the simple GA method to reach the final cost similar to our proposed method, takes longer time.

References

1. Wu, F., Zilberstein, S., Chen, X.: Online planning for multi-agent systems with bounded communication. Artif. Intell. **175**(2), 487–511 (2011)
2. Avellar, G.S., Thums, G.D., Lima, R.R., Iscold, P., Torres, L. Pereira, G., et al.: On the development of a small hand-held multi-uav platform for surveillance and monitoring. In: 2013 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 405–412, IEEE (2013)
3. Iscold, P., Pereira, G.A., Torres, L.A.: Development of a hand-launched small uav for ground reconnaissance. IEEE Trans. Aerosp. Electron. Syst. **46**(1), 335–348 (2010)
4. Burgard, W., Moors, M., Stachniss, C., Schneider, F.E.: Coordinated multi-robot exploration. IEEE Trans. Robot. **21**(3), 376–386 (2005)

5. Chandler, P.R., Pachter, M., Rasmussen, S., Schumacher, C.: Multiple task assignment for a uav team. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, p. 4587 (2002)
6. Richards, A., Bellingham, J., Tillerson, M., How, J.: Coordination and control of multiple uavs. In: AIAA Guidance, Navigation, and Control Conference, Monterey, CA (2002)
7. Schumacher, C., Chandler, P., Pachter, M., Pachter, L.: Constrained optimization for uav task assignment. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, pp. 1–14. American Institute of Aeronautics and Astronautics Inc, Reston, VA, USA (2004)
8. Eun, Y., Bang, H.: Cooperative task assignment/path planning of multiple unmanned aerial vehicles using genetic algorithm. J. Aircraft **46**(1), 338–343 (2009)
9. Shima, T., Rasmussen, S.J., Sparks, A.G., Passino, K.M.: Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms. Comput. Oper. Res. **33**(11), 3252–3269 (2006)
10. Potvin, J.-Y.: Genetic algorithms for the traveling salesman problem. Ann. Oper. Res. **63**(3), 337–370 (1996)
11. Cruz Jr, J.B., Chen, G., Li, D., Wang, X.: Particle swarm optimization for resource allocation in uav cooperative control. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, pp. 1–11, Providence, USA (2004)
12. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297, Oakland, CA, USA (1967)
13. Rokach, L.: A survey of clustering algorithms. In: Data Mining and Knowledge Discovery Handbook, pp. 269–298. Springer (2010)
14. Burkard, R.E., Dell'Amico, M., Martello, S.: Assignment Problems. Revised Reprint, Siam (2009)
15. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Res. Logist. Quart. **2**(1–2), 83–97 (1955)
16. Jonker, R., Volgenant, A.: A shortest augmenting path algorithm for dense and sparse linear assignment problems. Computing **38**(4), 325–340 (1987)
17. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**(1), 269–271 (1959)
18. Goldberg, D.E., Lingle, R.: Alleles, loci, and the traveling salesman problem. In: Proceedings of the First International Conference on Genetic Algorithms and Their Applications, pp. 154–159. Lawrence Erlbaum Associates, Publishers (1985)
19. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robot. Auton. Syst. **57**(5), 469–483 (2009)
20. Billard, A., Calinon, S., Dillmann, R., Schaal, S.: Robot programming by demonstration. In: Springer Handbook of Robotics, pp. 1371–1394. Springer (2008)
21. Duvallet, F., Stentz, A.: Imitation learning for task allocation. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3568–3573. IEEE (2010)
22. Ratliff, N.D., Silver, D., Bagnell, J.A.: Learning to search: Functional gradient techniques for imitation learning. Auton. Robots **27**(1), 25–53 (2009)

Modeling of Violin Playing Robot Arm with MATLAB/SIMULINK

Jargalbaatar Yura, Mandakh Oyun-Erdene, Bat-Erdene Byambasuren and Donghan Kim

Abstract In this research, we consider modeling of violin playing robot arm. MATLAB/SIMULINK are used for modeling of robot arm with seven degrees of freedom is considered, which is flexible than previously used robot arms with 6 joint for violin playing. Also, previous robot system which used Mitsubishi industrial robot arm (RV-2SD) is updated. In this model, torque, current consumption and voltage of each joint can be measured. Dynamixel-Pro from Robotis Co., Ltd. is used for the joints. This robot arm with seven joint has same range of movement (RoM) with human. This makes our designed robot good at violin playing. This paper presents basic violin playing technique, 3D modeling using Solidworks software, PID control system of servo motor using MATLAB/SimMechanics tool, and physical system of servo motor.

Keywords Violin playing robot • Robot arm • Servo motor • PID control • Human–robot interaction

J. Yura (✉) · D. Kim
Department of Electronics and Radio Engineering,
Kyung Hee University, Seoul, Korea
e-mail: jargalbaatar@khu.ac.kr

D. Kim
e-mail: donghani@khu.ac.kr

M. Oyun-Erdene · B.-E. Byambasuren
Department of Electric Technique, School of Power Engineering,
Mongolian University of Science and Technology, Ulaanbaatar, Mongolia
e-mail: mandakh@must.edu.mn

B.-E. Byambasuren
e-mail: baterdene@must.edu.mn

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_20

1 Introduction

Due to the rapid development of industrial robots, the robotics research area has widened to individual and household robots [1]. According to this robotics technology, a human–robot interaction (HRI) has come into the spotlight. An entertainment robot that is capable of delivering emotional expression and human-friendly mutual information is also developed. A musical instrument playing robot is mainly developed through the university. Waseda University has developed 'Waseda Saxophonist Robot' [2] and 'Flutist Robot' [3]. As an example, wind instrument is played by controlling the flow of artificial air using the air pump. Using the advanced control technology, Toyota has developed a violin playing robot [4] and shown the actual demonstration. In addition, Ryukoku University has developed a violin playing robot using emotions [5]. Kyung Hee University has modeled the bow of violin robot as a spring-damper model, which leads improving the sound quality based on the Q-score evaluation of played sound [6, 7].

A violin playing technique and its system is presented in this paper. In order to solve problems of previous violinist robot, a new 3D model of robot arm with 7 degrees of freedom is considered. There is used Solidworks software. Furthermore, *.XML file and graphic format file are created. Thus, MATLAB/Simscape block can be created by 7 degrees of freedom. In this system of robot arm, loads on each joint of motor angle are tested.

2 Violin Playing Robot

2.1 Violin Playing Robot System

A violin playing robot uses an industrial vertical multi-joint robot arm as shown in Fig. 1. The robot arm has 6 degrees of freedom and is controlled using TCP/IP communication. It is made up of AC servo motors, where the maximum load is 3 kg and the maximum speed is 4,400 mm/sec. To play a violin, the bow is attached at the end effector of the robot arm. The two-axis load cell is mounted on the bow handle to measure the horizontal and vertical force. As shown in Fig. 1, the violin is fixed to the robot body. For an accurate analysis of the played sound, an electrical violin that is resistant to external noise is used.

2.2 Violin Playing Technique

Violin makes a sound by exporting the string vibration into an air. Typically there are three methods of playing violin—plucking, bowing, and striking. Among then,

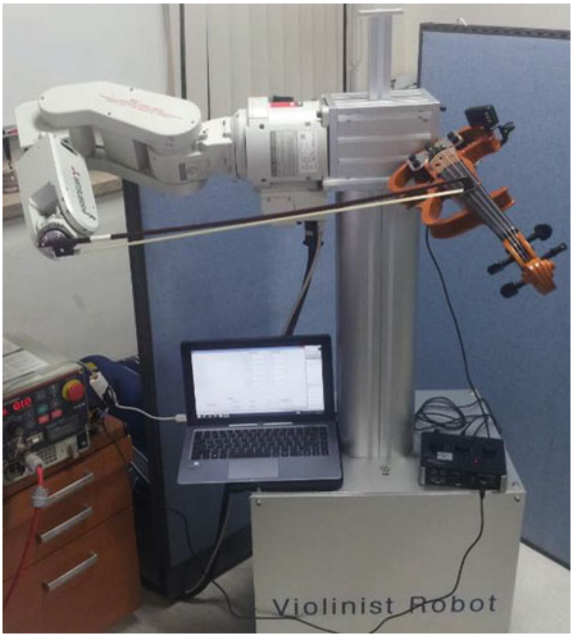


Fig. 1 The snapshot of violin playing robot [6]

the bowing method is the most common technique, which uses a bow. The bow is used to play four strings, where each string has a unique frequency. The sound is generated as the strings of violin are bowed and vibrated.

Figure 2 shows the three important factors when playing violin—bowing force (F), bowing velocity (V), and sound point (P) [8]. Note that the sound point is the distance between the violin bridge and the bow location. As shown in Fig. 2, the starting position of bow, X_S, is the tip of bow, whereas the last position, X_E, is the bow handle.

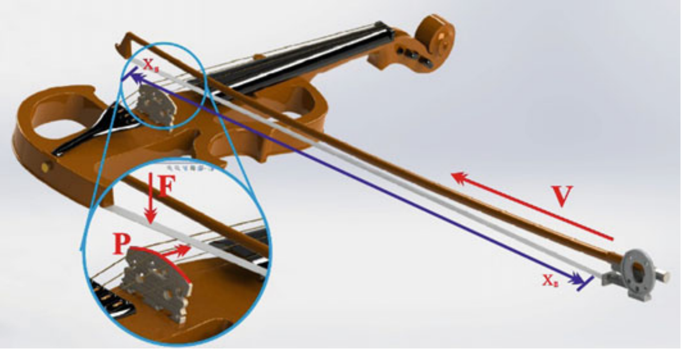


Fig. 2 Three important factors when playing violin—bowing force (F), bowing velocity (V), and sound point (P)

3 Requirement of New Violin Playing Robot Arm

Some researchers used Mitsubishi industrial robot arm (RV-2SD) for violin playing [6]. It is shown in Fig. 1. However, this industrial robot arm cannot measure torque, voltage, and current of each joint. In addition, it is not flexible compared with human arm. From these reasons, there is considered two types servo motor of Dynamixel-Pro, "Robotis" Co., Ltd. This servo motor specification is described as follows: operation voltage is 24 V; speed with no load is 33.1 RPM; current with no load is 1.65 A; continuous operation speed is 27.9 RPM; torque is 44.2 N.m; current is 9.3 A; resolution is 501900 steps per 1 full rotation; backlash is 3.5 arcmin; and communication interface is RS485 [9]. This type servo motors are used in new robot arm with 7 joints. Advantages of this type robot arm are follows:

1. Rigid joint length can be changed by optional value in this model.
2. There can be measured current consumption and voltage of each joint.
3. Torque of each joint can be determined.

Robot arm with 7 joints can move with human arm. Therefore, violin playing technique will be increased. 3D model of the robot arm with 7 joints is implemented in Solidworks software. 3D model is shown in Fig. 3. Physical model of servo motor and its PID control block of each joint are implemented using MATLAB/SimMechanics tools by Simscape block. Figures 8 and 9 show this physical model and PID control.

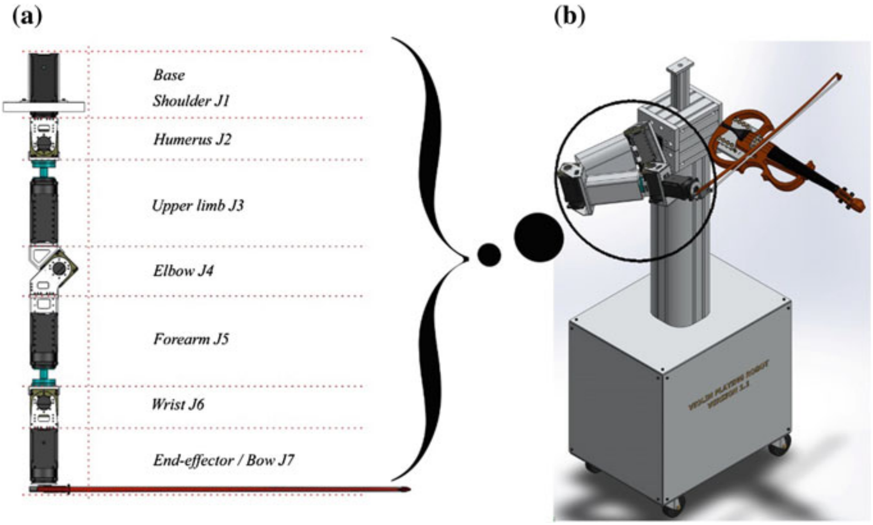


Fig. 3 **a** Robot arm with 7 joints. **b** Violinist robot model

4 Anthropomorphic Robot Arm with Seven Degrees of Freedom

4.1 Solidworks Model

A three-dimensional solid model of violin includes robot's body, 7 joints. In addition, XML file, STL file, and Simscape block of three-dimensional solid model are exported from Solidworks by using SimMechanics tools.

Simscape blocks, just as you do not have to specify this information when you connect real physical components. The Physical Network approach, with it is through and across variables and nondirectional physical connection, automatically resolves all the traditional issues with variables, directionality, and so on.

Figure 4 shows three-dimensional solid model of seven-axis robot arm in Simulink.

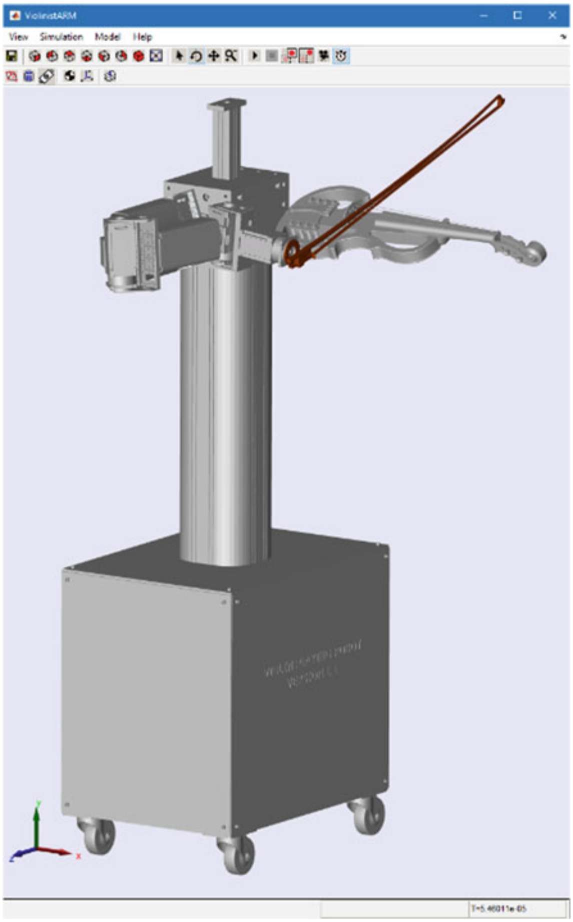


Fig. 4 Three-dimensional solid model in Simulink

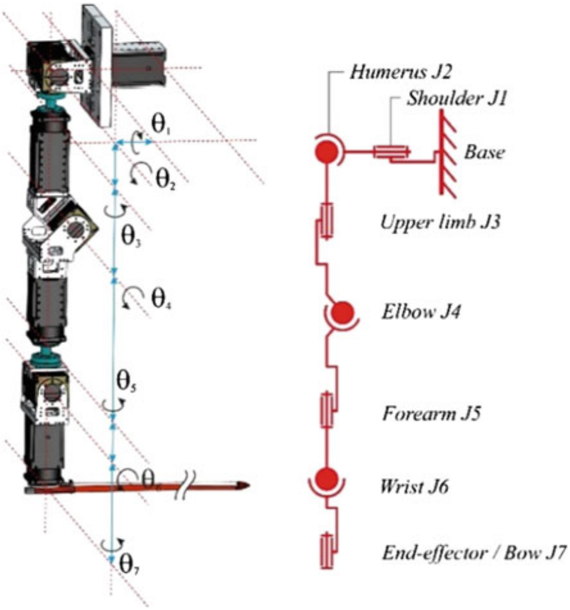


Fig. 5 Denavit–Hartenberg parameters of Violin playing robot

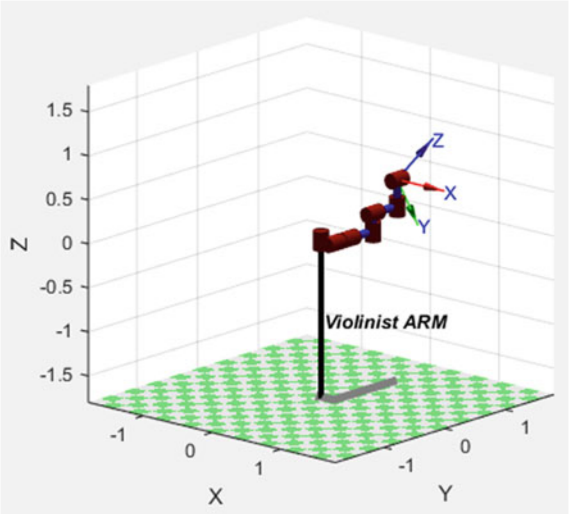


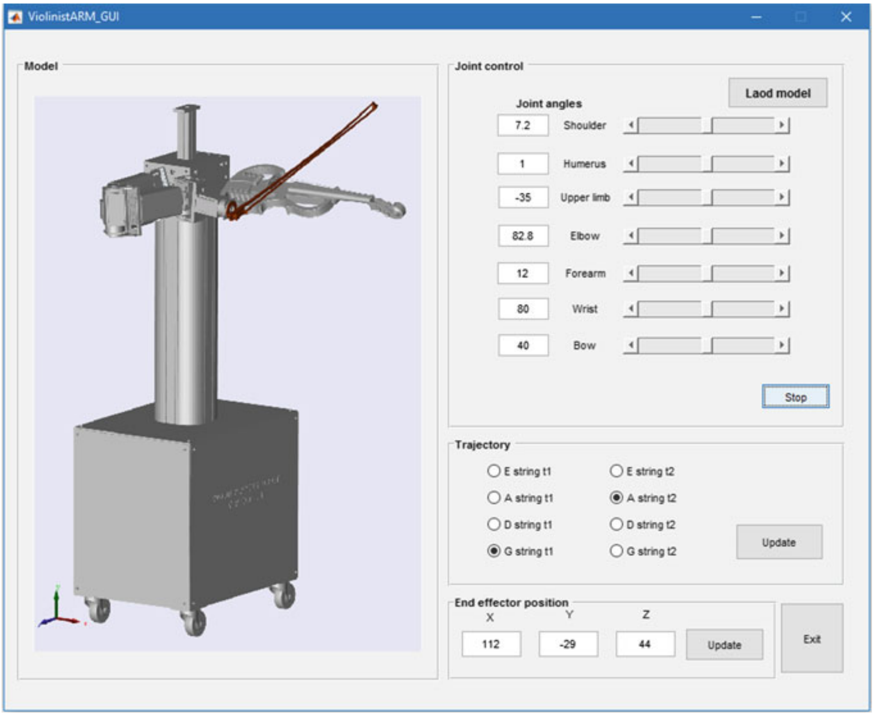
Fig. 6 The seven-link robot arm

4.2 Kinematics Model of the Seven-DOF Manipulator

In robot simulation, system analysis needs to be done, such as the kinematics analysis, its purpose is to carry through the study of the movements of each part of the robot mechanism and its relations between itself. The kinematics analysis is divided into forward and inverse analysis. The forward kinematics consists of

Table 1 D-H Link parameters table

Joint	θ_j	d_j	a_j	α_j	σ_j	$q_{\min}$	$q_{\max}$
1	q_1	0	1	1.571	0	−180	180
2	q_2	0	0.4318	0	0	−100	100
3	q_3	0.762	0	−1.571	0	−180	180
4	q_4	0	0.4318	1.571	0	−138	138
5	q_5	0	0.353	0	0	−180	180
6	q_6	0.8318	0	0	0	−108	108
7	q_7	0	0	1.047	0	−180	180



ViolinistARM_GUI

Model

Joint control

Laod model

Joint angles

7.2	Shoulder
1	Humerus
-35	Upper limb
82.8	Elbow
12	Forearm
80	Wrist
40	Bow

Stop

Trajectory

○ E string t1 ○ E string t2
○ A string t1 ● A string t2
○ D string t1 ○ D string t2
● G string t1 ○ G string t2

Update

End effector position

X	Y	Z
112	-29	44

Update

Exit

Fig. 7 The control interface

finding the position of the end effector in the space knowing the movements of its joints as

$$^{j-1}A_j(\theta_j, d_j, a_j, \alpha_j) = T_{Rz}(\theta_j)T_z(d_j)T_x(a_j)T_{Rx}(\alpha_j) \tag{4.1}$$

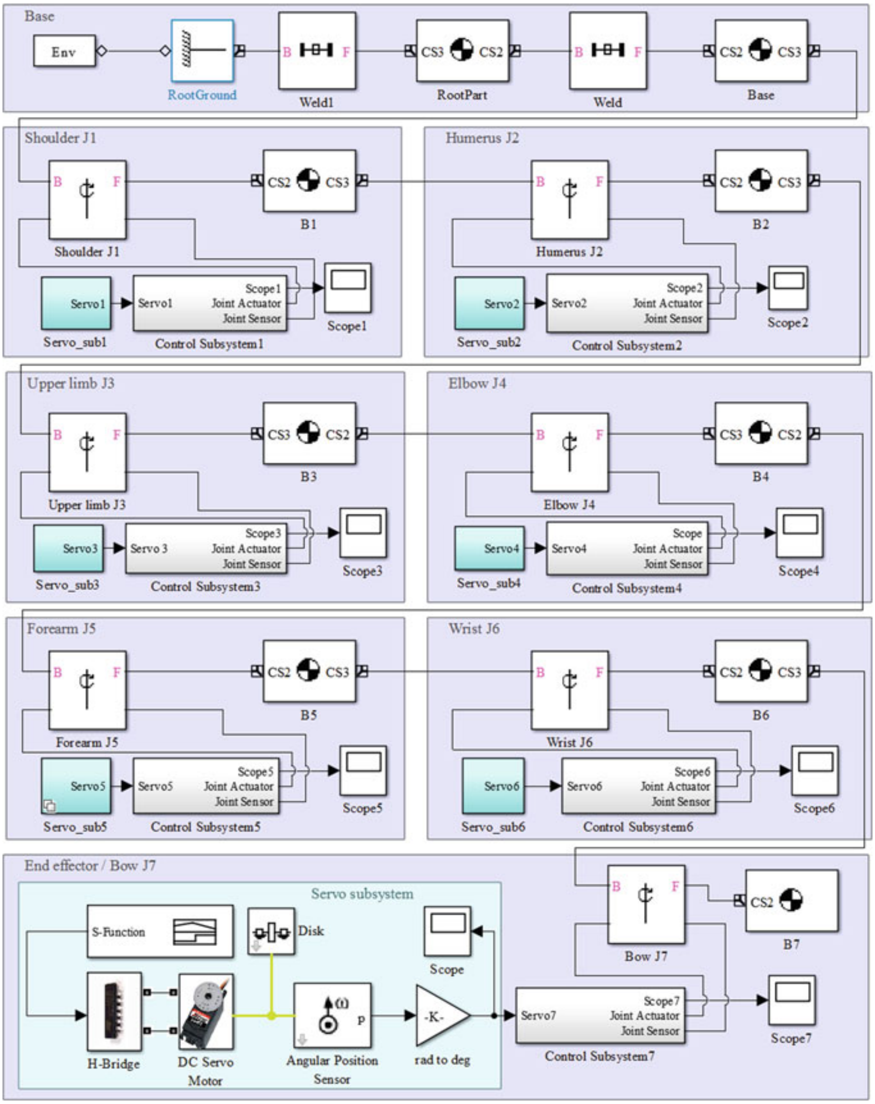


Fig. 8 Physical model in Simscape

which can be expanded as

$$
{}^{j-1}A_j = \begin{pmatrix} \cos\,\theta_j & -\sin\,\theta_j \cos\,\alpha_j & \sin\,\theta_j \sin\,\alpha_j & a_j \sin\,\theta_j \\ \sin\,\theta_j & \cos\,\theta_j \cos\,\alpha_j & -\cos\,\theta_j \sin\,\alpha_j & a_j \sin\,\theta_j \\ 0 & \sin\,\alpha_j & \cos\,\alpha_j & d_j \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.2}
$$

and the inverse kinematics consists of the determination of the joint variables corresponding to a given end effector position and orientation [10].

$$q = \mathrm{K}^{-1}(\xi) \tag{4.3}$$

Figure 5 shows the Denavit–Hartenberg or D-H convention (Fig. 6 and Table 1).

4.3 GUI Interface and Simscape

Creation of solid modeling and physical model of the robot arm control to ensure that, MATLAB/GUI interface with the prepared. Control desktop using the GUI interfaces via the programming language MATLAB m-file editor out of writer. GUI interface is used by two ways for violin playing robot control. First usage is control of each joint, which is controlled rotation angle variable of joints. Other usage is control of end effector, which is controlled by position at trajectory section of GUI. In end effector position frame, end effector position at bow of violin playing robot can be generated by Update command button. Figure 7 shows the designed interface for control of simulation.

In other to obtain three-dimensional model of system, the physical model in Simscape is considered. It is shown in Fig. 8.

In physical model, subsystems of servo motor control for each joint of three-dimensional model are presented. Furthermore, physical model of DC servo motor is shown in Fig. 9. In this model, input variable of each joint should be inserted. First, the angle variable of each joint is described in GUI interface. Furthermore, movement of each servo motor on the joint is carried out by Start command button.

In addition, string position of violin can be selected in trajectory panel of GUI interface. After that, manipulator moves by selected string position of violin.

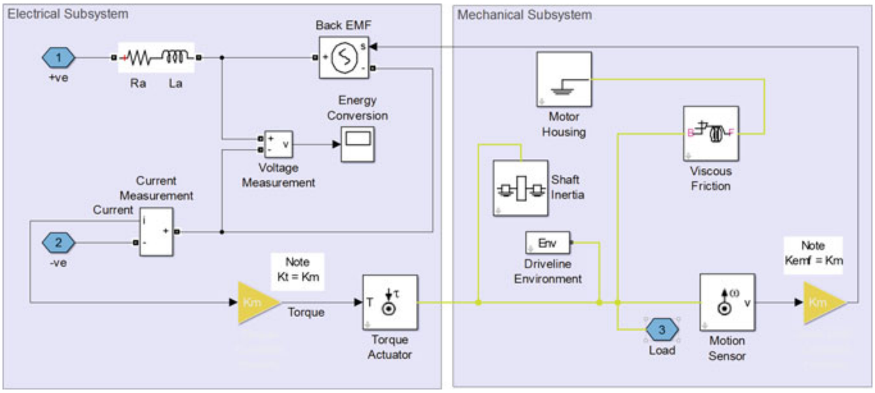


Fig. 9 Physical model of DC Servo motor [11]

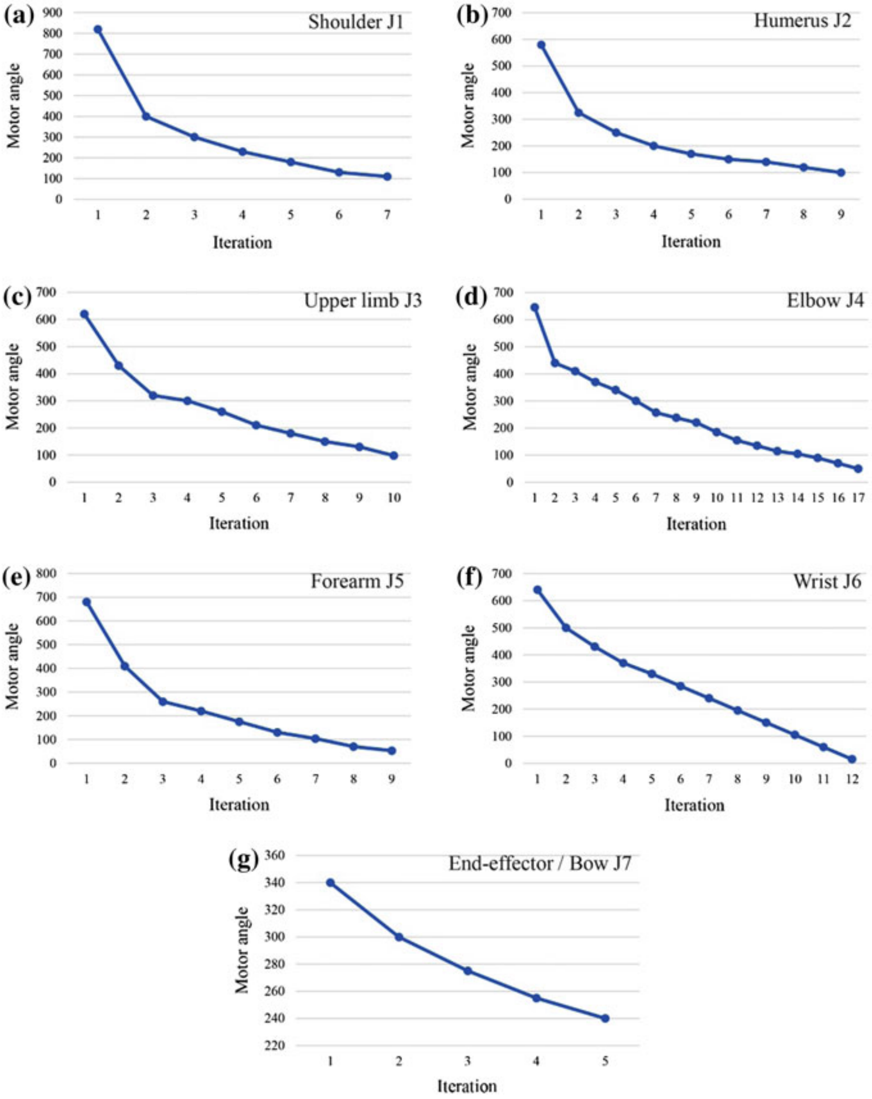


Fig. 10 The iteration, motor angle graph: **a** Shoulder J1 servo motor. **b** Humerus J2 servo motor. **c** Upper limb J3 servo motor **d** Elbow J4 servo motor. **e** Forearm J5 servo motor. **f** Wrist J6 servo motor. **f** End effector/Bow J7 servo motor

MATLAB/Simulink environment-Simscape physically modeled servo motor in used in the simulation. Physical modeling of the servo motor is shown in Fig. 9.

The switching signal according to the points of the robot arm joints to be 6.6 s with a maximum transmission time. Any servo motor drive, depending on starting point time of 6.6 s with the transmission of a signal is applied, the starting point, which means to be moved 360^0. So according to the desired angle of rotation of the drive signal is applied varies between 0 and 6.6 s.

5 Simulation Results

The model of the entire physical system by using MATLAB/GUI. Each joint position of the robot arm has been controlled by RC servo motors. Target angle values of each motion point are entered by using control interface. The program file is created as a MATLAB/m-file under the user interface that contains.

Figure 10 shows motor angle of each joint. Desired angle is inserted in GUI interface. After that, real angle will be moved to desired angle by iteration. Iteration step is not same for each joint angle. From figure, we can see that joint angle is decreasing.

6 Conclusion and Future Work

In this paper, 3D modeling of robot arm with 7 joints is researched. 3D model was created Solidworks software. In addition, MATLAB/SimMechanics tools are used. Note that, physical model of servo motor and its PID control system were implemented. From results, those combination models were successfully implemented. Furthermore, angle, torque, current consumption and voltage of each joint can be measured in this model.

In the future, physical model with several sensors and auditory feedback, and real experimental system will be continuously researched.

Acknowledgments This research was supported by Technology Innovation Program of the Knowledge economy (No. 10041834, 10045351) funded by the Ministry of Knowledge Economy (MKE, Korea), the National Research Foundation of Korea Grant funded by the Korean Government (No. 2012R1A1A2043822, 2014S1A5B6035098) and the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the Global IT Talent support program (IITP-2014-ITAH0905140110020001000100100) supervised by the IITP (Institute for Information & communications Technology Promotion).

References

1. Ayres, R., Miller, S.: The impact of industrial robots. International Publication CMU-R1-TR-81-7, Robotics Institute, Carnegie Mellon University (1981)
2. Solis, J., Takanishi, A., Hashimoto, K.: Development of an anthropomorphic saxophone-playing robot. Adv. Soft. Comput. **83**, 175–186 (2010)
3. Solis, J., Taniguchi, K., Ninomiya, T., Yamamoto, T., Takanishi, A.: Development of Waseda flutist Robot WF-4RIV: Implementation of auditory feedback system. Int. Conf. Robot. Autom. 3654–3659 (2008)

4. Yoshihiro Kusuda.: Toyota's violin-playing robot. Int. Industrial Robot: An International Journal, Vol. 35 Iss: 6 (2008) 504–506
5. Shibuya, K., Matsuda, S., Takahara, A.: Toward developing a violin playing robot—Bowing by Anthropomorphic Robot Arm and Sound Analysis. In: Proceedings of 16th IEEE International Symposium on Robot and Human Interactive Communication, pp. 763–768 (2007)
6. Jo, W., Park, H., Lee, B., Kim, D.: A study on improving sound quality of violin playing robot. In: Proceedings of the 6th International Conference on Automation, Robotics and Applications, pp. 185–191 (2015)
7. Min, B.C., Matson, E.T., An, J., Kim, D.: Improvement of violinist robot using a passive damper device. J. Intell. Rob. Syst. **72**(3–4), 343–355 (2013)
8. Cronhjort, A.: A computer-controlled bowing machine (MUMS). Department for Speech, Music and Hearing Quarterly Progress and Status Report (STL-QPSR), vol. 33, pp.61–66 (1992)
9. Web.: Robotis Co., Ltd, Dynamixel pro. http://support.robotis.com/en/techsupport_eng.htm#product/dynamixel_pro.htm
10. Corke, P.I.: Robotics, Vision and Control Fundamental Algorithms in Matlab, vol. 73 2011. ISBN 978-3-642-20143-1. doi:10.1007/978-3-642-20144-8
11. Web.: The MathWorks, Inc. Simulink Design Optimization, DC Servo Motor Parameter Estimation http://www.mathworks.com/help/sldo/examples/dc-servo-motor-parameter-estimation.html?s_tid=srchtitle
12. Schelleng, J.C.: The bowed string and the player. J. Acous. Soc. Am. **53**(1), 26–41 (1973)
13. Yilmaz, E.N., Battal, O.: Using active learning in motor control and matlab simulation. J. Autom. Control Eng. **3**(4), 299–303 (2015)
14. Simscape User's Guide, the MathWorks, Inc. U.S.A., pp. 15–49 (2015)
15. Shih-An Li, Chung-Wei Weng, Yi-Hong Chen, Chia-Hung Lo, Min-Hao Yang, Yi-Chun Lin, Ming-Hua Hsieh, Ching-Chang Wong.: Servo motor controller design for robotic manipulator. Int. Intelligent Signal Processing and Communication Systems (ISPACS 2012), (2012) 254–257
16. Norrlof, M.: Iterative Learning Control Analysis, Design, and Experiments. ISBN 91-7219-837-0 ISSN 0345-7524, No. 653 (2000)

Author Biographies

Jargalbaatar Yura is M.S. course student in the Department of Electronics and Radio Engineering at Kyung Hee University.

He has received her B.S. degrees in Electrical Engineering from the Mongolian University of Science and Technology, Mongolia, in 2013, respectively. His current research interests include automation of electrical systems and robotics.

Mandakh Oyun-Erdene has received her B.S. and M.S. degrees in Electrical Engineering from the Mongolian University of Science and Technology, Mongolia, in 2008 and 2011, respectively. And she received her M.S. degree in Electrical Engineering from Kyung Hee University, Korea, in 2014. Her current research interests include automation of electrical systems and robotics.

From 2011, she began as a lecturer at the College of Power Engineering, Mongolian University of Science and Technology.

Byambasuren Bat-Erdene received his B.S. and M.S. degrees in Electrical Engineering from the Mongolian University of Science and Technology, Mongolia, in 2001 and 2002, respectively. He received his Ph.D. degree in Electrical Engineering from Kyung Hee University, Korea, in 2013.

He has been with the Department of Electric Technique at the College of Power Engineering, Mongolian University of Science and Technology, where he is currently an associate professor. His research interests include relaying protection, automation of electrical systems, control systems, microprocessor systems, and robotics.

Dong-Han Kim received his B.S., M.S., and Ph.D. degrees in Electrical Engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejon, Korea, in 1995, 1998 and 2003, respectively.

He has been with the Department of Electronics and Radio Engineering at Kyung Hee University, where he is currently an associate professor. He was a postdoctoral fellow at University of Illinois at Urbaba-Champaign in 2003. He was the recipient of the first award of a humanoid robot competition at the FIRA robot soccer in 2002. His current research interests are in the area of a multiagent robot system.

Gaze Control of Humanoid Robot for Learning from Demonstration

Bum-Soo Yoo and Jong-Hwan Kim

Abstract Robots can learn knowledge by observing demonstration of humans. As tutees, robots need to not only observe human behaviors, but also make proper feedbacks for human tutors because learning is an interactive process in which information is delivered in bidirectional ways between humans and robots. Gaze is an adequate method for robots to provide human tutors with feedbacks that robots are concentrating on current learning because gaze directly represents where they are paying attention to. This paper proposes a gaze control algorithm with a state machine in learning from demonstration. A human tutor shows demonstration in front of a robot tutee, and the robot tutee observes the demonstration for learning. The robot tutee perceives external environment through its camera, recognizes a human and objects, and figures out a state at which the robot tutee is situated. Then, the robot tutee gazes at proper targets that are predefined by the state machine. The human tutor also adjusts the demonstration to make learning more effectively according to the robot tutee's feedbacks. The effectiveness of the proposed method is demonstrated through the experiments with a robotic head with 17 degrees of freedom, developed in the RIT Lab., KAIST.

Keywords Face detection · Gaze control · Learning from demonstration · Object recognition · Robotic head · State machine

1 Introduction

With the advancement of intelligence technology, robots are becoming intelligent, and they have been gradually coming into our daily life [1]. They need to be able to learn new knowledge and to update their knowledge according to situations.

B.-S. Yoo (✉) · J.-H. Kim
School of Electrical Engineering, KAIST, 291 DaeHak-ro, Yuseong-gu,
Daejeon, Republic of Korea
e-mail: bsyoo@rit.kaist.ac.kr

J.-H. Kim
e-mail: johkim@rit.kaist.ac.kr

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_21

Learning from humans is one of the powerful tools because programmers may not expect in all situations that robots will encounter [2].

Robots can learn from humans by observing humans' demonstration [3]. Learning from demonstration is an interactive process in which information is delivered in a bidirectional way, not unidirectional way between humans and robots [4]. Human tutors deliver knowledge to robot tutees with proper methods, and robot tutees make feedbacks for human tutors to express their states such as understanding and targets they are paying attention to. Then, the human tutors adjust their demonstrations to build joint attention that makes learning more effectively [5]. Thus, robots need to provide feedbacks to humans when they learn knowledge from humans' demonstration.

Robots can make feedbacks by their gaze because gaze definitely represents where they are paying attention to [6]. This paper proposes a gaze control algorithm with a state machine in learning from demonstration. A human tutor shows a demonstration in front of a robot tutee, and the robot tutee perceives external environment including the human tutor and objects with a camera. Based on a state machine, the robot tutee gazes at proper targets that are predefined in the state machine. Then, according to the robot tutee's feedbacks, the human tutor adjusts his demonstration to help the robot tutee to focus on the demonstration. The proposed algorithm is demonstrated by the robotic head, developed in the Robot Intelligence Technology Lab., KAIST. A human tutor uses a watering pot for watering a flower in front of the robotic head, and the robotic head gazes at a proper direction according to perceived situations.

A state machine is used for gaze control in learning from demonstration. This is because, according to the direct matching hypothesis, when humans observe other people's behaviors, they control their gaze as if they do such behaviors [7]. It means that a state machine is a proper method for gaze control in learning from demonstration if a state machine can be applied to humans' gaze control in performing tasks. When humans perform a task, they need information for the task and gaze is used to acquire the required information. Since the same task requires the same information, humans show a large consistency in gaze control [8]. Therefore, a state machine is a proper method for gaze control not only in learning from demonstration, but also in performing tasks.

The paper is organized as follows. Section 2 describes an object recognition algorithm, a face detection algorithm, and a state machine used in the paper. Section 3 describes experimental results with the robotic head. Last, we conclude with important remarks in Sect. 4.

2 Gaze Control Algorithm

The robotic head perceives external environment with a RGB-D camera and control its head to gaze at proper targets according to perceived situations. Figure 1 shows a predefined state machine of gaze control for learning the demonstration of watering a flower. At first, the robotic head gazes at a face of a human tutor. Gazing at a

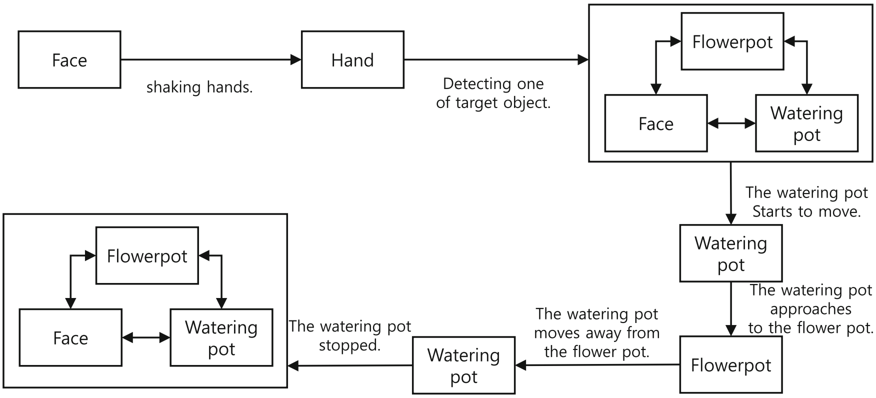

Face
shaking hands.
Hand
Detecting one of target object.
Flowerpot
Face
Watering pot
The watering pot Starts to move.
Watering pot
The watering pot approaches to the flower pot.
Flowerpot
The watering pot moves away from the flower pot.
Watering pot
The watering pot stopped.
Flowerpot
Face
Watering pot

Fig. 1 A state machine for gaze control of watering of a flower

face works as an ostensive signal by making an eye contact to make learning more effectively [9]. The robotic head detects hands of the human tutor and changes its gaze direction to the hands. As a predefined starting signal, the human tutor shakes his hands, and the robotic head starts to pursuing his hands. During pursuing the hands, the robotic head detects the watering pot and flower pot, and it changes its gaze direction to one of the objects and the face of the human tutor. The robotic head gazes at one of them sequentially until the human tutor performs behaviors with the object. Note that a face is chosen as one of attractions even if it is task-irrelevant [10]. When the human tutor holds the watering pot and raises it, the robotic head notices movement of the watering pot and gazes at the watering pot because the watering pot is related to human behaviors. When the human tutor brings the watering pot to the flower pot, the robotic head changes its gaze direction to the flower pot, the destination of the movement. After watering, the human tutor takes the watering pot away from the flower pot, and the robotic head changes the gaze direction from the flower pot to the watering pot. The robotic head gazes at the objects and the faces sequentially when there is no movement of the watering pot.

2.1 Face Detection

Gazing at a face could produce an eye contact and make the learning more effectively. A cascade of classifiers with haar-like features in the OpenCV are used to detect faces. A window is slid in the perceived image and the image in the window is delivered to the saccade of classifiers. Figure 2 shows the cascade of classifiers, a series of weak filters. Instead of checking whether the input image is a face or not, each filter, F_1, F_2, …, F_n, only checks whether it satisfies a specific condition or not. An image satisfying the condition is passed to the next filter, and an image is discarded if it does not satisfy the condition. Only images satisfying all conditions are

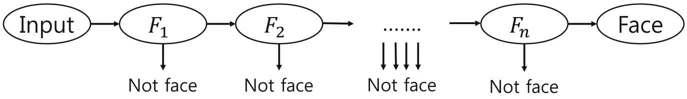


Fig. 2 The cascade of classifiers used for face detection

regarded as faces. The cascade of classifiers shows good performance with a small computation burden because images discarded in early stage are no longer to be processed. This filtering process is repeated with the change of the sliding window size to contain all scales of faces.

Note that to make the algorithm robust to noise, frontal and profile of faces are detected simultaneously, and once faces are detected, they are tracked with the HSV color information and depth information.

2.2 *Object Recognition*

The robotic head also needs to recognize objects in the demonstration. Figure 3 shows the object detection process. The input image is filtered with two steps. First, pixel points whose depth information is larger than a certain threshold is filtered. This step can remove background in the demonstration. Second, based on an assumption that objects are usually placed on the plane, not in the air, a plane is found by the random sample consensus (RANSAC) algorithm, and pixel points in and under the plane are removed. Therefore, the robotic head can detect objects only on the plane. After the filtering, rest regions are considered as objects, and the convolutional neural network (CNN) in the Convolutional Architecture for Fast Feature Embedding (CAFFE) is used to recognize objects. As one of feed-forward artificial neural networks, the CNN consists of several layers such as rectified linear unit layers, convolutional layers, and pooling layers. Each neuron in the CNN can only look at a small portion of the input image, called a receptive field. The receptive fields are slightly overlapped and tiled the input image. Each neuron predicts the output based on its receptive field, and results of predictions are gathered gradually to one output by forwarding process. The CNN shows the state-of-the-art performance because it has a strong point in learning feature extraction by itself [11].

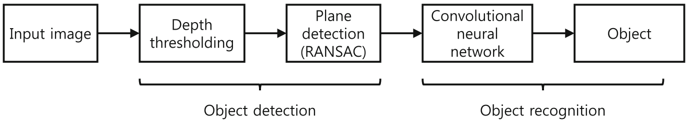


Fig. 3 The object recognition process

3 Experiments

3.1 Robotic Head

Figure 4a shows the robotic head in the experiment [12]. Its height, width, length, and weight are 160,00 mm, 218.00 mm, 139.00 mm, and 2.54 kg, respectively. It perceives external information with one RGB-D camera. Table 1 shows the degrees of freedom (DoFs) of the robotic head. In case of the neck, one DoF is, respectively, assigned to each yaw, pitch, and roll axis, and one DoF is added to the pitch axis for nodding. In this paper, 3 DoFs for the pitch and yaw axes of the neck and 2 DoFs of the eyes are used. The robotic head controls these DoFs to make the selected gaze direction be positioned at the center of its visual inputs.

The robotic head performs gaze control in real environment. It needs to respond quickly according to perceived information. Therefore, two graphic processing units (GPU) of GTX980 and GT630 were used. They showed about 8–10 frames per second (fps), which is hard to called a real-time system. However, since information in real environment is continuous and does not change drastically, the robotic head can perform gaze control in real environment with a little delay.

3.2 Experimental Result

The experiment was conducted in real environment, as shown in Fig. 5. There was a table in front of the robotic head, and a watering pot and flower pot were placed on the

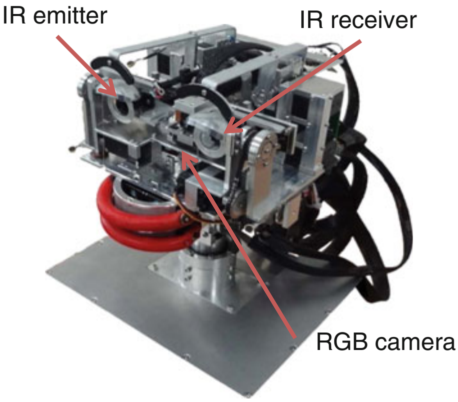


Fig. 4 The robotic head used in the experiment

Table 1 Degrees of freedom of the robotic head

Part	Eye lids	Eye brows	Eyes	Lips	Jaw	Neck
DoF	2 × 2	1 × 2	2	2 × 2	1	4

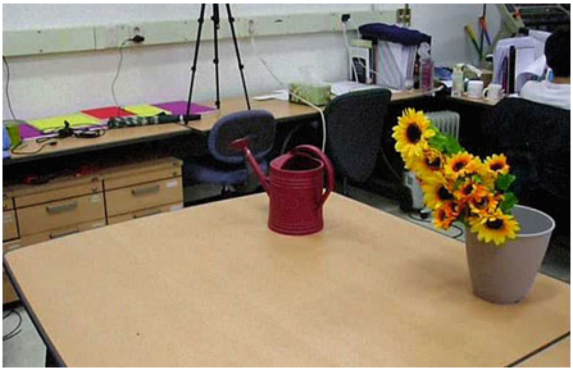


Fig. 5 Experiment environment

table. A human tutor on the opposite side of the table performed watering with the pot, and the robotic head showed feedbacks to the human tutor with gaze. According to feedbacks, the human tutor adjusted his behaviors to make the demonstration more effectively.

Snapshots of the experiments are shown in Fig. 6, where the first column represents target gaze direction marked with a circle, the second column represents a face detection marked with circles, the third column represents object recognition, and the fourth column represents skeletons of the demonstrator whose hands are marked with red lines. In the third column, rectangles and circles, respectively, represent detected and tracked objects. A skeleton of the demonstrator in the fourth column was calculated with a simple unprojection algorithm to a camera coordinate.

When the robotic head started its gaze control, it detected a face, a watering pot, a flower pot, and a human, as shown in Fig. 6a. It gazed at the face of the human. When human shook hands, the predefined signal for starting the demonstration, the robotic head started to pursue the hands. During the pursuing, the robotic head detected the watering pot and flower pot, and gazed at them sequentially, as shown in Fig. 6b and c. In Fig. 6c, somebody walked in, passed through, and seated at a chair, and the robotic head detected his face. However, since the state machine considered the first tracked face as the tutor, the newcomer was ignored in the demonstration. In Fig. 6d and e, the tutor held the watering pot and raised it. Then, the robotic head detected its movement and gazed at it. In Fig. 6f, the tutor brought the watering pot to the flower pot and watered the flower, and the robotic head gazed at the flower pot. At this time step, the tutor slowed his behavior for the robotic head to track the watering pot stably. In Fig. 6g, the tutor took the watering pot away from the flower pot, and the robotic head gazed at the watering pot. At this step, the tutor also moves the watering pot slowly for robot to track it. Finally, the demonstration was finished, and the robotic head gazed at the face, watering pot, and flower pot, sequentially again, as shown in Fig. 6h. The robotic head provided a proper gaze direction according to the perceived situations with the help of the state machine, and the human tutor also recognized the gaze and slowed down his demonstrations to make the robotic head follow his demonstrations.

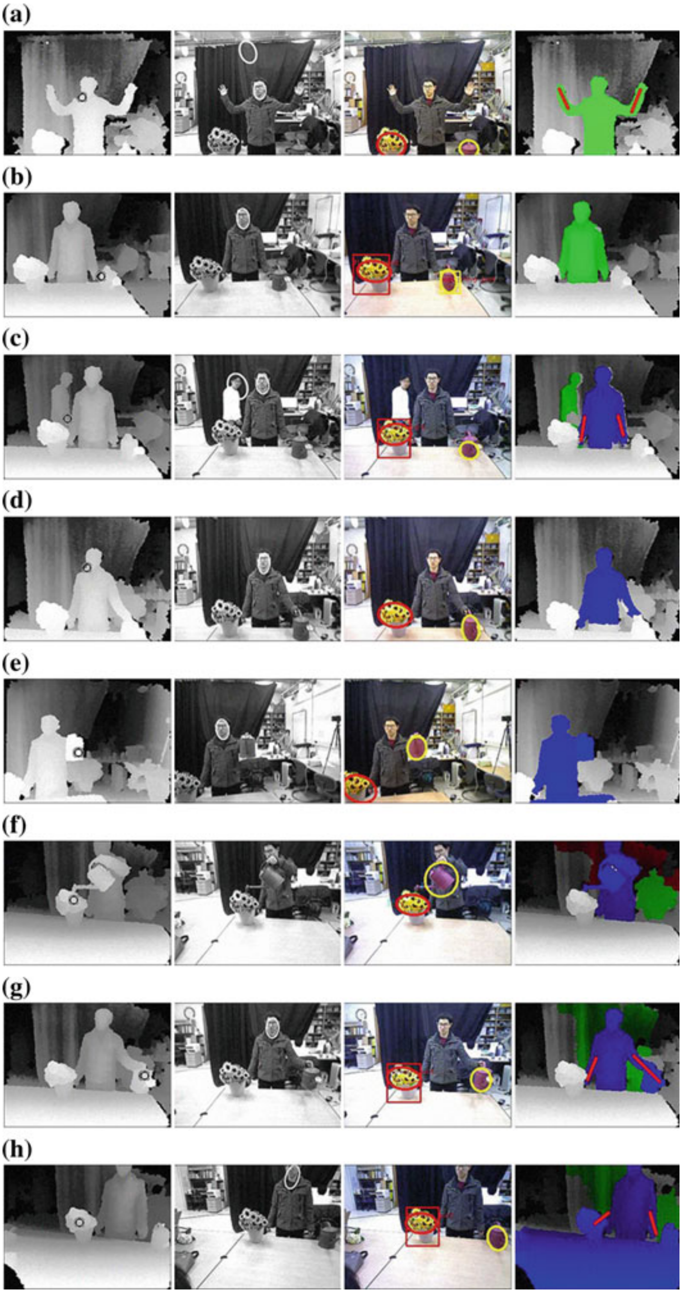


Fig. 6 **a** The robotic head gazed at a human tutor. **b** The robotic head gazed at objects and a face, sequentially. **c** Even through the robotic head found a person, it ignored him. **d** The tutor held the watering pot. **e** The tutor raised the watering pot. **f** The tutor brought the watering pot to the flower pot. **g** The tutor took the watering pot away from the flower pot. **h** The tutor finished his behaviors, and the robotic head gazed at the objects and the face, sequentially

4 Conclusion

This paper proposed the gaze control algorithm for learning from demonstration. For perceived image, the robotic head detected faces with the cascade of classifiers and recognized objects with the CNN. Based on the recognized information, the robotic head produced feedbacks with its gaze, and a human adjusted his behaviors using the feedbacks. The proposed algorithm was demonstrated through the experiment with the robotic head. The proposed gaze control algorithm will increase the performance of robots' learning from humans.

Acknowledgments This work was partly supported by Institute for Information communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0101-15-0551, Virtual Creatures with Digital DNA), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2014R1A2A1A10051551), and the Technology Innovation Program, 10045252, Development of robot task intelligence technology, supported by the Ministry of Trade, Industry, and Energy (MOTIE, Korea).

References

1. Kim J-.H., Choi S-.H., Park I-.W., Zaheer S.A.: Intelligence technology for robots that think. IEEE Comput. Intell. Mag. **8**(3), 70–84 (2013)
2. Breazeal, C., Scassellati, B.: Robots that Imitate Humans. Trends Cogn. Sci. **6**(11), 481–487 (2002)
3. Thomaz, A.L., Cakmak, M.: Learning about objects with human teachers. In: ACM/IEEE International Conference on Human Robot Interaction, LA Jolla, CA, USA, pp. 15–22 (2009)
4. Pitsch, K., Volmer, A.L., Fritsch, J., Wrede, B., Rohlfing, K., Sagerer, G.: On the loop of action modification and the recipient's gaze in adult-child-interaction. Gesture and Speech in Interaction, Poznan, Poland, pp. 1–17 (2009)
5. Huang C.M., Thomaz A.L.: Effects of responding to, initiating and ensuring joint attention in human-robot interaction. In: IEEE International Sympoisum on Robots and Human Interactive Communications, Atlanta, GA, USA, pp. 65–71 (2011)
6. Mora A., Gals D.F., Kanda T., Hagita N.: A teleoperation approach for mobile social robots incorporating automatic gaze control and three-dimensional spatial visualization. IEEE Trans. Syst. Man Cybern. **43**(3), 630–642 (2013)
7. Flanagan, J.R., Johansson, R.S.: Action plans used in action observation. Nature **424**(6950), 769–771 (2003)
8. Tatler, B.W., Hayhoe, M.M., Land, M.F., Ballard, D.H.: Eye guidance in natural vision: reinterpreting salience. J. Vis. **11**(5), 1–23 (2011)
9. Senju, A., Csibra, G.: Gaze following in human infants depends on communicative signal. Curr. Bio. **18**(9), 668–671 (2008)
10. Cerf, M., Harel, J., Einhäuser W., Koch, C.: Predicting human gaze using low-level saliency combined with face detection. Adv. Neur. In. 241–248 (2008)
11. LeCun, Y., Bottou, L., Bengio, Y., Haffer, P.: Gradient-Based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
12. Yoo, B.-S., Kim, J.-H.: Fuzzy integral-based gaze control of a robotic head for human robot interaction. IEEE Trans. Cybern. **45**(9), 1769–1783 (2015)

A Novel Design of a Full Length Prosthetic Robotic Arm for the Disabled

Vudattu Sachin Kumar, S. Aswath, Tellakula Sai Shashidhar and Rajesh Kumar Choudhary

Abstract This paper shows the design methodology of a humanoid robotic arm with realistic mechanical structure and performance. Realistic and mechanically robust structure for a prosthetic arm was developed in Solid Works. The torque, power requirements and cost estimation were assessed systematically by interfacing the model with SimMechanics software. The robotic arm is equipped with several robot servo motors which perform as links between arms and perform arm movements by interfacing with a robot servo controller and the PIC16F886 microcontroller. The robot servo controller has the capability to drive the servo in controlled position, speed, and acceleration modes. Due to the complexity of the arm kinematics, machine-learning techniques, which rely less on precise mathematical analysis, are implemented. ANFIS is one such machine-learning technique which helps in decision-making and control of robotic arms. This paper implements a MATLAB-derived multilayered ANFIS controller using a PIC16F886 microcontroller as a supervisory control for a 6 DOF robotic arm. This type of robotic arm has many advantages such as simple structure, high flexibility, low energy consumption, quiet operation, and sensory feedback which make it a prosthetic arm with very high resemblance to a normal arm. A good tradeoff between cost and performance is achieved in order to meet the goal of less expensive and useful

V.S. Kumar · T.S. Shashidhar
Department of Electrical and Electronics Engineering, Amrita Vishwa Vidyapeetham,
Kollam 690525, Kerala, India
e-mail: sachinkumarvudattu10@gmail.com

T.S. Shashidhar
e-mail: saisashidhar2003@gmail.com

S. Aswath (✉)
Department of Electronics and Communication Engineering, Amrita Vishwa
Vidyapeetham, Kollam 690525, Kerala, India
e-mail: aswathashh10@gmail.com; aswathmars2024@gmail.com

R.K. Choudhary
Department of Mechanical Engineering, Amrita Vishwa Vidyapeetham,
Kollam 690525, Kerala, India
e-mail: rkkapooria@gmail.com

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_22

robotic arm for the disabled. The practically built arm is tested with predefined paths and random positional targets with in work space and results are shown to act satisfactorily.

Keywords Robotic arm · Prosthetics · ANFIS · Control system · Servo motor · Fuzzy logic · Clustering · SimMechanics

1 Introduction

Robots are generally used to perform unsafe, hazardous, highly repetitive, and unpleasant tasks. They have many different functions such as material handling, assembly, arc welding, resistance welding, and machine tool load and unload functions, painting, spraying, etc. There are mainly two different kinds of robots: a service robot and an industrial robot. A service robot is a robot that operates semi or fully autonomously to perform services useful to the well being of humans and equipment, excluding manufacturing operations [1, 2].

An industrial robot, on the other hand, is officially defined by ISO (International Standards Organization) as an automatically controlled and multipurpose manipulator programmable in three or more axes [3]. Industrial robots are designed to move material, parts, tools, or specialized devices through variable programmed motions to perform a variety of tasks. An industrial robot system includes not only industrial robots but also any devices or sensors required for the robot to perform its tasks as well as sequencing or monitoring communication interfaces. Robotic manipulators, especially of the SCARA type, have found wide range application from small-scale sectors to high-end industries. They also play a primary role in helping disabled people to do their work in daily life. They are often used in inaccessible places, space, and robotically assisted surgeries. Designing an arm which can mimic human motions with appropriate forces and torques is a challenging task since the design involves the consideration of a large number of conflicting parameters such as battery management, appropriate force, and torque estimation, avoiding singularities, cost, modeling, and control of arm while still maintaining an anthropomorphic nature.

With the increased use of industrial robotic arms, a logical evolutionary step is to imitate detailed human movements. A group of engineers developed an eight degree of freedom (DOF) robot arm, this robot is able to grasp many objects, ranging from a pen to a ball, simulating the hand of human being [4]. Another group of students in Korea made an innovative robotic arm that could perform dancing, weight lifting, writing Chinese calligraphy, and color classification [5]. In space, the Space Shuttle Remote Manipulator System, known as the SSRMS or Canadarm, and its successor, is example of multi-degree of freedom robot arms that have been used to perform a variety of tasks such as inspections of the space shuttle (using a specially deployed boom with cameras and sensors attached at the end effector) and satellite deployment and retrieval maneuvers from the cargo bay of the space shuttle [6].

Another group of people in Mexico has completed a competitive low-cost robot arm with 4 DOF-[7]. One of the most important problems in robot kinematics and control is finding the solution of inverse kinematics. Traditional solution methods such as geometric, iterative, and algebraic are inadequate if the joint structure of the manipulator is complex. Complexity of modeling and control of arm is proportional to the DOF of the manipulator. As the complexity of a robot increases, obtaining the inverse kinematics is difficult and computationally expensive. For a 6-DOF robotic arm, these methods become inaccurate because of increased complexity in mathematical modeling and computation management [8].

Soft computing techniques provide a beneficial approach in the modeling and control of such complex systems. Systems with less mathematical model knowledge could be controlled successfully with neural networks, and systems with incomplete knowledge could be modeled and controlled using Fuzzy systems. Artificial Neural Networks (ANN) have a limitation with complexity and in higher DOF robotic arms, lack of intuition makes fuzzy modeling tough. It is well known that a solution exists in adaptive neuro-fuzzy inference systems (ANFIS), where the learning ability of neural networks to assist the Fuzzy systems to represent the knowledge expressed in the form of linguistic rules [9–12].

It is possible to create an ANFIS control system with a limited mathematical representation of the system [13]. Computer simulations conducted on 2DOF and 3DOF robot manipulators show the effectiveness of the approach [14].

The paper focuses on the design of a low-weight robotic arm with 6-DOF which is to be mechanically stable, economically reliable, and should mimic the natural human arm movements. The humanoid robotic hand is also designed to be cost-effective. Many different approaches exist for minimizing the weight of a robot. These approaches are based on the use of

(1) High-performance actuators
(2) Composite material selection
(3) Efficient mechanical design
(4) Advanced control and sensing techniques

Our work aims at developing a framework to generate arm motions of a humanoid robot in two aspects of human likeness and minimal torque requirements. The robotic arm with 6-DOF delivers fast, accurate, and repeatable movements. It features base rotation, a shoulder, an elbow, wrist motion, wrist rotation, and a functional gripper. This paper also focuses on the application of ANFIS on robotic arms with higher DOFs and larger reachable sets, For a 6 DOF arm, it can reach to 40,000 data points with 1 cm resolution in end-effector positioning. The number of neural networks and the number of membership functions in each ANFIS controller need to be optimized in order to obtain satisfactory control performance and reduce computational cost. The cost estimation and power requirements were assessed systematically by interfacing the model with SimMechanics. This type of robotic arm has many advantages like low cost, simple structure, high flexibility, low energy consumption, quiet operation, and sensory feedback.

2 Mechanical Design

The following aspects of the humanoid robotic arm such as movement and material handling have been handled with respect to design, implementation, and development. Movements inculcated in the design replicate the actual human arm, wherein the cost and complexity of arm are kept under check. A complete mechanical design of a full length prosthetic hand was developed with marginally increased functionality. Table 1 shows the humanoid robotic arm specifications and the Fig. 1 shows the components of the mechanical design of the humanoid robotic arm.

Table 1 Humanoid robotic arm specifications

Design payload	300 g
Degrees of freedom	6
Arm mass	1200 g
Actuator type	Dc servo motor
Average current draw	10 A/h
Max reachable height	60 cm

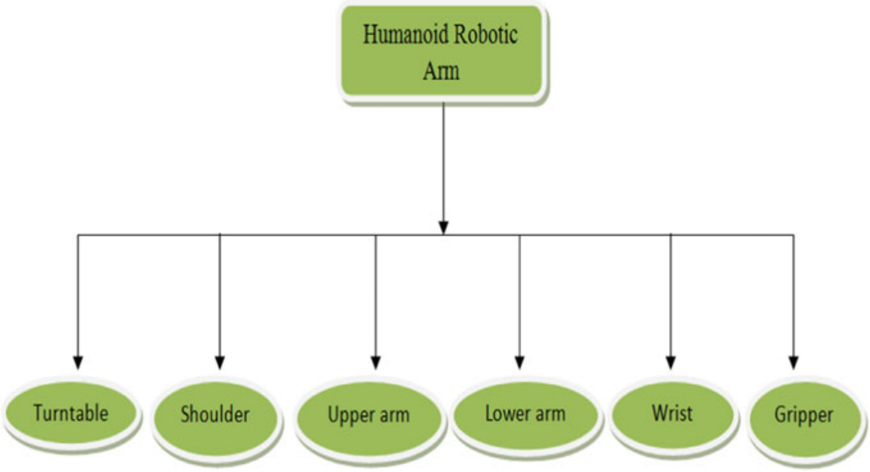


Fig. 1 Components of the mechanical design of the humanoid robotic arm

Table 2 Preliminary weight description

Component description	Weight (g)
First link	92
Second link	157
End effecter/payload	300
First motor assembly	185
Second motor assembly	205
Weight to payload ratio (moving parts only)	3.1:1

2.1 Design Selection-Humanoid Robotic Arm

Table 2 shows the preliminary weight description of the humanoid robotic arm. In choosing materials and the shape for the fabrication of the humanoid robotic arm, the following were taken into consideration:

(1) Weight of the robotic arm
(2) The mode of manufacturing
(3) The ease of manufacturing the parts
(4) Ease of assembly
(5) Cost
(6) Strength and durability of the parts

2.2 Material Selection

The appropriate materials considered for developing the model are plastic polymer, carbon fiber, and aluminum. Thus among them, Billet 6061 aluminum was selected based on the factors that include its work envelop, dimensional accuracy, workability, maintainability, machinability, high strength, lower cost, strength, thickness, weldability, good strength-to-weight ratio, and availability. Table 3 below shows the mechanical properties of Billet 6061 aluminum.

Table 3 Mechanical properties of Billet 6061 Aluminum

	Ultimate tensile strength (MPa)	0.2 % Proof stress (MPa)	Brinell hardness (500 kg load, 10 mm ball)	Elongation 50 mm diameter (%)
0	110–152	65–110	30–33	14–16
T1	180	95–96		16
T4	179 min	110 min		
T6	260–310	240–276	95–97	9–13

2.3 Force Analysis

Figure 2 shows the force diagram used for load calculations. The calculations were carried out only for the joints that have the largest loads, since the other joints would have the same motor, i.e., the motor can move the links without problems.

These force representation values are also used for torque calculations

W_d 0.065 kg (weight of link *DE*)
W_c 0.115 kg (weight of link *CD*)
W_b 0.126 kg (weight of link *CB*)
L 0.3 kg (payload)
C_m 0.045 kg; D_m 0.105 kg (weight of motor)
L_{BC} 0.26 m (length of link *BC*)
L_{CD} 0.22 m (length of link *CD*)
L_{DE} 0.088 m (length of link *DE*)

Calculating the sum of forces in the Y-axis using the loads is shown in Fig. 3 and solving for C_y and $C_{b,}$ is shown in the Eqs. (1)–(4).

$$\sum Fy = (L + W_d + D_m + W_c + C_m) * g - Cy = 0 \tag{1}$$

$$C_y = (0.630) * 9.8\,\mathrm{m/s}^2 = 6.174\,\mathrm{N} \tag{2}$$

Fig. 2 Force representation of robotic arm

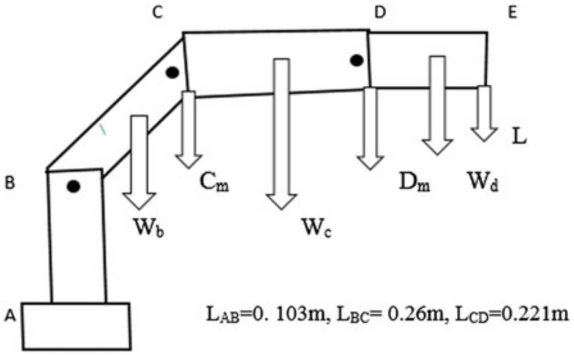
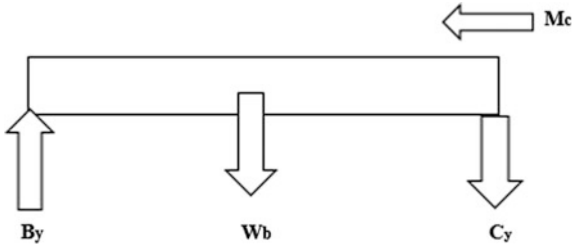


Fig. 3 Force diagram of link CB



$$\sum Fy = (L + W_d + D_m + W_c + C_m + W_b)^* g - Cb = 0 \tag{3}$$

$$C_b = (0.756)^* 9.8\,\mathrm{m/s}^2 = 7.408\,\mathrm{N} \tag{4}$$

Similarly, sum of moments around point C is obtained by Eq. (5), and point B is obtained by Eq. (6), torque at C and B is obtained by Eqs. (7) and (8).

$$\begin{aligned}\sum M_c = &-\left(\frac{W_c{}^* L_{cd}}{2}\right) - W_d{}^*\left(L_{cd} + \left(\frac{L_{de}}{2}\right)\right)\\ &-(L_{cd} + L_{de}) - D_m{}^*(L_{cd}) + M_c = 0\end{aligned} \tag{5}$$

$$\begin{aligned}\sum M_b = &-L(L_{bc} + L_{cd} + L_{de}) - W_d\left(L_{bc} + L_{cd} + \left(\frac{L_{de}}{2}\right)\right)\\ &-D_m{}^*(L_{bc} + L_{cd}) - W_c{}^*\left(L_{bc} + \left(\frac{L_{cd}}{2}\right)\right) - C_m{}^*(L_{bc})\\ &-W_b{}^*(L_{bc}/2) + M_b = 0\end{aligned} \tag{6}$$

$$M_c = 0.1397 \quad N - m \tag{7}$$

$$M_b = 0.2362 \quad N - m \tag{8}$$

From the above calculations, all these parameters are obtained for servo motor selection.

3 Solid Work Design-Humanoid Robotic Arm

The mechanical design of a humanoid robotic arm is based on a robotic manipulator which mimics the natural functions of a human arm. The links of such a robotic arm are connected by joints allowing rotational motion and those links are considered to form a kinematic chain. The other end of the kinematic chain of the manipulator is called the end effector and it is analogous to the human hand. Figure 4 shows the solid works layout of a humanoid robotic arm. As shown, the end effector is not included in the design because a commercially available gripper is used. This is because that the end effector of a robotic arm structure increases the degrees of freedom, which directly affects the complexity level of model and control of arm. It is much easier and economical to use a commercial one than to build it.

The humanoid robotic arm joints are actuated by electrical motors. Dynamixel servo motors were chosen, since they include positional feedback in real time to the motors and adjust the position according to the requirements. These servo motors were selected also based on the maximum torque required by the structure and the maximum allowable load that can be withstood by each joints of the designed

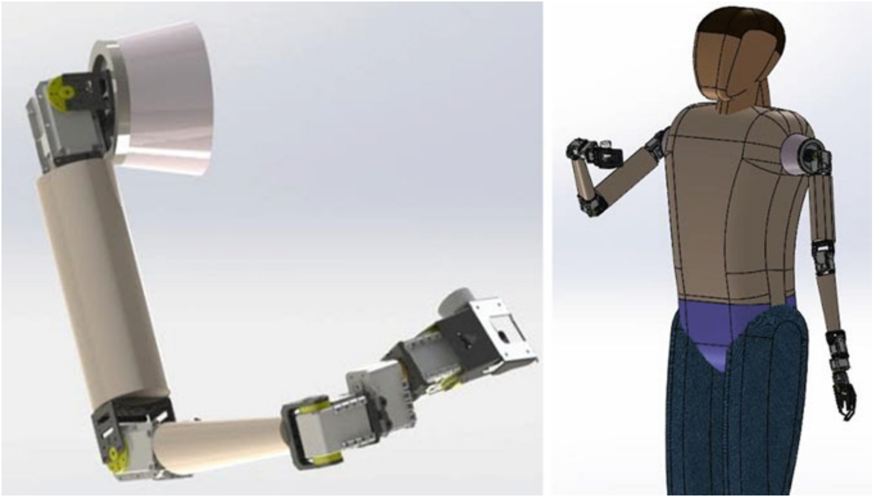


Fig. 4 Solid works layout of humanoid robotic arm

robotic arm. The mass properties of the designed solid works humanoid robotic arm are shown below.

Mass = 931.49 g
Volume = 768467.77 mm^3
Surface area = 331957.92 mm^2
Center of mass: (mm)
X = 338.97, Y = 210.63, Z = 853.52

Principal axes of inertia and principal moments of inertia (grams * square millimeters):

Taken at the center of mass

$$\begin{aligned} I_x &= (0.23, 0.19, 0.95) \quad P_x = 1164786.73 \\ I_y &= (0.58, -0.81, -0.02) \quad P_y = 44982080.21 \\ I_z &= (0.78, 0.55, -.030) \quad P_z = 45345554.06 \end{aligned}$$

Moments of inertia: (grams * square millimeters)

Taken at the center of mass and aligned with the output coordinated system

$$\begin{array}{lll} L_{xx} = 42908809.21 & L_{xy} = 1753414.06 & L_{xz} = 9657538.43 \\ L_{yx} = 1753414.06 & L_{yy} = 43503229.04 & L_{yz} = 8024069.70 \\ L_{zx} = 9657538.43 & L_{zy} = 8024069.70 & L_{zz} = 5080382.75 \end{array}$$

Moments of inertia: (grams * square millimeters)

Taken at the output coordinate system.

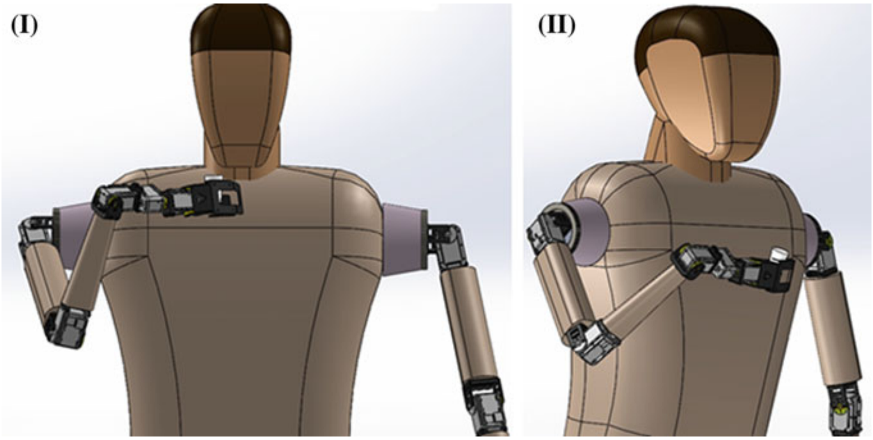


Fig. 5 Robotic arm 3D model 7(I) Front view 7(II) Side view

$$
\begin{array}{lll}
I_{xx} = 762828393.57 & I_{xy} = 68259101.39 & I_{xz} = 279154134.6 \\
I_{yx} = 68259101.39 & I_{yy} = 829124865.68 & I_{yz} = 175485733.3 \\
I_{zx} = 279154134.68 & I_{zy} = 175485733.3 & I_{zz} = 153433965.3
\end{array}
$$

Figure 4 shows the 3D model of the humanoid robotic arm. Figure 5(I), (II) show the front and side view of the robotic arm.

4 Design Challenges

Structural parameters like mobility, connectivity, and redundancy are to set up a mechanical structure which corresponds to a human arm, these parameters defines a number of actuators and the degrees of freedom required for the required movement of the platform. However this approach has disadvantages in adding more features.

The number of actuators implemented directly escalates the cost of the device and also reduces the stability due to weight. As the number of actuators increases, the time span of the process also increases. A large amount of humanoid arms have low quality and fewer than 6-DOF with a backlash and less payload. Thereby, it is seen that the robotic arm is affected by many parameters. We can say that the payload of an arm is the sum total effect of the axis and the workspace extensive care has to be undertaken in order to develop a humanoid robotic arm successfully.

Any humanoid robotic arm is devised with an aim of providing services to people in a selected work area. It is so designed that it has the size and dimensions of an average human arm while being as light as possible. This robotic arm should have 2DOF in a shoulder joint, 1DOF at an elbow joint and 3DOF at the wrist joint. The advantage of electrical motors is being environment friendly and noise less unlike usual hydraulic cylinders. The function of linear actuators is the same as that of the muscles operating with the principle of change of length.

5 SimMechanics

In order to capture the static and the dynamic properties of the humanoid robotic arm, both the mechanical properties of all its components, such as the moment of inertia, as well as its servos' dynamic responses, must be known to a certain degree of accuracy. These dynamic properties will be used in Simulink and SimMechanics in order to get an accurate simulator for the real humanoid robotic arm aiming at a good strategy. Figure 6 shows the modeling of robotic arm using SimMechanics.

Using the same design parameters which were implemented in solidworks, the model was built using SimMechanics software for force analysis. We have modeled this arm by considering gravity effect, constraints, and singularities. This figure represents the eating posture of a humanoid robotic arm. The designed arm should mimic the natural movements of a human arm. This eating posture of a humanoid robotic arm ensures the basic needs of a human arm. The torque required at each of the actuators of the modeled arm was simulated to analyze how much force is required to get this eating posture movement of the modeled arm.

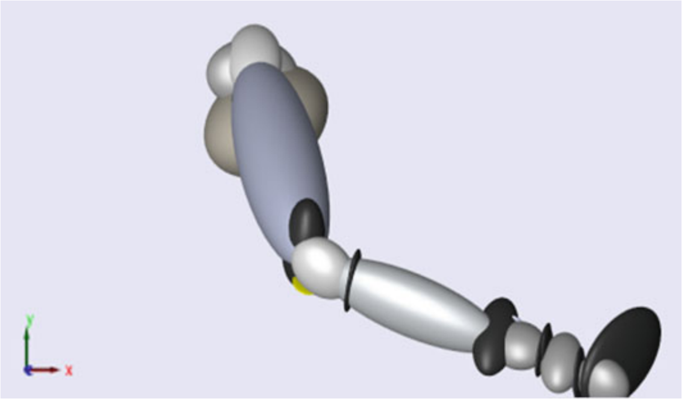


Fig. 6 Modeling of robotic arm using SimMechanics

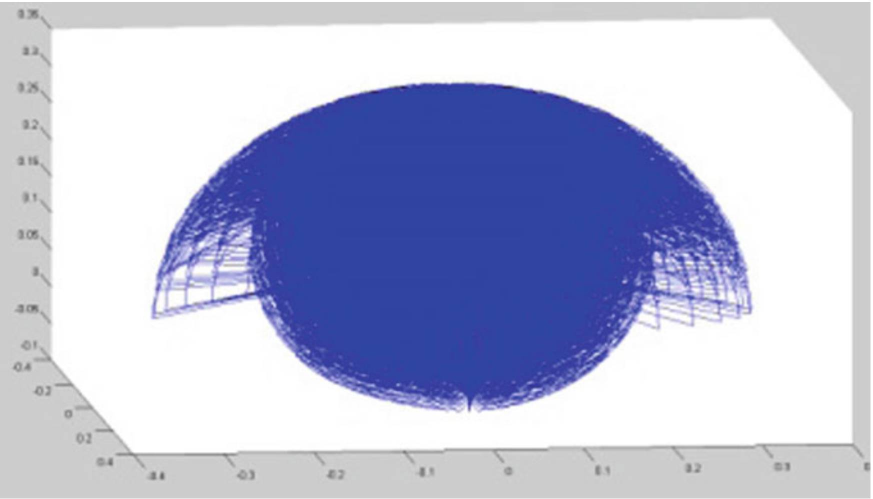


Fig. 7 Workspace region of the robotic arm

6 Data Generation

SimMechanics model developed based on the solid works layout of arm, includes the properties of mass, inertia, motor dynamics, etc. This model acts as a practical arm under gravity and generates dynamic data without any sensor interface. This model obtains data by behaving as real arm avoiding the link singularities and mimicking the practical motion of real human arm. A plot of 40,000 end-effector positions with 1 cm resolution is shown in Fig. 7. Obtained link angles and their corresponding end effector positions are used in training data to create ANFIS controller [9].

7 Control System-ANFIS in PIC16F886

ANFIS is a neural network with hybrid learning rules based on sugeno fuzzy interface system, which maps the input and output data. ANFIS architecture is developed to control each link of arm. Data obtained from SimMechanics model of arm is clustered to divide data into several statistical criteria. This procedure reduces the complexity on ANFIS training to find out the appropriate membership functions. Developed ANFIS in MATLAB is used for training the clustered data [9].

In order to make portable arm, the ANFIS network shown above has been programmed in microcontroller. This section explains the techniques and modifications followed in implementing MATLAB-generated ANFIS into an 8-bit microcontroller. To implement ANFIS in microcontroller, generated FIS is

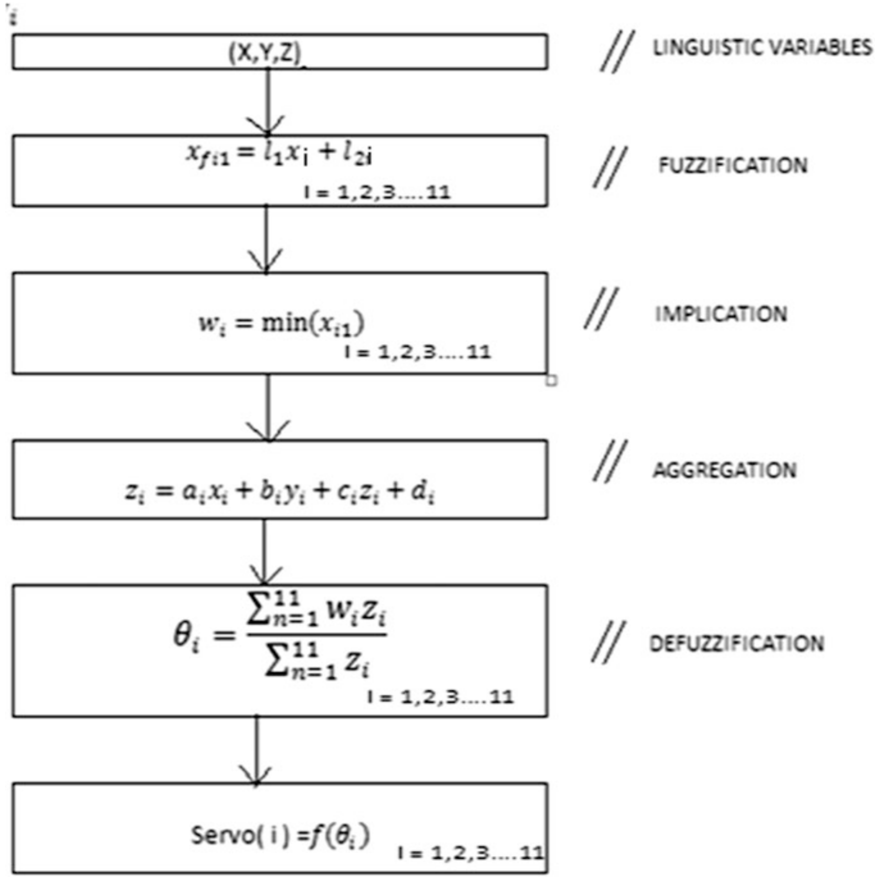


Fig. 8 MicroController implementation of ANFIS flow chart

modified to triangular membership functions [15–17]. Weights of rules can be obtained from the trained data obtained from the SimMechanics simulation. Output equations are de-fuzzified using Sugeno fuzzy inference method and finally the output angles are obtained using the weighted average method [13]. The flow chart for microcontroller implementation of ANFIS is shown in Fig. 8.

8 Simulations and Results

8.1 Simulink Simulation

The glass to mouth posture of a robotic arm has its own features in terms of distances in 3D space environment such as delta X, delta Y, and delta Z,

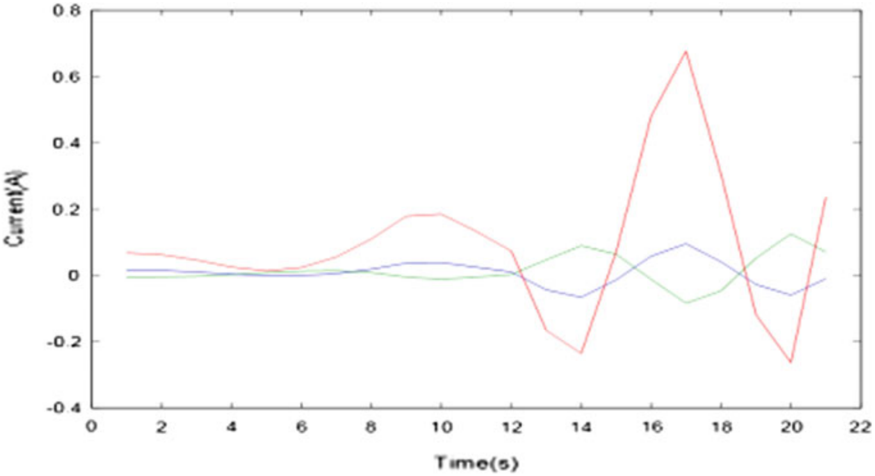


Fig. 9 Current versus time plot obtained at shoulder actuator

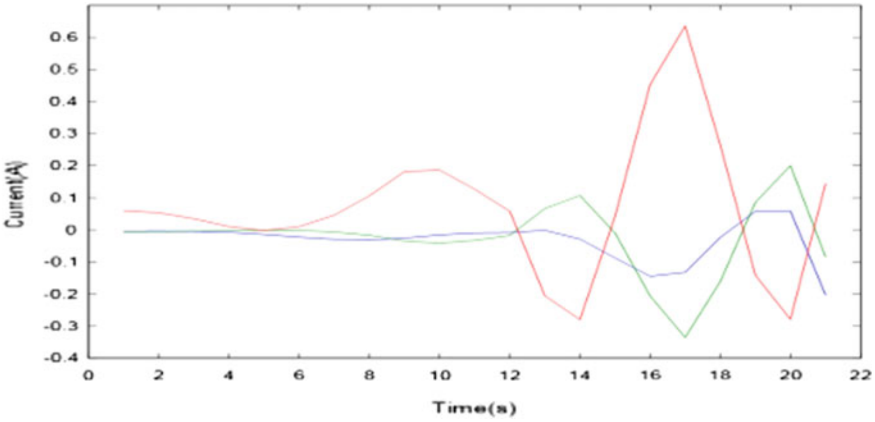


Fig. 10 Current versus time plot obtained at elbow actuator

correspondingly. With Simulink as an interface, the corresponding current vs. time plots were obtained and the power consumption of the arm can be found through these plots. The below-mentioned graphs from Figs. 9, 10, 11 and 12 are the current versus time plots obtained corresponding to the torque sensed by each of the actuators of a robotic arm.

Shoulder Actuator
From shoulder motor to base rotation length
Distance: 10.35 cm
Delta X: 4.76 cm, Delta Y: 8.61 cm, Delta Z: 3.19 cm
Total Length: 36.43 cm

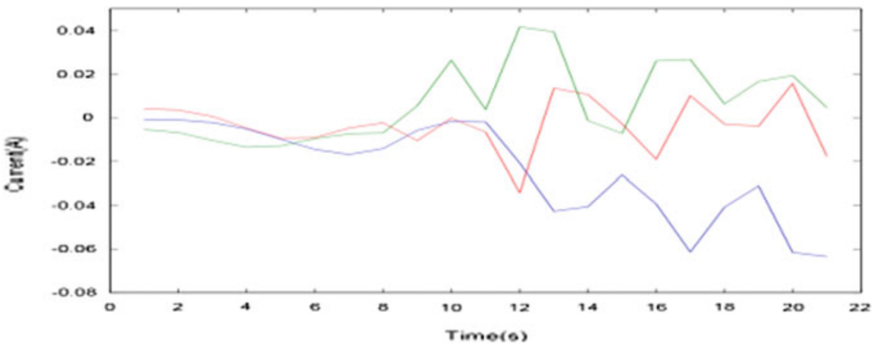


Fig. 11 Current versus time plot obtained at wrist-up actuator

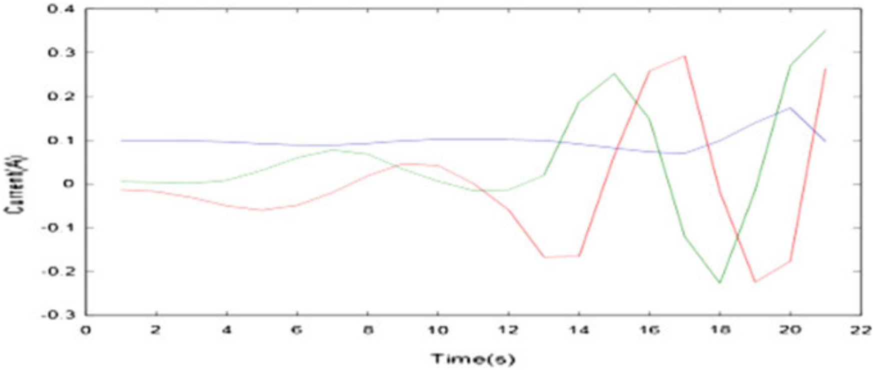


Fig. 12 Current versus time plot obtained at Gripper actuator

Elbow Actuator
From wrist to elbow length
Arc/Circle Measurements Distance: 22.51 cm
Delta *X*: 5.48 cm, Delta *Y*: 2.32 cm, Delta *Z*: 21.7 cm

Wrist-up Actuator
From gripper to wrist length
Arc/Circle Measurements Distance: 12.49 cm Delta *X*: 0.709 cm Delta *Y*: 1.171 cm Delta *Z*: 1.418 cm

Gripper Actuator
Distance: 8.80 cm
Delta *X*: 3.75 cm Delta *Y*: 0.56 cm Delta *Z*: 7.94 cm

8.2 Energy Requirement Durability

With any battery-operated device, energy use is very important. Table 4 shows the power consumption of humanoid robotic arm.

From this power usage table, the total current required for the sample movements posture of a humanoid robotic arm is 9.738 A. While more testing will be instructive, a reasonable estimate is that typical household and office tasks will lead to an average current of 10 Amperes. Therefore, six continuous hours of arm use would consume 12 Ah. This should be acceptable for most users, and daytime charging can help restore range. Overall cost analysis and estimation has been done as show in Table 5.

Table 4 Power consumption

S no	Type of actuator	Maximum current (Ampere)	Total current for each actuator (Ampere)
1	Gripper	0.351	2.0871
2	Wrist rotation	0.1034	0.6067
3	Wrist up	0.0417	0.5019
4	Elbow	0.6374	3.3187
5	Shoulder-A	0.677	1.9641
6	Shoulder-B	0.6	1.2629

Table 5 Cost estimation

S. no	Description	Quantity	Unit cost	Total cost
1	Big gripper (Ax-12 a)	1	9242.34	9242.34
2	Wrist servo motor (Ax-12 a)	2	2807.87	5615.74
3	Elbow servo motor (Mx-6 4 T)	1	13640.21	13640.21
4	Shoulder servo motor (Mx106 T)	1	30076.04	30076.04
5	Turntable (Mx-106 T)	1	38302.96	38302.96
6	FR05-H101k (bracket)	1	1823.9 5	1823.95
7	FR08-H101k brackets	1	1817.17	1817.17
8	Wiring extensions(24 in., 3 pinwire)	1	609.75	609.75
9	Power supply (12 V, 10 amps)	1	1212.46	1212.46
10	Miscellaneous		10000	10000
		Total cost		112340.62

9 Conclusion and Future Works

To sum up, design methodology and implementation of a humanoid robotic arm of 6 DOF with realistic mechanical structure and performance have been obtained. Modeled an arm in solid works based on requirements and corresponding mechanical analysis is done in SimMechanics to obtain the stability. Data required for ANFIS training is obtained by giving all the possible theta to modeled arm kinematics. Practical implementation of moving the robot to the desired position is done using trained ANFIS simulations. A simpler and more attractive way to turn the hand prosthesis into a product is to release it as a mechanical testing and research platform for universities. This type of robotic arm has many advantages such as simple structure, high flexibility, low energy consumption, quiet operation, and sensory feedback which make it a prosthetic arm with very high resemblance to a normal arm. As future works, the following has been taken into consideration.

- The designed robotic arm should be manufactured using 3D printer.
- Remodeling the design by adding 1dof between the shoulder and elbow which is analogous to the human arm.
- Show the device to prosthetics and amputees and get their feedback on the current system.
- Redesign mechanical packaging to further reduce the size and weight of the system
- Orientation-based control algorithms can be developed.

Acknowledgments The authors would like to thank Mechatronics and Intelligence Systems Research Lab, Department of Mechanical Engineering, Amrita University, Amritapuri campus for providing support to carry out the research and experiments.

References

1. Vijayan, A., Medini, C., Singanamala, H., Nutakki, C., Diwakar, S.: Classification of Robotic Arm Movement using SVM and Naïve Bayes Classifiers. Amrita School of Biotechnology, Amrita Vishwa Vidyapeetham, Clappana (P.O.), Kollam, Kerala
2. Industrial and Service Robots, IFR International Federation of Robotics, 2010.: http://www.ifr.org/home
3. Manipulating Industrial Robots—Vocabulary, International Organization for Standardization Standard 8373 (1994)
4. Duc, L.B., Syaifuddin, M., et al.: Designing 8 degrees of freedom humanoid robotic arm. In: International Conference on Intelligent and Advanced Systems, Kuala Lumpur, 25–28 Nov 2007, pp. 1069–1074 (2007)
5. Wang, R.J., Zhang, J.W., et al.: The multiple-function intelligent robotic arms. FUZZ-IEEE J. Korea **20–24**, 1995–2000 (2009)
6. Carignan, C.R., Gefke, G.G., Roberts, B.J.: Intro to space mission design: space robotics. In: Seminar of Space Robotics, University of Maryland, Baltimore, 26 March 2002

7. Elfasakhany, A., Yanez, E., Baylon, K., Salgado, R.: Design and development of a competitive low-cost robot arm with four degrees of freedom. In: Modern Mechanical Engineering, 2011, vol. 1, pp. 47–55
8. Abdallah, C.T., Dawson, D.M., Lewis, F.L.: Robot Manipulator Control: Theory and Practice (2003)
9. Dharmana, M.M., Shashidhar, S., Kumar, S., Chaithanya.: Embedded ANFIS as a supervisory controller for a 6-DOF robotic arm. Int. J. Eng. Res. **3**(5), 318–320
10. Craig, J.J.: Introduction to Robotics: Mechanics and Control, 2nd edn. Addison-wesley Publishing company, New York (1989)
11. Schacherbauer, A., Xu, Y.: Fuzzy control and fuzzy kinematic mapping for a redundant space Robo. Technical Paper, CMU-RI-TR-92–12, Carnegie Mellon University (1992)
12. Araki, S., Nomura, H., Hayashi, I., Wakami, N.: A self-generating method of fuzzy inference rules. In: Proceedings: 1991 IFES Conference, pp. 1047–1058 (1991)
13. Xu, Y., Nechyba, M.: Fuzzy inverse kinematic mapping: rule generation, efficiency, and implementation. Technical Paper, CMU-RITR-93-02, Carnegie Mellon University (1993)
14. Alavandar, Srinivasan, Nigam, M.J.: Inverse kinematics solution of 3DOF planar robot using ANFIS. Int. J. Comp. Commun. Control **3**, 150–155 (2008)
15. Jang, J.-S.R.: ANFIS: adaptive network based fuzzy inference systems. IEEE Trans. Syst. Man Cybern. **23**(03), 665–685 (1993)
16. Jand, J., Sun, C., Mizutani, E.: Neuro Fuzzy and Soft Computing. Prentice-Hall, Upper Saddle River (1997)
17. Sadjadian, H., Taghirad, H.D., Fatehi, A.: Neural Networks approaches for computing the forward kinematics of a redundant parallel manipulator. Int. J. Comput. Intell. **2**(1), 40–47 (2005)

Preliminary Study in a Novel Robotic-Assisted Femoral Shaft Fracture Reduction System

Qing Zhu, Xingsong Wang, Xiaogang Sun and Bin Liang

Abstract The minimally invasive surgical operation of intramedullary nailing has prevailed as a standard technique for a definite stabilizing treatment in femoral shaft fractures, and reduction is a critical step before carrying out an invasive operation. However, there are some disadvantages about this technology, such as the frequent radiation exposure for the operator and patients. This paper describes a novel cuff-type robotic-assisted femoral shaft fracture repositioning system, and preliminary discusses the constitutive modeling of passive transverse mechanical properties of skeletal muscle. Skeletal traction and fragments alignment for femur fracture reduction can be performed separately, which are standard procedures for surgeons in clinical. Correspondingly, the robotic system consists of distraction device, cuff-type reduction unit and remote control system. Preliminary experiments based on artificial models have been done, and the results show the effectiveness of the proposed robotic-assisted system, and indicate the potential value for further study.

Keywords Fracture reduction robot · Cuff-type unit · Skeletal muscle · Compressive behavior

1 Introduction

Recently, the concept of Biological Osteosynthesis is a focus in fracture reduction surgery, which emphasizes the equilibrium on fracture stability and soft-tissue integrity, considering on biological characteristics of the bone, free of interruptions

Q. Zhu · X. Wang (✉) · X. Sun
School of Mechanical Engineering, Southeast University, Nanjing 211189, China
e-mail: xswang@seu.edu.cn

Q. Zhu
e-mail: seu_andrew@163.com

B. Liang
Department of Orthopedics, Nanjing First Hospital, Nanjing 210006, China

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_23

for the original physiological environment of bone growth and fracture healing [1, 2]. Intramedullary nailing has been the gold standard technique in the management of lower limbs fractures with satisfactory treatment results [3–5]. Before executing an intramedullary nail surgery, fracture fragments are repositioned to reduce the displacement along the femoral shaft axis assisted by traction instruments. Then surgeons insert the guide wire and implant the Schanz screws when the temporary fragments alignment is completed by manual manipulation. However, only the experienced surgeon can acquire satisfactory consequences, and maintaining the better reduction status for a long time is still a challenge. Additionally, since the operation is carried out under the observation of the C-arm X-ray machine, so the risk of radiation exposure is high for both the medical staff and patients. Inspired by the robotic-assisted surgery as De Vinci in the surgical area [6], numerous scholars began to study the instrument assistant long bone fracture reduction skill. Bouazza-Marouf et al. first introduced the robotic-assisted fracture reduction in 1995 [7] and since then several experimental robotic systems have been designed and applied in the practice.

Westphal et al. introduced a robotic surgical telemanipulator for femoral shaft fracture reposition [8]. They developed a manipulator based on an industrial robot Stäubli (model RX 90), and created the needed joint values to control the end effector moving to the target position with its built-in CS7B controller unit. The operator could guard the repositioning procedure with haptic feedback by the equipped force/torque sensor in the end effector. The surgeon can control the movement of the robot system from an input console directly on base of personal computer, where the manipulator ran at a master–slave mode. Seide et al., using hexapod robot kinematics, have been conducting extensive research in developing a robotic-assisted fracture reduction system [9]. Füchtmeier et al. also developed a femur fracture reduction robot system based on an industrial robot Stäubli (model RX 130) [10]. The surgeon can also monitor the reduction forces and torques with a six-axis force-torque sensor equipped in the robot during the reduction process. Hung et al. described a robotic system, with the additional features of automatic controlled flexion of the knee joint, individual traction of thigh and leg, and foot rotation [11]. Mukherjee et al. have designed a parallel robot for femoral shaft fracture reduction based on the Stewart platform. The robot was comprised of two ring plates (ground plate and mobile plate), which were connected by six linear actuators and each actuator can generate driving force of 400 N. The force–torque model can also allow the reduction path to be planned and probed in three-dimensional while examining the positon and expected desired forces involved [12, 13].

In the paper, we first describe the mechanical structure of the novel robotic-assisted fracture reduction system, whereafter introduce the control model based on the constitutive modeling of passive transverse mechanical properties of skeletal muscle. Finally, preliminary experiments on artificial models were performed to testify the feasibility of the setup. Results show the effectiveness of the proposed robotic-assisted fracture reduction system, and indicate the potential value for further research development. To our knowledge, no automated robotized executions of long bone fracture reduction without invasive to the patients based on transverse mechanical properties of the muscle have been published so far.

2 Mechanical Design

2.1 Musculoskeletal Interaction

The femur is well protected with powerful muscles, and the forces by the muscle contraction protect the femur from most external damage. However, the force can also cause serious fracture displacement and make reduction much more difficult. The surgical treatment of broken limbs often necessitates applying appropriate force to displace one or both ends of a broken bone against the natural forces of muscles and tendons [10]. Six degree of freedom (DOF) should be adjusted to complete the reduction of femoral shaft fracture, and traction is the most important. The displacement parameters can be obtained from the lateral and anterior/posterior (AP) views (Fig. 1).

The process of fracture reduction consists of several consecutive steps. First, bone fragments are distracted from one another to release the displacement along the femoral shaft axis assisted by traction instruments. Then temporary fragments alignment is finished to insert the guide wire and implant the Schanz screws. Only the well qualified surgeon can obtain better reduction performance, and maintaining the satisfactory reduction state for a long time is another challenge. In view of that, we introduce a novel robotic-assisted fracture system based on the principle of manual reduction. Figure 2 shows the fracture fragments alignment process in 3-DOF (x, y, ϕ).

2.2 Manipulator Robot

We have developed a measuring system to obtain the forces required during the process of femoral fracture reduction in previous studies. The measurement and

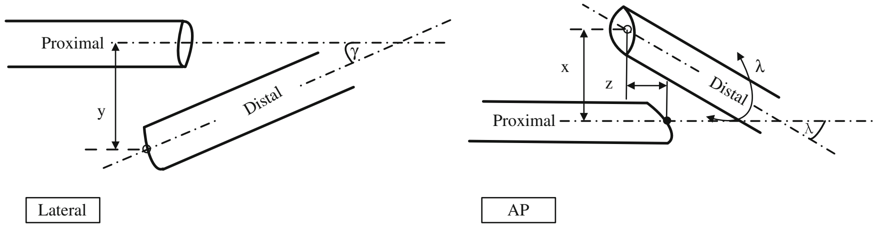


Fig. 1 Displacement parameters in lateral and AP views

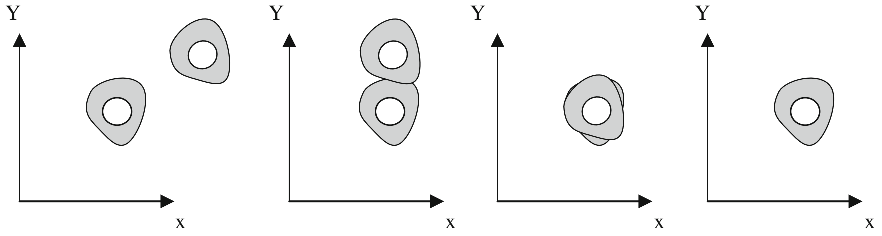


Fig. 2 Fracture fragments alignment process

analysis of the reduction mechanical parameters provides theoretical basis for the development of fracture reduction robot and procedure planning. According to the test results, the maximum measured peak force along the shaft axis in distraction could be up to 379.7 N, and the maximum alignment force in other two axes directions are 48.4 and 240.1 N [14], respectively. It is difficult for a compact surgical robot to provide the necessary power. In view of the surgeon's experience, skeletal traction and fracture fragments alignment are achieved, respectively, [15] in our robotic system. The robotic-assisted fracture reduction system includes a distraction apparatus to complete the operative distraction (Fig. 3), a cuff-type

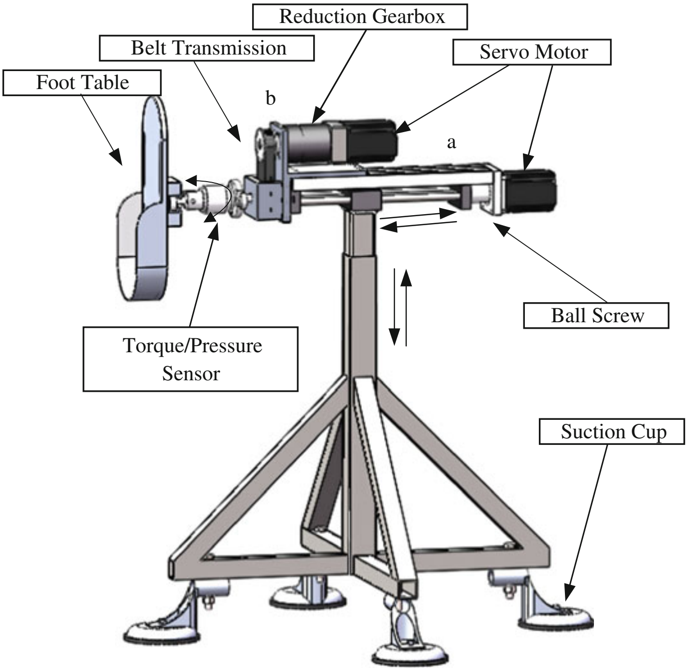


Fig. 3 Device for distraction

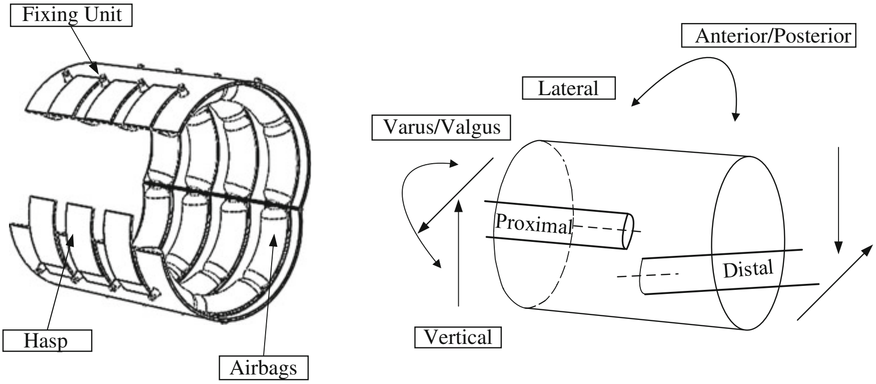


Fig. 4 Fragments alignment mechanism

repositioning unit to manipulate the fracture fragments for alignment (Fig. 4) and a control system to achieve the performance.

Distraction Device. The distraction device comprises a ball screw (a in Fig. 3) and a gear pair (b in Fig. 3), with direct driving by two servo motors (Yasukawa Electric (China) Company Ltd.), respectively. The location and height of the setup can be manually regulated to fit for various patients. Its location and height can be manually adjusted to fit for various patients. Displacement distance of the ball screw is 300 mm, which is enough for the distraction along the femoral shaft axis. The technical parameters of the servo motor are shown as follows: the maximum rotational speed n is 3600 r/min, the rated torque M is 0.637 N · m. The maximum force and torque are 900 N and 11.5 N · m, respectively. The result shows that the equipment can export sufficient forces for repositioning in these 2-DOF (z, ϕ).

Fragments Alignment Unit. The cuff-type repositioning unit consists of a constituent of airbags, an air compressor, and a fixed unit (Fig. 4). The airbag is molded by thin latex materials (1 mm). And each airbag is covered with tensile fabric, which can protect the airbag from bursting with an extra inflation and facilitates the fixation. The force-air pressure relationship of the airbags indicates a linear relationship with different inflation displacement, and the linear coefficient decreases with the displacement of the latex materials increasing. The capacity of the device meets the requirements for fragments alignment.

Air pressure sensors are installed between the airbags and air compressor for real-time monitor of the force applied on the long bone fragments. The air compressor can supply the system with a maximum 0.3 MPa pressure, which is the power source for the fracture fragments repositioning in other four DOFs. The airbags fixation unit is made of X-ray permeable carbon fiber material. Carbon fiber is a material consisting of fibers about 5–10 μm in diameter and composed mostly of carbon atoms. The properties of carbon fibers, such as high stiffness, high tensile strength, low weight, low thermal expansion, and free of X-ray, make them very popular in aerospace, civil engineering, military, and motorsports. Carbon fibers are usually combined with other materials to form a composite. Fabric made of woven carbon filaments is used in this research. Accordingly, a personal computer is used for trajectory planning, signal acquisition of the transducers, and controlling the robot system.

2.3 Transverse Mechanical Properties of Skeletal Muscle

Virtual modeling of the human body in compression are widely used in impact biomechanics [16]. Skeletal muscle presents a complex fiber-oriented structure consisting of about 80 % water, 3 % fat and 10 % collagenous tissues [17]. It therefore displays a viscoelastic behavior and has anisotropic properties. Finite element models can be used to predict deformations during transient loading [18]. These models require a good understand of the tissue mechanical properties.

Bosboom et al. performed experiments to determine the passive transverse mechanical properties of skeletal muscle in vivo, and a plane stress model of the cross section was developed for each muscle to assess the stress–strain distributions [19]. Van Loocke et al. also presented the elastic and viscoelastic properties of passive skeletal muscle using uniaxial unconfined compression tests performed on fresh orcine muscle tissue in various loading rates in vitro, and a model was developed to represent the properties depending on the observed experimental data [17, 18, 20]. Aifantis et al. published experimental data on transverse mechanical properties of collagen fibers from nanoindentation [21]. It was reported that the elastic modulus was 63 ± 4 MPa, while the viscosity was 14 GPa–56 GPa s.

However, the robotic system presented in this paper will be used in long bone fracture patients. Relaxants are injected before the reduction start in the operation, which may affect the transverse mechanical properties of the muscle. Transverse mechanical properties of muscles with relaxants injection should be discussed in our future work.

3 Results

Five preliminary experiments have been carried out with interested artificial fracture models, without regard to the transverse mechanical properties of muscles due to time constraints and limited model preparation. The errors in x, y, ϕ directions were measured by the endoscopic placed in the marrow cavity of the artificial models, and the error in z direction was measured by the camera recorded from the side (Fig. 5). According to the results of simulate fracture reduction tests, the axial displacement, the lateral displacement, the vertical displacement, and rotation angular were measured. Figure 6 shows the reduction errors and statistics of the tests. The axial, lateral, and vertical maximum errors were 2.5, 2.0, and 2.6 mm.

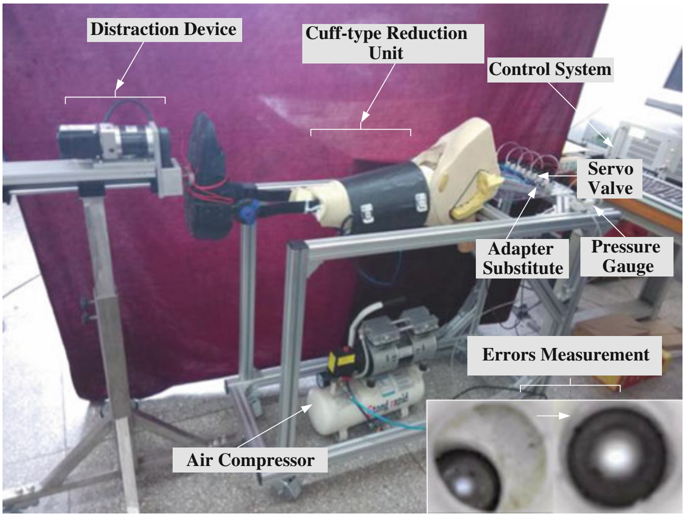
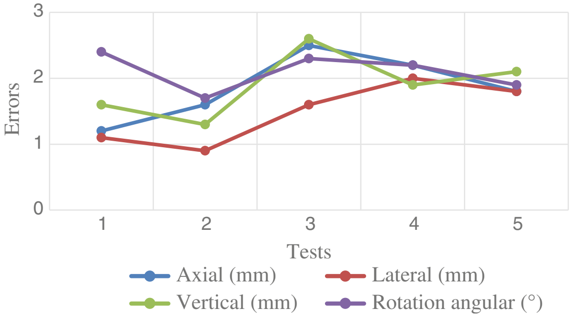


Fig. 5 Preliminary experiments of robot-assisted femoral shaft fracture reduction on artificial models

Fig. 6 Position errors



The maximum error of rotation was 2.4°. Obviously, the results in line with the requirements of the medical standard and demonstrates acceptable precision of this robotic system.

4 Conclusion and Future Work

Resistance of soft-tissue contraction force is the key problem in obtaining and maintaining alignment status. Nonetheless, excessive traction which leads to extension and distraction of distal fragment is undesirable. Repetitive manipulations or distraction of the proximal/distal fragments may lead to extra soft-tissue trauma.

Major soft-tissue injury correlates with the development of systemic inflammatory response syndrome and multisystem organ failure [22]. Considering the surgeon's clinical practice, skeletal traction and fracture fragments alignment are performed separately in our robot system. The robot system mainly consists of a distraction device to accomplish the preoperative distraction and rotation displacement of the circumferential of the femur shaft, a cuff-type reposition unit to manipulate the proximal/distal fragments for alignment. Furthermore, these two mechanical structures are adjustable to fit for various patients.

The goal of this research is to develop a robotic-assisted fracture reduction system to liberate the surgeon from the operating room. The repositioning process of the setup is exactly accordance with the surgeon does in clinical. The distraction part consists of a linear stage and a gear pair, actuated by two servo motors, achieving the axial displacement and circumferential rotation displacement, respectively. The cuff-type reduction unit, complying with the gravity and muscle forces, manipulates the fracture fragments for repositioning. Airbags contraction forces provide lateral/vertical forces and torques, completing reset on the other four DOFs. The procedure of the operation is controlled by the flexible control system in combining with endoscope imaging. With this auxiliary system, it is feasible to obtain accurate reduction of the fracture in all directions under visible control. The results show the effectiveness of the proposed robotic-assisted system, and indicate the potential value for further study.

Although the noninvasive robot-assisted fracture system is proved to be advantageous for those superiority mentioned above, such as little radiation exposure to medical staff and patients, safety, enough power, and fracture repositioning accuracy. There are still number of limitations. First, experiments have been performed without regard to the transverse mechanical properties of muscles due to time constraints and limited model preparation. Second, only the middle shaft femoral fractures can the novel robotic system be adopted. Finally, artificial lower limbs models without muscles covered were used for the tests, which cannot reflect the real situation during the operation. In our future work, particular attention is placed on transverse compressive behavior of muscle with relaxants injection, as this is the relevant deformation mode in human/machine impacts, and then develop the constitutive model of the muscle to ultimately manipulating the remote control system. Moreover, the setup will be applied in clinical practice to further verify the feasibility, and further research is necessary to subsequent apply intramedullary nails when the alignment status is achieved with the assistance of robot.

Acknowledgements This research was supported by the Fundamental Research Funds for the Central Universities, the Graduate Student Scientific Research Innovation projects of General University of Jiangsu Province (No. KYLX_0101), and the Fundamental Research Funds for National Health and Family Planning Commission of P. R. China (W201307). The authors would like to thank the medical stuff of the Orthopaedic Trauma department of Nanjing First Hospital, who provided us an opportunity to observe the whole operation process personally, allowing us to know the surgery better. We also thank the Medical imaging department for performing the analysis with constructive discussions.

References

1. Gerber, C., Mast, J.W., Ganz, R.: Biological internal fixation of fractures. Arch. Orthop. Trauma Surg. **109**, 295–303 (1990)
2. Palmer, R.H.: Biological osteosynthesis. Vet. Clin. North. Am. Small Anim. Pract. **29**, 1171–1185 (1999)
3. Winquist, R.A., Hansen, S.T., Clawson, D.K.: Closed intramedullary nailing of femoral fractures: a report of five hundred and twenty cases. J. Bone Joint Surg. Am. **66**, 529–539 (1984)
4. Kempf, I., Grosse, A., Beck, G.: Closed locked intramedullary nailing: its application to comminuted fractures of the femur. J. Bone Joint Surg. Am. **67**, 709–720 (1985)
5. Cleather, D.J., Goodwin, J.E., Bull, A.M.J.: Hip and knee joint loading during vertical jumping and push jerking. Clin. Biomech. **28**, 98–103 (2012)
6. Warisawa, S., Ishizuka, T., Mitsuishi, M., Sugano, N., Yonenobu, K., Nakzawa, T.: Development of a femur fracture reduction robot. In: Proceedings of IEEE International Conference on Robotics and Automation, New Orleans, USA, pp. 3999–4004 (2004)
7. Bouazza, M.K., Browbank, I., Hewit, J.R.: Robotic-assisted internal fixation of femoral fractures. J. Eng. Med. **209**, 51–58 (1995)
8. Westphal, G.R., Winkelbach, S., Wahl, F., Gösling, T., Oszwald, M., Hufner, T., Krettek, C.: Robotassisted long bone fracture reduction. Int. J. Robot. Res. **28**, 1259–1278 (2009)
9. Seide, K., Faschingbauer, M., Wenzl, M. E., Weinrich, N., Juergens, C.: A hexapod robot external fixator for computer assisted fracture reduction and deformity correction. Int J. Med. Robot. Comput. Assist. Surg. **1**, 64–69 (2004)
10. Füchtmeier, B., Egersdoerfer, S., Mai, R., Hente, R., Dragoi, D., Monkman, G., Nerlich, M.: Reduction of femoral shaft fractures in vitro by a new developed reduction robot system 'RepoRobo'. Injury Int. J Care Injured **35**, 113–119 (2004)
11. Hung, S.S., Lee, M.Y.: Functional assessment of a surgical robot for reduction of lower limb fractures. Int. J. Med. Robot. Comput. Assist. Surg. **6**, 413–421 (2010)
12. Mukherjee, S., Rendsburg, M., Xu, W.L.: Surgeon-instructed, image-guided and robot-assisted long bone fractures reduction. In: Proceedings of 1st International Conference on Sensing Technology, Palmerston North, pp. 78–84 (2008)
13. Graham, A.E., Xie, S.Q., Aw, K.C., Xu, W.L., Mukherjee, S.: Robotic long bone fracture reduction medical robotics, In: Bozovic, V. (ed.) Medical Robotics, I-Tech, Vienna, pp. 85–102 (2008)
14. Zhu, Q., Sun, X.G., Wang, X.S., Wu, Q.C., Liang, B.: Development of intraoperative noninvasive force measuring system during femoral fracture reduction. In: Proceedings of 10th Annual IEEE International Symposium on Medical Measurements and Applications, Italy, pp. 335–339 (2015)
15. Ye, R.H., Chen, Y.H., Yau, W.P.: A simple and novel hybrid robotic system for robot-assisted femur fracture reduction. Adv. Robot. **26**, 83–104 (2012)
16. Pietsch, R., Wheatley, B. B., Haut Donahue, T. L., Gilbrech, R., Prabhu, R., Liao, J., Williams, L. N.: Anisotropic compressive properties of passive porcine muscle tissue. J. Biomech. Eng. **136**(11) (2014)
17. Van Loocke, M., Lyons, C.G., Simms, C.K.: Viscoelastic properties of passive skeletal muscle in compression: Stress-relaxation behaviour and constitutive modelling. J. Biomech. **41**, 1555–1566 (2008)
18. Van Loocke, M., Lyons, C.G., Simms, C.K.: A validated model of passive muscle in compression. J. Biomech. **39**, 2999–3009 (2006)
19. Bosboom, E.M.H., Hesselink, M.K.C., Oomens, C.W.J., Bouten, C.V.C., Drost, M.R., Baaijens, F.P.T.: Passive transverse mechanical properties of skeletal muscle under in vivo compression. J. Biomech. **34**, 1365–1368 (2001)

20. Van Loocke, M., Simms, C.K., Lyons, C.G.: Viscoelastic properties of passive skeletal muscle in compression—cyclic behavior. J. Biomech. **42**, 1038–1048 (2009)
21. Aifantis, K.E., Shrivastava, S., Odegard, G.M.: Transverse mechanical properties of collagen fibers from nanoindentation. J. Mater. Sci. Mater. Med. **22**, 1375–1381 (2011)
22. Levy, R.M., Prince, J.M., Yang, R., Mollen, K.P., Liao, H., Watson, G.A., Billiar, T.R.: Systemic inflammation and remote organ damage following bilateral femur fracture requires Toll-like receptor 4. Am. J. Physiol. Regul. Integr. Comp. Physiol. **291**(4), R970–R976 (2006)

Pet Care Robot for Playing with Canines

Chan Woo Park, Jong-Hyeon Seon, Jung-Hoon Kim and Jong-Hwan Kim

Abstract This paper observes interactions between a robot and pets, especially canines, and also investigates the feasibility of pet care robot (PCR) as an alternative for human in terms of pet caring based on animal–robot interaction (ARI) with experiments. The paper demonstrates the effectiveness of PCR for playing with pets, which is designed based on ARI, ethology, and user-centered approaches. The experiments were performed to observe the interactions in two conditions: when the dogs meet the robot at the first time and when the dogs already meet the robot for some times. The result summarizes that canines are able to show better interaction performances with robots through repetitive learning with their owners. Furthermore, this result also suggests the feasibility of PCR as an alternative for human pet sitter in terms of the playing function.

1 Introduction

Pets, especially canines, take important roles in modern life. With the level of interest about raising a pet at home, various academic approaches to verify influences of pets on human and society have been reported. A recent study argued that having a pet results in positive effects on physical health status, social participation, and contribution on social capital accumulation [1]. However, there were some observations

C.W. Park (✉) · J.-H. Seon · J.-H. Kim · J.-H. Kim
The School of Electrical Engineering, KAIST 291 Daehangno, Daejeon,
Yuseong-gu 305-701, Republic of Korea
e-mail: cwpark@rit.kaist.ac.kr

J.-H. Seon
e-mail: johseon@kaist.ac.kr

J.-H. Kim
e-mail: jhookim@kaist.ac.kr

J.-H. Kim
e-mail: johkim@rit.kaist.ac.kr

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_24

that dogs that are left alone suffer from separation anxiety syndromes [2]. Therefore, there is a need for development of goods and research about caring pets when owners are away from home for their trips.

With a long history of domestic dogs, human–dog interactions have been reported naturally. Beginning with a report by Romanes in 1883 about human–dog interaction, research on cognition of dogs and studies on animal–human interaction (AHI) have been initiated [3]. As a demand for pet sitting increases, a shift from AHI to animal–computer interaction (ACI) or animal–robot interaction (ARI) gains importance in AHI research field. This paradigm shift is connected to realization of PCR, which observes status of the human's partners and looks after them, as an alternative for human pet sitter.

There have been several studies to implement PCR for various kinds of pets. Lakatos measured interactions between dogs and humanoid robot, which performs gestures like pointing, and suggested design method that helps robots be more social to dogs [4]. Another research sought for new interplaying ways by performing experiment, whether dogs can react to the information gained from the TV screen [5]. Kim first defined the term 'ARI', proposed qualifications for PCR, and suggested three functions of PCR: 'Remote Control', 'Playing function', and 'Caring function' [6]. All of these studies aimed to learn how robots can interact effectively and ethically with pets [7].

Based on concepts of ARI and PCR, this paper demonstrates design approaches for playing functions on PCR with experiments and observes differences in dogs' behavior when dogs meet a PCR never seen before and when they meet the PCR after accustomed to it. To implement the robot for playing function in PCR, design methods based on ARI, ethological knowledge, and user-centered design approaches are used for the robot design.

This paper is organized as follows. Section 2 introduces ethological backgrounds and design approaches using concepts of ARI and PCR. Section 3 shows experimental procedures and results. Section 4 provides discussion on the interpretation to the results and future works.

2 Materials and Methods

2.1 Background

Until now, most of research field focused only on human-centered robots. The PCR concept, however, argues that PCR should show different features from characteristics of human-centered robots and suggests considerations on ethology, breed, characteristics, and size of pets. These should be implied in hardware and behavior design. In this section, ethological backgrounds on plays in dogs and design approaches for PCR-based on ARI are introduced.

Play in dogs Nowadays, most ethologists agree that plays include specific distorted behavior patterns that deviate from their original functions and goals [8]. For social mammals that have complex behavior patterns, play offers promotions in behavioral routines, physical, mental exercises, and enforcement on individual relations [9]. Dogs often use playing for building hierarchical relationships [8]. Moreover, plays also have a role as a 'training for the unexpected [10]' finding individual fits for a new challenge under the safety provided by parents or groups [11].

There have been some reports about playing of dogs. Rooney showed that dogs prefer human–dog plays rather than dog-dog plays in object plays because they do not have to compete with human [12]. Furthermore, there was a report about comparing differences in human–dog, social robot–dog, and asocial robot–dog interactions. The report suggested that as more and more the robot become social or manlike, dogs can understand its intention better and be more interactive with it [4].

ARI and PCR In this paper, the robot that plays a role as playing function in PCR is used for showing feasibility of PCR as an alternative for human pet sitters in terms of playing with pets. ARI concept is inspired from Human–Robot Interaction (HRI) concept. It is defined as interactions between animal and robot without any involvement or interference of human. PCR is defined as a robot that interacts with a pet based on its current state for the purpose of pet caring [6].

To be PCR, the robot should satisfy some necessary conditions of ARI. For playing function, required conditions can be summarized as follows:

1. The robot should satisfy ethological playing function.
2. The robot should be familiar with pets in terms of appearances, noises, and scents.
3. The robot should not give fear to the pets. Thus its behavior patterns should be based on 'user-centered' approaches [7].

2.2 Structures

For PCR, the robot should satisfy three necessary conditions that were introduced in above section. To meet those conditions, the robot for the experiment is designed as follow. The robot is designed to provide playing function that is one part of the PCR functions, adopting 'user oriented service providing' method [6]. It means the robot in this experiment is charged only on a playing function, so the robot has no autonomous feedback systems and sensors for vision and auditory. Thus, in this experiment, humans are responsible for observation on current state of pets and giving feedbacks into the robot via watching dogs in separate areas and remote controlling using smartphone.

The spherical robot connected with network through Wi-Fi, which can be controlled via a smartphone application based on android OS, is used for the experiment. To cover noises of motors and attract concerns of dogs, speaker broadcasts voices of the owner. Figure 1 shows the appearance of the robot and Fig. 2 summarizes the overall system.

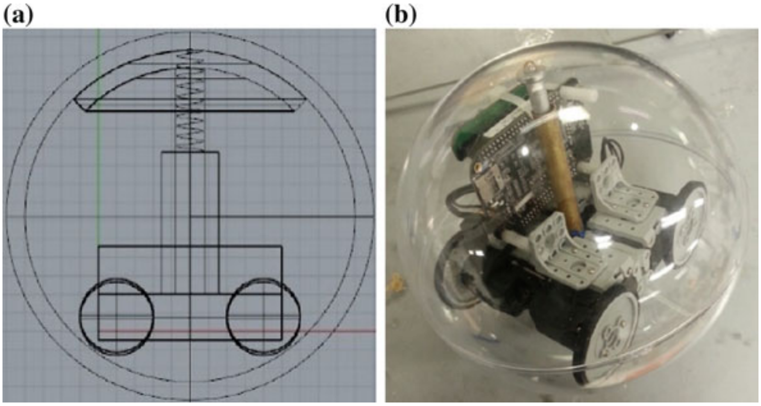
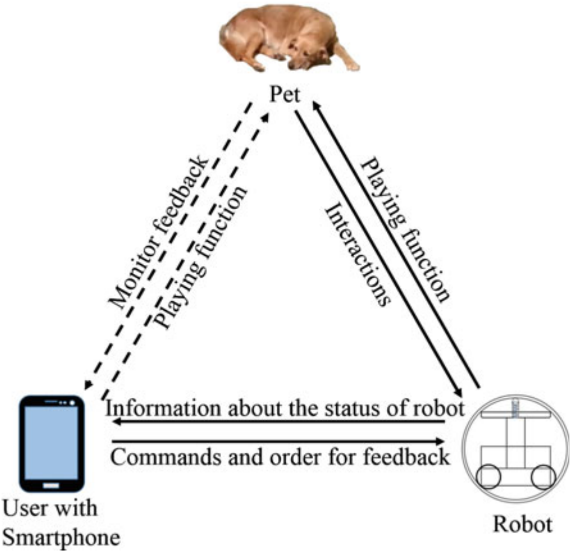


Fig. 1 Design of the robot: **a** Simple structural modeling of the robot. **b** Implementation of the robot without sensors

Fig. 2 Overall system



Hardware The robot is covered by acrylic shell with a diameter of 180.0 mm, and composed of three layers: lower layer, middle layer, and upper layer. Lower layer is responsible for movement, power supply, and keeping balance of the robot. Besides, it contains four DYNAMIXEL AX-18 as motors, a battery, a weight pendulum and plastic frames. Middle layer is composed of an usb hub, a microprocessor (beagle bone black), a Wi-Fi dongle, a sound card, a speaker, and an inertial measurement unit (IMU). Finally, the upper layer takes charge of supporting overall structure with suspension systems and a damper.

Table 1 Short overview of the experimental subject

Index	Breed	Classification	Characteristic index	Sex
1	Golden retriever	Large-sized	5—intimacy, like physical touch	F
2	Golden retriever	Large-sized	3—calm, mild	M
3	Pembroke welsh corge	Medium-sized	3—careful, smart	M
4	Mongrel	Medium-sized	2—timid, cowardly	F
5	Chihuahua	Small-sized	4—outgoing, curiosity	F

Software Movements of the robot are remotely controlled via Wi-Fi networks using an application based on android smartphone. Users can give appropriate commands through the smartphone to manipulate speed, control, automatic navigation, and voice broadcast of the robot. The processor, which receives IMU measurement values, helps the robot to maintain its balance and change its direction to desired one.

2.3 Subjects

For the experiment, subjects are chosen by the following standards:

1. They should have various sizes, breeds, and characteristics.
2. They should have similar experiences in playing.
3. They should have a similar amount of repulsion or familiarity with the objects which are used in object plays.

Thus, five different dogs which are familiar to globelike toys and have the same owner were chosen for subjects of the experiment with an assumption that if they belong to the same owner, they have similar experiences on playing with the same objects. Through the observation before the experiment, personality indexes were marked as 1 for very inactive to 5 for very active. Table 1 summarizes specifications of the subjects.

3 Experiment

The results are marked as average values from grades of three observers. The observers gave grades 0 to 10 for interactions between a dog and the robot. The intention of the experiment is to find answers for the following two questions: *'Can dogs interact with the robot which they never saw before?'* and *'Can dogs interact with the robot after accustomed to it?'*.

3.1 Procedures

Procedure I As a first step, the observers watch the interactions between subjects and the robot which is switched off. This procedure is executed for providing enough time to the subjects for getting accustomed to the experimental environments and smell of the robot.

Procedure II By not providing any information to subjects, the robot gets into close touch with subjects broadcasting voice of the owner through a speaker for 8 min. In this procedure, the observers mark points for each subject how the dogs interact with the robot and record behavior of each dog. The purpose of this step is to know how dogs behave when they meet the robot that they never saw before, but a familiar appearance robot which broadcasts the voice of the owner.

Procedure III In the third procedure, owner introduces the robot to the subjects. To show harmlessness of the robot to the dogs, the owner shows each function of it such as moving or voice broadcasting three times to the subjects. This step lasts for 10 min.

Procedure IV Last procedure lasts for 8 min. Similar to the second procedure, the robot approaches the subjects with broadcasting the owner's voice. From this step, observers give points for the interactions and record differences between the results from this procedure and the results of the Procedure II.

3.2 Results

To answer the suggested questions, the results concentrate on comparing the results when dogs are not accustomed to the robot (Procedures I and II) and accustomed to the robot (Procedures III and IV). For Procedure I, all subjects were not so interested in the robot. They showed typical playing patterns: playing with their tails or chasing each other. By moving and broadcasting the voice, however in Procedure II, subject 1 and 2 showed strong curiosity, and responded to the motions and voices of it. Subject 3 reacted with wary investigation to determine whether the robot is a threat or not. Subject 4 and subject 5 showed strong wariness, and subject 5 failed to interact.

For Procedure III, as the owner showed the robot is harmless, all subjects seemed comfortable, but they displayed a tendency on concentrating to their owner, rather than the robot. In the last procedure, all subjects showed moderate wariness and better performances in interactions with the robot. Subject 3 and 4 reacted more actively in this procedure compared to Procedure II and subject 5 still showed hesitation on making contact with the robot.

Table 2 summarizes the results. The improvement is measured as differences in interaction factor between after learning process (Procedure IV) and before the process (Procedure II) divided by the former factor. The factor of Procedure III have no influence on the improvement, since existence of the owner effects on the interactions.

Table 2 Summary of the results

Index	Procedure II	Procedure III	Procedure IV	Improvement (%)
1	8.67	10.00	9.33	7.07
2	6.33	9.33	8.33	24.01
3	3.67	8.33	6.67	44.98
4	2.33	6.00	5.00	53.40
5	0.33	3.67	1.67	80.24

There are two key points for the experiment. One is verifying the feasibility of PCR as a substitutive pet sitter in the absence of the owner at home, in terms of the playing function based on ARI. The other is observing interaction capabilities of canines with the robot never saw before and differences in interactions with nonacquaintance robot and acquaintance one.

The result shows that canines can interact with a robot never seen before; however, they showed different degrees in interaction performances. Besides, it also revealed that canines outperformed after they learn that the robot is harmless. Thus, the result from the experiments suggests the feasibility of PCR as an alternative for human pet sitters in terms of playing with pets.

4 Conclusion

This paper demonstrated a design method for PCR for playing with canines. The robot was designed based on ARI, ethology, and user-centered approaches with experiments. Through the experiments, it was revealed that PCR can be the alternative of human pet sitter when the owner is absent, in terms of playing function. Besides, the experiments also implicate that by learning that the robot was harmless, canines showed better performances in interaction with it. It infers that PCR can become more effective as an alternative for pet sitter with repetitive learning with their owners. The results also reveal that dogs which have smaller sizes than the robot or careful characteristics cannot adapt to the robot easily. Therefore, the PCR should be designed small enough to give no fear to dogs considering the breeds and characteristics. Since of lack of sensory data, the experiment only tested the playing function of PCR. As a furtherwork, it is needed that testing overall functions of PCR including caring function as well.

Acknowledgments This work was supported by a grant from the National Research Foundation of Korea (NRF) funded by the Korea government (MSIP) (No. NRF-2014R1A2A1A 10051551).

References

1. Wood, L., Giles-Corti, B., Bulsara, M.: The pet connection: pets as a conduit for social capital? Soc. Sci. Med. **61**(6), 1159–1173 (2005)
2. Voith, V.L.: Separation anxiety in dogs. The Compendium on Continuing Education for the Practicing Veterinarian (USA) (1985)
3. Romanes, G.J.: Animal Intelligence, vol. 44. Appleton (1883)
4. Lakatos, G., Janiak, M., Malek, L., Muszynski, R., Konok, V., Tchon, K., Miklósi, Á.: Sensing sociality in dogs: what may make an interactive robot social? Anim. cogn. **17**(2), 387–397 (2014)
5. Hirskyj-Douglas, I.: Is my dog watching tv? In: Animal Computer Interaction. NordiCHI'14 (2014)
6. Kim, J.-H., Choi, S.-H., Kim, D., Kim, J., Cho, M.: Animal-robot interaction for pet caring. In: 2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), pp. 159–164. IEEE (2009)
7. Mancini, C.: Animal-computer interaction: a manifesto. Interactions **18**(4), 69–73 (2011)
8. Miklósi, Á.: Dog Behaviour, Evolution, and Cognition. Oxford University Press (2014)
9. Immelmann, K.: Behavioral Development: the Bielefeld Interdisciplinary Project. CUP Archive (1981)
10. Gould, J.: The quarterly review of biology. Quart. Rev. Biol **51**(2), 211–244 (1976)
11. Burghardt, G.M.: The Genesis of Animal Play: testing the Limits. MIT Press (2005)
12. Scott, J.P., et al.: Animal behaviour. Anim. Behav. (1958)

Part II
Computational Intelligence and Intelligent Design for Advanced Robotics

Making a More Reliable Classifier via Random Crop Pooling

Yeakang Lee, Jiwhan Kim, Minju Jung and Junmo Kim

Abstract Deep neural networks have achieved state-of-the-art performance for a variety of pattern-recognition tasks. In particular, the deep convolutional neural network (CNN), which is composed of several convolutional layers with a nonlinear activation function, pooling layers, and fully connected layers or an optional global average pooling layer, has received significant attention and is widely used in computer vision. Some research is now replacing a top fully connected layer with global pooling to avoid overfitting in the fully connected layers and to achieve regularization. This replacement is very important because global pooling with additional convolutional layers can eliminate restrictions on the necessity for fixed-size or fixed-length input in the fully connected layers. In this paper, the top global pooling layer is focused on, which is used in place of the fully connected layer and creates a simple and effective pooling operation called random crop (RC) pooling. Additionally, how to attain regularization in the top RC pooling layer is discussed. RC pooling randomly crops the feature maps so that only the images with sufficiently scaled and centered objects can be well-trained. This approach achieves comparable accuracy on the CIFAR-10/100 and MNIST.

Y. Lee (✉) · J. Kim · M. Jung · J. Kim
Statistical Inference and Information Theory Laboratory, Department
of Electrical Engineering, KAIST, Daejeon, South Korea
e-mail: askhow@kaist.ac.kr

J. Kim
e-mail: jhkim89@kaist.ac.kr

M. Jung
e-mail: alswn0925@kaist.ac.kr

J. Kim
e-mail: junmo.kim@kaist.ac.kr

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_25

1 Introduction

In recent years, convolutional neural networks (CNN) [10] have achieved state-of-the-art results for many computer vision tasks. One reason for this is that deep learning uses large datasets, such as ImageNet [4] and Microsoft COCO [14], which are recent developments. For this reason, despite increasing numbers of the parameters in deep architecture, networks are able to prevent overfitting. To accommodate this, several regularization methods have been developed, with many including dropout, dropconnect, ℓ_1/ℓ_2 regularization, and multitask learning. These methods only deal with layers containing learnable parameters, such as the convolutional layer and fully connected layer. Stochastic pooling [20] draws selected activation from a stochastic process and suggests regularization for pooling. To the best of our knowledge, this is the first work to consider spatial location selection as the output of the region in feature maps. Replacement of the fully connected layer with global average pooling is a compelling issue in recent CNN architecture. Because increasing nonlinearity in fully connected layers can be replaced by stacking convolutional layers, global average pooling gives the same linear transformations without learnable parameters. Following up the recent trend to use global pooling operations, an efficient global pooling operation named random crop (RC) pooling is proposed in this research. The contributions of this paper are as follows:

1. Regularization method using global average pooling
   The main limitation of replacing a fully connected layer with global average pooling is that the use of a fertile regularization method (e.g., dropout) can be restricting. A simple and new method for regularizing the global average pooling layer is proposed.
2. High precision classifier
   The proposed method enforces the deep neural network with global average pooling to focus only on the restricted images with sufficiently scaled and centered objects. The examples images are focused using RC pooling, as shown in Fig. 1. Therefore, deep neural networks focus only on those images in the training phase that have a more accurately defined decision boundary.

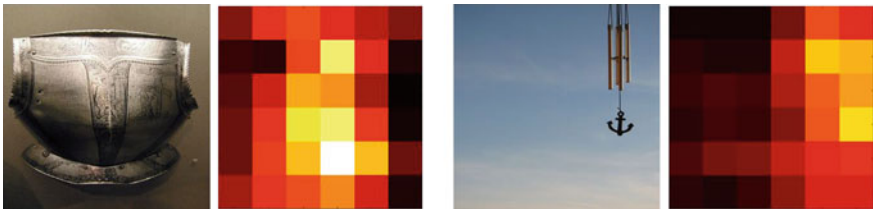


Fig. 1 Feature maps in the mid-level and high-level layers of CNNs use information from the inferior locations of original images. *Left* a good example of an image containing a large *centered* object and its corresponding feature map at the *top* of the network before global average pooling; *Right* a bad example of an image containing a small side-positioned object and its corresponding feature map

RC pooling not only improves regularization but also lets the deep networks know whether an object in a training image is located in the center of the input and is sufficiently scaled. In previous work, the deep networks trained using entire images or poorly cropped images can exhibit a particular bias in their backgrounds, because of the inaccurate object location in the input images. This often causes significant generalization errors when recognizing unseen images. Therefore, deep networks were developed in this study to recognize training images with centered objects and to generalize the top global pooling layer.

2 Related Work

A classical CNN is composed of two alternating convolutional and pooling layers, following a global average pooling or a fully connected layer. A pooling layer, called a subsampling layer, aggregates the features extracted and passed through the convolutional layers and nonlinear activation functions. For pooling applications, the pooling layer plays two main roles. First, it increases the robustness of small local translations and reduces the size of feature maps to alleviate computational complexity. Additionally, it provides summarized statistics of the outputs of previous feature extraction layers. In the final stage of deep networks, global pooling provides compact statistics of input from the entire network without any learnable parameters. The formal notation derived by ℓ_p pooling [2] is summarized as follows:

$$x^{\ell+1} = \text{pool}(\mathbf{x}^{\ell}) = (|x_1|^p + \cdots + |x_i|^p)^{1/p}, \forall i \in R, \tag{1}$$

where x is the element of a pooling region and R is the pooling region in a feature map. After the nonlinear activation function, which has a positive output, ℓ_p pooling becomes average pooling or max pooling. When $p = 1$, ℓ_p pooling corresponds to average pooling, and $p = \infty$ corresponds to max pooling. At the top of the neural networks, global average pooling or global max pooling are selected occasionally, instead of fully connected layers. In Network In Network (NIN) [13] and GoogleNet [17], global average pooling is used in place of a fully connected layer, and in [12], a global max pooling substitutes the fully connected layer.

Generalized ℓ_p pooling [2], max pooling, and average pooling are deterministic, and a pooling rule is predetermined by prior knowledge. In stochastic pooling, the pooling rule to determine which activation values are going to feed forward is drawn from a multinomial distribution formed by activation in pooling regions. Even though pooling removes important spatial information, it is widely used in many deep architectures to reduce the complexity.

In [15], to achieve considerably high recall using few region proposals, an additional label, which denotes whether the input patch contains an object roughly centered in the input and whether the object is fully contained in the patch, is included. However, compared to their work, the network with RC pooling can recognize these constraints by itself, instead of using labels.

3 Random Crop Pooling

3.1 Global Average Pooling Versus Fully Connected Layers

Recent deep architecture, such as VGG [16], Alexnet [9], showed an outstanding performance in a variety of computer vision tasks. These models incorporate the fully connected layers on top of the network prior to applying a softmax classifier. In contrast, ILSVRC 2014 classification winner GoogleNet [17] and NIN [13] achieved significant gains in CIFAR-10/100 and MNIST, by determining the overfitting problem in the fully connected layers and recommending the use of global average pooling to prevent overfitting and easily adapt a network according to the size of images. Even though global average pooling does not have learnable parameters, both global average pooling and the last linear fully connected layer have the same functional properties. They both perform linear transformations of the vectorized feature maps.

The difference only lies in whether the transformation matrix is prefixed or learned. In practice, the fully connected layers have drawbacks compared to a global pooling layer. The fully-connected layers increase model complexity and overfit the deep networks. In contrast, global average pooling, which has no additional parameters, can avoid this problem. Therefore, the use of a global average pooling layer or global max pooling layer, instead of a fully connected layer, can act as a regularizer. Table 1, which refers to NIN [13], shows the improved results from the CIFAR-10 dataset when global average pooling is used in place of fully connected layers, even though a dropout is used in the fully connected layer.

3.2 RC Pooling

Recent pooling operations, such as max pooling, average pooling, stochastic pooling [20], and generalized ℓ_p pooling [2], have achieved a significant progress in reducing computational complexity and in translating invariances. Additionally, stochastic pooling enables pooling to contribute to the regularization of deep networks. Previous pooling operations did not deal with spatial information of the elements in a pooling region. The proposed RC pooling can reduce dimensions, similar to the way the previous pooling layers have, and also reorganizes, significant and efficient data to contain the spatial information. The formulation for RC pooling is as follows:

Table 1 Global average pooling used in place of fully connected layers

Method	Test error (%)
mlpconv + fully connected	11.59
mlpconv + fully connected + dropout	10.88
mlpconv + global average pooling	10.41

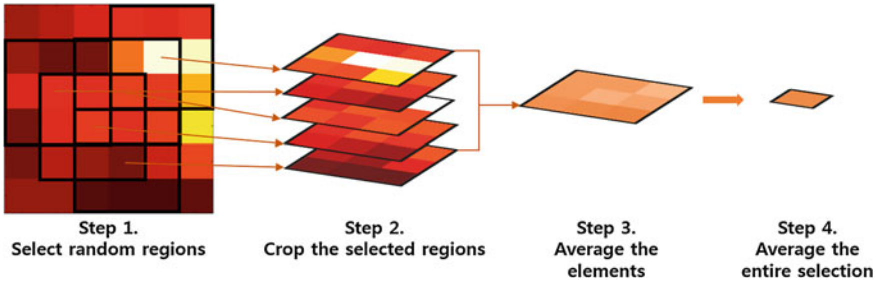
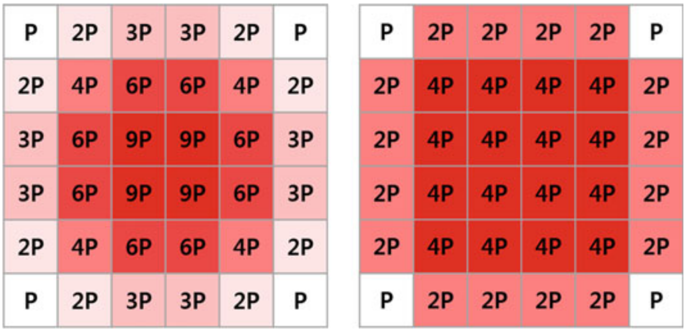


Fig. 2 RC pooling operation. When the crop size is 3 × 3, and the number of the crop is 5, RC pooling randomly extracts the local patches and then averages all of the extracted patches at each pixel location. *Left* the final feature map before RC pooling; *middle* the selected and cropped regions of the feature map; *right* the average of the cropped regions of the feature map

$$x^{\ell+1} = \text{RC-pool}(\mathbf{x}^{\ell}) = \frac{1}{\alpha}\sum_{j=1}^{\alpha}\frac{1}{\beta^2}\sum_{i=0}^{\beta^2} x^{\ell}_{i,j}$$

Here, the number of cropped regions is α, and $\beta \times \beta$ is the size of cropped regions. Only cases in which the height and the width of cropped maps are equal were considered. A formal description of RC pooling is composed of several random cropping processes in the final feature maps and to average the cropped values. First, the spatial locations x_{min} and y_{min} in the N×N feature map is selected α times. Then the regions are cropped to contain elements with a size of $\beta \times \beta$, from x_{min} to $x_{min} + \beta - 1$ and from y_{min} to $y_{min} + \beta - 1$. The algorithm for a final feature map is described in Fig. 2. At test time, aggregating randomly cropped feature map scores was not considered for computational efficiency. Following the classical global average pooling process, the entire feature map was averaged.

Because RC pooling selects crop locations randomly, the probability that these locations are selected as output in a pooling region is different for each location of a feature map. The side elements in a pooling region have a lower probability for selected than the center elements. For example, the farthest top-left element is selected only once when the cropping region is located in the farthest top-left region. However, other elements, which are not located on the sides of images can be selected more than once, because the number of crop regions containing those elements is more than one. Figure 3 shows different probabilities for different locations in the feature map. Well trained networks show an approximated and abbreviated object location in low resolution in the final feature maps before a global pooling layer is applied. Therefore, at training time, RC pooling provides a higher weight for the images in which an object is centered in the patch and fully contained in that patch. Conversely, RC pooling provides lower weights for the images with labels that may have been wrong. Figure 1 shows which images are evaluated as having high or low weights in RC pooling operations.



P	2P	3P	3P	2P	P
2P	4P	6P	6P	4P	2P
3P	6P	9P	9P	6P	3P
3P	6P	9P	9P	6P	3P
2P	4P	6P	6P	4P	2P
P	2P	3P	3P	2P	P

P	2P	2P	2P	2P	P
2P	4P	4P	4P	4P	2P
2P	4P	4P	4P	4P	2P
2P	4P	4P	4P	4P	2P
2P	4P	4P	4P	4P	2P
P	2P	2P	2P	2P	P

Fig. 3 In a feature map, the probability for selection varies according to the location. *Left* $\beta = 2$; *Right* $\beta = 3$

Table 2 The effectiveness of RC pooling compared to dropout before global average pooling

	Test error (%)
Baseline	10.41
Dropout before global average pooling	10.20
RC pooling	**9.71**

3.3 Comparison of Dropout and RC Pooling

Using the dropout regularization method before global average pooling has a critical limitation in both forward propagation and backward propagation. In a forward pass, dropout changes the activation values to zero, therefore, the absolute output value of the global average pooling is reduced. Then, the deep network learns slowly because propagated gradient through global average pooling with dropout becomes smaller. In RC pooling, when the number of cropped regions is half of the final feature map size and the size of cropped regions is 1×1, the expected property does not show a loss in the learning rate. However, if the model cannot maintain the spatial information, enforced activation does not provide accuracy improvements because the final feature map contains maximally encoded information with respect to the object location. The experimental results show the effectiveness of RC pooling compared to dropout before global average pooling for CIFAR-10 dataset (Table 2).

4 Experimental Evaluation

The proposed RC pooling to replace the fully connected layer and global average/max pooling was evaluated based on three benchmark datasets: CIFAR-10/100 and MNIST. This implementation is based on the *caffe*[1] [8] library. The NIN [13] model was chosen as the baseline model because it uses global average pooling at the

[1] https://github.com/BVLC/caffe.

top of the network. For MNIST and CIFAR-10/100, the training algorithms mostly follow those in [13]. Below, the implementation details of the model for each dataset is provided, with comparisons to state-of-the-art models.

4.1 CIFAR-10/100

The CIFAR-10 dataset consisted of 60,000 32 × 32 color images in ten classes. A collection of 60,000 images is split into 50,000 training and 10,000 test images. The CIFAR-100 dataset was approximately the same as the CIFAR-10 dataset, except that it contained 100 classes instead of 10. To compare this with other state-of-the-art models, the same network architecture used in [13] was adapted. The entire model was expressed as a nine layer convolutional layer and the additional pooling layers, Input-C(5, 192)-C(1, 160)-C(1, 96)-MP(3, 2)-C(5, 192)-C(1, 192)-C(1, 192)-MP(3, 2)-C(3, 192)-C(1, 192)-C(1, 10)-AP. Here, C(k, n) denotes the k × k convolutional layer with n output channels and a stride of 1. MP(k, s) denotes a k × k max pooling layer with a stride of 2, and AP denotes the average pooling layer. A max pooling layer followed after the first and the second convolutional layer, and a global average pooling layer was used before a softmax classifier. The model also used dropout regularization with a rate of 0.5 after the first and second max pooling layers. a rectified linear unit (ReLU) [9] was used for the nonlinear activation function. The images were only preprocessed using global contrast normalization without ZCA whitening, similar to [7, 13, 20]. Evaluations were completed using data augmentation. Each image was 4 pixels with zero padding on each side, cropped randomly, and flipped horizontally at training time without any advanced data augmentation techniques, such as scale and color jittering. At test time, no model averaging was performed. The NIN model with RC pooling achieved a 0.7 % improvement without data augmentation and a 0.58 % improvement with data augmentation in CIFAR-10. Additionally, RC pooling improved 1.58 % compared to NIN baseline results with CIFAR-100. RC pooling with a baseline model surpassed the classical global average pooling using the same baseline model and only modified the top layer of the network with the same architecture. In CIFAR-10, the final feature map size was 8 × 8. The number of cropped regions used was 5 and the crop size was 4 × 4. In CIFAR-100, other hyperparameters used were the same as in CIFAR-10, but instead the crop size was 3 × 3. Table 3 shows that RC pooling generalizes well using this model compared to previous results.

4.2 MNIST

The MNIST handwritten digit classification task consisted of 70,000 28 × 28 grayscale images in 10 digit classes. The digits contained a training set of 60,000 images and 10,000 testing images. For fair and explicit comparisons, the baseline model was

Table 3 Test error rates for CIFAR-10 and CIFAR-100 datasets

Method	Error (%)	
	CIFAR-10	CIFAR-100
No data augmentation		
Stochastic pooling [20]	15.13	42.51
Maxout networks [7]	11.68	38.57
Deeply supervised nets [11]	9.78	34.57
NIN + APL units [1]	9.59	34.40
NIN + Global average pooling [13]	10.41	35.68
NIN + RC pooling	9.71	34.52
Method	CIFAR-10	
Data augmentation		
Maxout networks [7]	9.38	
Dropconnect [19]	9.32	
Deeply supervised nets [11]	8.22	
NIN + APL units [1]	7.51	
NIN + Global average pooling [13]	8.81	
NIN + RC pooling	8.23	

set up based on the NIN architecture used in [13], which was used in the previous CIFAR evaluation. The weights were initialized by "*Xavier*" [6] initialization, which takes the variance of the weight as a proportion of the product of the inverse of filter size and the number of input channels. The number of filter channels was set to 96, 128, and 128 in three convolutional layers, each following two sequential 1×1 convolutional layers. Instead of the fully connected layer, global average pooling was used, which is similar to the NIN process in [13]. Using the same model architecture, a comparable result with an error rate of 0.38% was achieved, when the crop size was 3×3 and the number of cropped region is 5. Throughout the proposed RC pooling there were no modifications of the objective function or the model architecture except in the top pooling layer, and the accuracy surpassed the previous results. Table 4 summarizes the results and compares them to other state-of-the-art methods.

Table 4 Test error rates for the MNIST dataset (without data augmentation)

Method	Error (%)
Stochastic pooling [20]	0.47
Maxout networks [7]	0.45
Deeply supervised nets [11]	0.39
NIN + Global average pooling [13]	0.47
NIN + RC pooling	0.38

5 Conclusion

Inspired by the random crop method, which is widely used for data augmentation, RC pooling was proposed. The results reveal that it is possible to collect and train images that have well-positioned objects without looking backward when images were fed forward into a deep neural network. In addition, a new pooling method to preserve and analyze spatial locations of features was proposed. To the best of our knowledge, This is the first study in this line of research that deals with both regularization and spatial information in the pooling domain. In Imagenet classification, top RC pooling may not improve accuracy, because the Imagenet classification dataset contains many images with off-center and small objects. Nonetheless, RC pooling can be used for detection tasks, combined with R-CNN [5] and classification+localization tasks in ILSVRC. The CNN with RC pooling can provide more reliable scores for the chosen region. Therefore, if the images are first selected using other region proposal methods, such as selective search [18], edge boxes [21], and bing [3], RC pooling is a more powerful replacement to ensure that the classifier has a tight bound for precision.

Acknowledgments This research was supported by the MOTIE (The Ministry of Trade, Industry and Energy), Korea, under the Technology Innovation Program supervised by KEIT (Korea Evaluation Institute of Industrial Technology), 10045252, Development of robot task intelligence technology.

References

1. Agostinelli, F., Hoffman, M., Sadowski, P., Baldi, P.: Learning activation functions to improve deep neural networks (2014). arXiv:1412.6830
2. Bruna, J., Szlam, A., LeCun, Y.: Signal recovery from pooling representations (2013). arXiv:1311.4025
3. Cheng, M.M., Zhang, Z., Lin, W.Y., Torr, P.: Bing: binarized normed gradients for objectness estimation at 300fps. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3286–3293. IEEE (2014)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 248–255. IEEE (2009)
5. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580–587. IEEE (2014)
6. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)
7. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks (2013). arXiv:1302.4389
8. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, pp. 675–678. ACM (2014)

9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
10. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural comput. **1**(4), 541–551 (1989)
11. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets (2014). arXiv:1409.5185
12. Liang, M., Hu, X.: Recurrent convolutional neural network for object recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3367–3375 (2015)
13. Lin, M., Chen, Q., Yan, S.: Network in network (2013). arXiv:1312.4400
14. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: common objects in context. In: Computer Vision—ECCV 2014, pp. 740–755. Springer (2014)
15. Pinheiro, P.O., Collobert, R., Dollar, P.: Learning to segment object candidates (2015). arXiv:1506.06204
16. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv:1409.1556
17. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions (2014). arXiv:1409.4842
18. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. Int. J. Comput. Vis. **104**(2), 154–171 (2013)
19. Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using dropconnect. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 1058–1066 (2013)
20. Zeiler, M.D., Fergus, R.: Stochastic pooling for regularization of deep convolutional neural networks (2013). arXiv:1301.3557
21. Zitnick, C.L., Dollár, P.: Edge boxes: locating object proposals from edges. In: Computer Vision–ECCV 2014, pp. 391–405. Springer (2014)

Learning with Learning Robots: A Weight-Lifting Project

Igor Verner, Dan Cuperman, Anjali Krishnamachar and Sherri Green

Abstract Robotics projects of novice engineering students commonly focus on modeling predetermined reactive behaviors. This paper proposes an alternative approach, namely engaging students in creation of and experimentation with learning robots. To verify the feasibility of such approach, we conducted a case study, in which two novice engineering students constructed a humanoid robot and implemented a robot learning experiment. The project assignment was to build a humanoid robot capable to learn to adapt its posture while lifting various weights. The students successfully performed the project. Their robot learned from successes and failures of its trials while referring to analytical analysis made by a remote computer. Our case study showed that practice in teaching a robot to learn had significant advantages: It introduced the students to advanced concepts of robotics and AI, taught to perform engineering experiments by combining empirical and analytical methods, and inspired thinking about learning and meaning-making.

1 Introduction

There is an open debate on different strategies of educational robotics [1]. It has become a common practice to engage novice engineering students in projects focused on learning-by-doing activities of robot construction [2]. Constructivist

I. Verner (✉) · D. Cuperman
Technion Israel Institute of Technology, 3200003 Technion City, Haifa, Israel
e-mail: ttrigor@technion.ac.il

D. Cuperman
e-mail: dancup@technion.ac.il

A. Krishnamachar · S. Green
Massachusetts Institute of Technology, 77 Massachusetts Avenue,
Cambridge, MA 02139, USA
e-mail: anjalik@mit.edu

S. Green
e-mail: sagreen@mit.edu

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_26

educators point to the need to deepen learning-by-doing toward learning-by-thinking activities of selecting, organizing, and integrating knowledge [3]. Educators and researchers call to implement alternative approaches that will make robotics an attractive and meaningful subject for different learners [4].

In this paper, we propose and explore an approach in which the challenge of implementation of robot learning is used to inspire students' thinking. Accordingly, the student is assigned to implement a robot task in which the desired robot behavior cannot be preprogrammed, but has to evolve during its autonomous operation. In such a project, the student learns how to teach the robot to learn. This brings the student closer to the frontier of robotics and AI research, as development of learning capabilities of autonomous robots is one of the challenges faced by robot intelligence technology.

Robot competitions are an effective way to promote research in this area by posing standard benchmark problems in a real-world context [5]. One example of such a competition is the FIRA HuroCup, which poses a number of challenging tasks to be performed by humanoid robots. The goal of combining different tasks in the HuroCup is to avoid exploiting narrow specialized hardware solutions and encourage development of more sophisticated AI-based solutions [6].

One of the main problems in the HuroCup is keeping the humanoid robot stable throughout the tasks. This problem becomes challenging when the robot has to lift and carry unknown weights. Weight lifting is a benchmark problem of humanoid robotics that has been studied in general and in the HuroCup context, based on an analytical approach [7]. The analytical solutions cannot take into account all factors influencing the robot stability in the real world. They also tend to be mathematically complicated and difficult for understanding for novice students.

The project performed by our students addresses this problem by taking the empirical approach based on robot learning through self-directed experiments and basic analytical analysis. The humanoid robot implements a robot learning procedure to experientially determine a right posture for lifting a given weight.

This project is the first step of our ongoing educational study aimed to develop a strategy for engaging novice engineering students in an experiential inquiry into robot learning. The questions we asked at this step were as follows:

1. Can the chosen Bioloid kit be used to support student experimentation in robot learning?
2. What skills can students learn through such experimentation?
3. Can a project that focuses on robot learning foster students' awareness and understanding of learning?

The project was conducted at the Technion Center for Robotics and Digital Technology Education (http://edu-tech.technion.ac.il/). The participants were two female students, a freshman and a sophomore, majoring in mechanical engineering at MIT. They came to the Technion in the framework of the MIT international student internship program MISTI (http://misti.mit.edu/student-programs/internships).

In the following sections, we will present the project and the conclusions that we made with regard to the above questions.

2 Pedagogy of Robot Learning

The starting point in our project was the discussion of how people learn [8]. We considered the constructivist approach to education and how it can be interpreted in the context of robot learning. The following features of learning were emphasized:

- Learners are not passive recipients of information. They construct their knowledge through activities and interaction with the environment.
- Learning includes receiving, retaining, retrieving, and processing information.
- Knowledge is constructed in the form of internal (mental) representations of information from the environment with inferences about its meaning.
- Learners update the internal representations in light of new experiences.
- Learning leads to and is expressed in changes in learner's behavior.
- Learning is determined by the activity, its context, and outcomes.
- Learning is shaped through social interaction and discourse.
- While the learner is responsible for acquisition of knowledge, the teacher's responsibility is for creating a safe environment for learning.

One more educational concept that was discussed with the students was the concept of experiential learning. According to Kolb [9], experiential learning is the process in which knowledge is developed through four stages: concrete experience, reflective observation, abstract conceptualization, and active experimentation. The learner experiences a situation and then reflects on how he can use this experience to gain knowledge and a greater understanding of the world. This reflection motivates the learner to conceptualize a new idea or to change his previous understanding of a concept. The learner then applies these abstract concepts to the world around him.

In this project, we guided the students to apply the discussed educational concepts to the design of learning behaviors of the robot. For robots, experiential learning is necessary to adapt to unfamiliar situations and to react to changes in known situations and their environment [10]. In robot learning, machine learning is combined with robotics in an effort to impart upon robots cognitive and decision-making skills similar to those of humans. While there are many different ways to implement robot learning, our study focuses primarily on adaptive learning through experience. This involves the robot using trials to gather data to better understand how it should interact with its environment.

Specifically, we explore the task of weight lifting in which the robot is performing a series of trials to lift unknown weights. This task is especially topical for humanoid robots, as it severely undermines the stability of the robot that depends on the complex dynamics of the entire robot's body [11].

3 Robot Development

The humanoid robots in this project were constructed using the ROBOTIS Bioloid Premium kit (http://en.robotis.com/index/product.php?cate_code=121010). From our experience [12], the kit is a suitable platform for rapid prototyping of humanoid robots. This practice opens opportunities for students, even novices, to explore advanced concepts in robotics that have previously only been available to professionals.

3.1 Construction of the Robot

Following the instructions in the kit manual, the students constructed a humanoid robot with 18 servomotors, an IR sensor, an accelerometer, and a sound sensor (Fig. 1a). The code provided by the company included some basic preprogrammed movements, such as walking and standing up. Additional motions were programmed using the RoboPlus software. To enable communication between the robot and the remote computer, the students used the ROBOTIS BT-210 Bluetooth module. They developed the necessary software for the remote computer as a Python script.

3.2 Development of the Virtual Robot

A virtual model of the robot was developed using PTC Creo Parametric CAD modeling software (http://www.ptc.com/cad/3d-cad/creo-parametric). We imported

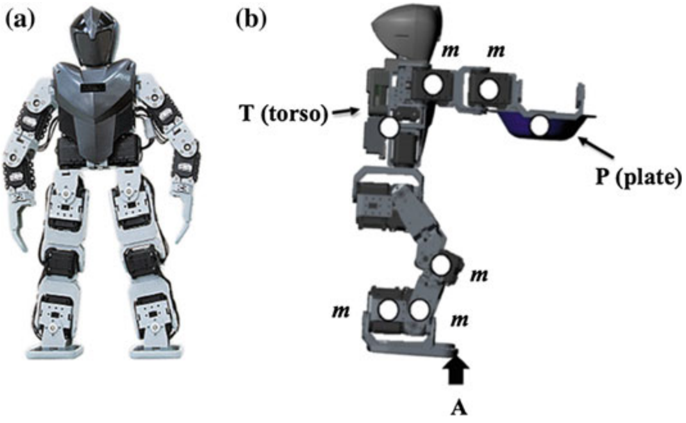


Fig. 1 **a** The humanoid robot and **b** the virtual model of the robot

into Creo the CAD models of the Bioloid parts available on the ROBOTIS Web site and assembled the virtual robot in the same way as the physical one. The completed virtual robot's joints were assigned the same range of motion as the physical robot. The virtual robot can thus serve as a useful instructional tool in learning to manipulate the physical robot. The Creo-based virtual model, used for the analytical analysis of robot stability, is presented in Fig. 1b.

3.3 Stability Analysis

A simplified model of the robot was used to analyze the stability of the robot holding different weights. In this model, the robot is represented by six blocks, each containing several motors and regarded as a point mass. The horizontal distances between the center of mass of each block and the edge of the robot's foot were measured in the position shown in Fig. 1b. The moment about point A was calculated as a function of the tilt angle of the robot. This simple analytical analysis allowed us to estimate the critical angle at which the robot would fall over for any given weight.

The critical angle α can be found from the equilibrium condition about point A:

$$M_A = 0 = T(L\ \sin(\alpha) + x_i) + 2m(d_1 + d_2 + d_3 + d_4 + d_5 + 2L\ \sin(\alpha)) + P(d_6 + L\ \sin(\alpha)) \tag{1}$$

$$\alpha = \sin^{-1}\left(\frac{-\left(Tx_i + 2m(d_1 + d_2 + d_3 + d_4 + d_5) + Pd_6\right)}{L(T + P + 4m)}\right) \tag{2}$$

where m is the mass of each motor, T is the mass of robot's torso (g), P is the mass of plate with added weight (g), L is the distance from hip rotation to CM of torso (m), and d_1–d_6 are distances from edge of robot's foot to the 6 blocks (m). The calculated angle value α is used in the analytical solution.

4 The Experiment

In the experiment, the robot learns to select the best posture to lift a given weight. During this learning process, two modes of operation are applied. In the data gathering mode, the robot experiments with lifting unknown weights, while each experiment is logged in the remote computer. In the advising mode, the robot tries to lift a measured weight by setting its standing posture following an advice from the remote computer.

4.1 Data Gathering Mode

In this mode, the robot tests its ability to hold a measured weight at various postures of body tilt angle. The robot evaluates its success or failure and sends the test results to a remote computer database. In each discrete experiment, the robot is given an unknown weight, while sitting down. As the hardware does not provide means to measure the weight directly, the measurement is carried out indirectly. We found that when the motors are idle and loaded by an external torque, the angular velocity of the motor axis is roughly proportional to this torque. Based on this finding, we act in the following way. The robot first records the original angle of the two motors in its shoulder joints and then makes those motors idle for a specific time interval (half a second). It then measures the new angles of the motors and calculates the resulting angle displacements. Using the mentioned proportionality, the robot calculates the amount of weight that it holds in its hands. After that, the robot performs weight lifting trials for body tilt angles ranging from 10° to 40°, each time attempting to stand up from the sitting position. Success or failure of each trial is self-evaluated by the robot. Here, success means that the robot succeeded to stand up holding the weight, while failure means that the robot falls over. The robot recognizes falling using an abrupt change in the values in its accelerometer.

A remote computer interacts with the robot and performs data collection, processing, and analysis. Following repeated trials, data for varying angles and weights are sent via Bluetooth to the computer and populate a database. The data sent after each trial include the weight lifted, the tilt angle taken, and the status of success or failure in performing the task. The trials are plotted on a graph, which has weight on the horizontal axis and tilt angle on the vertical axis. In such graph, '×' marks represent failure and dots represent success, as shown in Fig. 2. For example, the dot at 174 g and 24° represents a trial where the robot succeeded to lift a weight of

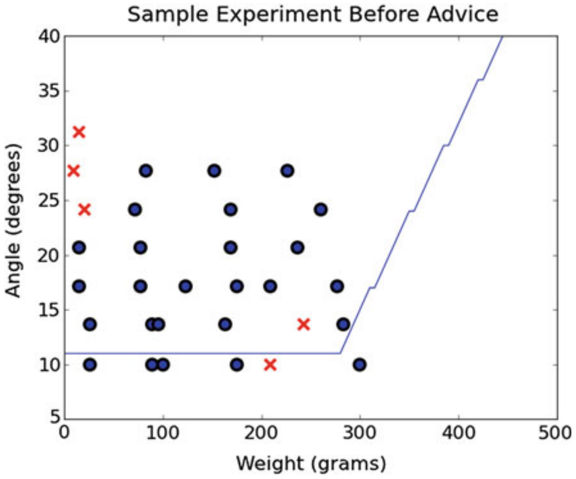


Fig. 2 Graph of a sample gathering data experiment

174 g at a tilt angle of 24°. On the other hand, the '×' at 210 g and 10° represents a failed attempt to hold 210 g at a tilt angle of 10°.

The graph is automatically updated each time the robot completes a trial. On the same axes, the script plots a function that describes the relationship between the critical angle α and the weight.

4.2 Advising Mode

In this mode, the robot just measures a weight and sends the value to the remote computer. The computer uses the empirical data of robot trials, accumulated in the database, together with the analytical solution, and sends a recommended tilt angle to the robot. The robot uses that tilt angle in a trial and determines its success or failure as described above. It sends the result of the trial to the computer, to be added to the database. Figure 2 presents sample data collected from one experiment involving 31 trials and the calculated angle weight dependency. One can see the discrepancies between the analytical and the actual successes and failures. Some of the experimental results presented in Fig. 2 appear to be duplicate measurements of roughly the same weight, for instance at an angle of 10° and a weight of 90 and 100 g. This occurs because of inaccuracies in weight measurement done by the robot.

The data gathered in Fig. 2 were used to advise the robot, and the results of 10 trials in advising mode are shown in Fig. 3.

The advising algorithm had an 80 % success rate, and we can see from the graph that the advice improved as the robot provided feedback on the previous advice. For example, we see the '×' at a weight of 334 g and an angle of 17°. After this failed trial, the robot was given the same weight again. It measured the weight as slightly

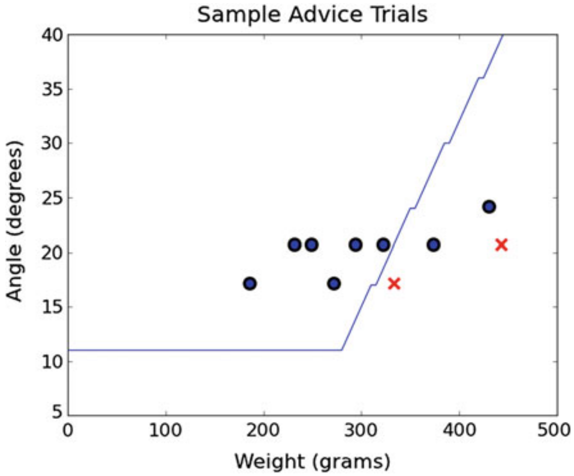


Fig. 3 Graph of a sample advising mode experiment

heavier (374 g), but the robot successfully held the weight because it increased the angle to 20°, as we can see with the dot at that point. The advising algorithm was 100 % successful for weights within the range for which the robot already had gathered data (0–300 g). The only failures occurred for weights that were beyond this range (334 and 443 g).

5 Discussion and Conclusion

Back to the first of the three questions posed in Sect. 1, the study shows that the humanoid robot constructed from the ROBOTIS Bioloid Premium kit can be used for experimentation in robot learning. To implement the robot learning project described in the previous sections, the basic platform of the Bioloid kit can be enhanced by the addition of the ROBOTIS BT-210 Bluetooth module and a remote computer used for data storage, analysis, and communication. We found Python to be an effective programming tool for this remote computer. The robot adapts its posture through the experiential learning cycle. At the concrete experience step, it gains experience by gathering experimental data through trials of lifting different weights. At the reflective observation step, the robot records and communicates to the computer information about successes and failures of the trials. At the conceptualization step, it receives from the computer results of data analysis, based on the comparison between the experimental results and the analytical calculation. Lastly, at the active experimentation step, the robot takes what it has learned from its past experiences and accordingly adapts its behavior.

To summarize, laboratory experiments confirmed that the robot is capable of improving its posture to successfully perform the weight-lifting task.

Regarding the second question, our follow-up showed that the project strongly engaged both students and exposed them to different areas of modern engineering that were new for them and are traditionally introduced in advanced studies. The students learned and applied concepts in robot control, remote communication, CAD-based analysis, human–robot interaction, and others. They acquired skills of engineering experimentation in determining basic mechanical properties such as center of mass, torque, and balance. They performed analytical calculations, compared results to the actual experimental results, and discussed discrepancies. The students developed skills of computer and robot programming. They got practical experience in presenting the main ideas of the project to robotics researchers, professional engineers, and school students.

To the third question, the study gave indications of an increase in students' awareness and understanding of learning following their participation in the project. The students implemented the discussed features of learning in the robot behavior and designed the experiment as an experiential learning cycle. The students presented their project to a group of 34 American high school pupils participating in the Technion international summer outreach program. They also developed an instructional unit on the subject.

The presented educational experiment is the first step in developing a strategy for engaging novice engineering students in an experiential inquiry into robot learning. Following its positive results, the authors plan to develop a curriculum to expose advanced high school students to robot learning and basic engineering concepts.

The robot learning experiment can be enhanced by using an Internet-of-Things platform to facilitate Web-based communication between multiple robots and a shared database. Further stability analysis can be performed with the virtual robot and CAD software.

Acknowledgements This project was supported by the MIT MISTI Israel program and the PTC grant. The authors thank PTC employees Dr. Michael Reitman, Eldad Finkelstein, Galya Lin, and Amir Merksamer for technical consultation.

References

1. Alimisis, D.: Educational robotics: open questions and new challenges. Themes Sci. Technol. Educ. **6**, 63–71 (2013)
2. Eguchi, A.: Educational robotics as a learning tool for promoting rich environments for active learning (REALs). Handbook of Research on Educational Technology Integration and Active Learning. IGI Global, PA, pp. 19–47 (2015)
3. Mayer, R.E.: Should there be a three-strikes rule against pure discovery learning? Am. Psychol. **59**, 14–19 (2004)
4. Rusk, N., Resnick, M., Berg, R., Pezalla-Granlund, M.: New pathways into robotics: strategies for broadening participation. J. Sci. Educ. Technol. **17**, 59–69 (2008)
5. Anderson, J., Baltes, J., Cheng, C.T.: Robotics competitions as benchmarks for AI research. Knowl. Eng. Rev. **26**, 11–17 (2011)
6. Anderson, J., Baltes, J., Tu, K.Y.: Improving robotics competitions for real-world evaluation of AI. In: AAAI Spring Symposium on Experimental Design for Real-World Systems, Stanford, USA, pp. 1–8 (2009)
7. Kuo, C.H., Kuo, Y.C., Chen, T.S.: Process modeling and task execution of FIRA weight-lifting games with a humanoid robot. Lect. Notes Comput. Sci.: Adv. Auton. Robot. **7429**, 354–365 (2012)
8. Bransford, J.D., Brown, A.L., Cocking, R.R.: How People Learn: Brain, Mind, Experience, and School. National Academy Press, Washington, DC, USA (1999)
9. Kolb, D.: Experiential Learning: Experience as the Source of Learning and Development, pp. 21–38. Prentice-Hall, Englewood Cliffs, N.J. (1984)
10. Nehmzow, U.: Self-organisation and self-learning robot control. In: IEEE Symposium on Self-Learning Robots, London, pp. 7–8 (1996)
11. Vukobratović, M., Borovac, B.: Zero-moment point-thirty five years of its life. Int. J. Human. Rob. **1**, 157–173 (2004)
12. Verner, I., Cuperman, D., Cuperman, A., Ahlgren, D., Petkovsek, S., Burca, V.: Humanoids at the assistive robot competition RoboWaiter 2012. In: Kim, J.H. et al. (eds) Robot Intelligence Technology and Applications. Advances in Intelligent Systems and Computing, vol. 208, pp. 763–774 (2013)

Ensemble of Vector and Binary Descriptor for Loop Closure Detection

Mohammed Omar Salameh, Azizi Abdullah and Shahnorbanun Sahran

Abstract Loop closure detection plays an important role in vSLAM for building and updating maps of the surrounding environment. An efficient vSLAM system needs an informative descriptor for landmark description and stable model for making decisions. Most of the solutions dependent on using a single descriptor for landmark description, whereas other solutions proposed to use a combination of descriptors. However, these solutions still have the limitation in correctly detecting a previously visited landmark. In this paper, an ensemble of loop closure detection is proposed using Bayesian filter models for making decisions. In this approach, a set of different keypoint descriptors is used as input to bag-of-word descriptors. After that, these descriptors, i.e., SIFT, SURF, and ORB, are used to construct Bayesian filter models and ensemble learning algorithm for loop closure detection. The proposed approach is validated on a public dataset, namely City-Center dataset (CiC). The results shown that the proposed ensemble algorithm outperforms single model and existing loop closure detection system approaches. It gives 87.96 % for ensemble learning and 86.36 % for the best single model and 37, 80, 81 % for FAB-MAP, PIRF-Nav2.0, and RTAB-MAP, respectively.

Keywords Appearance-based localization · Loop closure detection · Ensemble learning

M.O. Salameh (✉) · A. Abdullah · S. Sahran
Pattern Recognition Research Group, Center for Artificial Intelligence Technology,
Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia,
43600 Bangi, Malaysia
e-mail: m.omar82@siswa.ukm.edu.my

A. Abdullah
e-mail: azizia@ukm.edu.my

S. Sahran
e-mail: shahnorbanun@ukm.edu.my

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_27

1 Introduction

Robots are designed to serve people in their daily tasks in large areas of long-term operations. A robot is required to construct a map to identify the surrounding environment and to localize its position within this map. The map building and localization processes are interrelated and it is known as Simultaneous Localization and Mapping (SLAM) [7, 12, 19]. Vision sensors, such as cameras, are commonly used in SLAM. Cameras provide rich information about the environment, are low in cost, light and compact, and consume less power. SLAM that comes with a visual sensor is known as Visual Simultaneous Localization and Mapping (vSLAM). One type of map that is used with vSLAM is the topological map, which has the structure of a graph, with the nodes containing visual landmarks for locations explored, and the edges representing the relationship between neighboring nodes (i.e., neighbors by location or by time) [7, 12, 19]. Loop closure detection is a process that runs online through the SLAM session to correct and update the maps (i.e., by the robot trajectory, subject to drift and error) by identifying the new locations and reducing the duplication in the maps. Loop closure detection ideally runs for each new image against all locations in the map, where the map sizes are increasing continuously. It will be challenging to run the loop closure in real time [2, 7, 15, 16, 22]. One of the popular approaches to loop closure detection is Fast Appearance-based Mapping (FAB-MAP) [6]. The proposed FAB-MAP uses a single keypoint descriptor, namely Scale-Invariant Feature Transform (SIFT), and also an offline bag-of-word descriptor (BoW) for landmark description and a Bayes filter for predicting the loop closure candidates. Another loop closure detection system, namely Real-time Appearance-based Mapping (RTAB-Map) [16], is proposed. This system uses the single keypoint descriptor Speeded-Up Robust Features (SURF) and an online BoW for landmark description. RTAB-Map splits the memory into two parts: (a) One part has a fixed size that can keep the operation of the loop closure detection running under real-time constraints. This part of the memory contains the new locations and the most frequently visited ones. (b) The other part of the memory can be increased and contains the rest of the locations. A solution by a single descriptor leads to lack of discriminating information, which then affects the ability to recognize previously visited locations.

Visual feature combination approaches have been proposed to recover the lack of single descriptor landmarks. A combination of global features with local features was proposed by [10, 11], while a hierarchical descriptor and multilayer descriptors were proposed by [18, 20, 21]. However, these solutions still have the same error rate as a single-detection system and face the problem of overfitting due to the combination of multiple features to construct a dance feature vector.

This paper presents an ensemble of loop closure detection capable of running in real-time operations. The ensemble uses a set of Bayesian filter models for making decisions. The proposed method uses keypoint descriptors, namely SIFT, SURF, and Oriented FAST and Rotated BRIEF (ORB), for extracting visual landmarks.

After that, the BoW for each keypoint descriptor is constructed for description and Bayesian filter in making decision.

The proposed solution was tested on a public dataset (City-Centre [6]) containing 2474 outdoor images, and the result shows a significant improvement over the single descriptor loop closure detection, where the ensemble of loop closure detection outperformed the other approaches on the same dataset. It gives recall performance 87.92, 37, 80 and 81 % for FAB-MAP [6], PIRF-Nav2.0 [13] and RTAB-Map [16], respectively.

The remainder of this paper is organized as follows: Sect. 2 presents a review of the relevant methods on the appearance-based vSLAM and the loop closure detection. Section 3 describes the map building process and how a location is registered. Section 4 describes the ensemble loop closure procedure in detail. Section 5 shows the experimental setup, while Sect. 6 gives the detection results and the conclusion.

Contributions of this paper. This paper presents a vector and binary descriptor ensemble for loop closure detection. (1) It evaluates the ensemble learning for multi-Bayesian filters for loop closure detection with a discriminating location description. (2) It describes the location registration using online BoW with three different descriptors.

2 Related Works

The appearance-based loop closure detection is the heart of the appearance-based SLAM. The problem with the loop closing detection lies in the inability of the robot to distinguish between the places that might have been visited in the past (saved in the map), and the place which is being visited for the first time [2, 7, 15, 16, 22]. The general scheme of the appearance-based loop closure detection is shown in Fig. 1. Garcia-Fidalgo and Ortiz [8, 9] used the image representation method to classify loop closure detection approaches into local descriptors, global descriptors, and combination approaches.

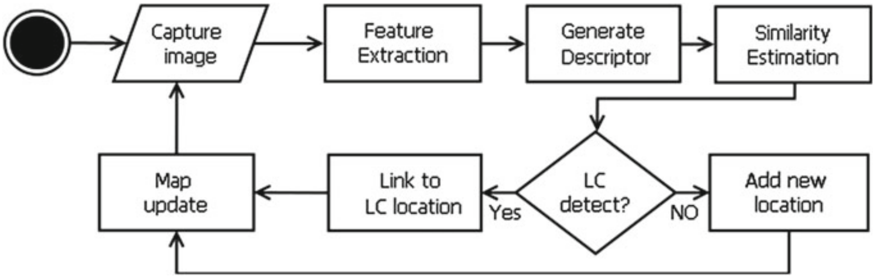


Fig. 1 Popular structure for appearance-based loop closure detection

This section presents a description of the visual descriptors (SURF, SIFT, and ORB), followed by an explanation of the online BoW construction and, finally, a brief description of the RTAB-Map as a platform for testing the proposed approach.

2.1 Image Features

Image features is a symbolic representation of the contents of an image through the detection of interest points in the given image and the extraction of features from these points. The features extraction is the first step in the loop closure detection process. The proposed approach used three descriptors (SURF, SIFT, and ORB), and a brief description of these features is given below:

Scale-Invariant Feature Transform (*SIFT*): Key points are detected as maxima over scale and space in Differences of Gaussians (DoG) convolutions. The features are represented as histograms of gradient orientations around the key points of the detected scale. SIFT has 128 dimensions.

Speeded-Up Robust Features (*SURF*): This is a keypoint detector method based on a fast Hessian detector with maximum points of the Hessian matrix. The feature orientation for each key point is found using the HAAR wavelet within a circular radius and is evaluated according to a pyramid scale-space. SURF has 64 dimensions.

Oriented FAST and Rotated BRIEF (*ORB*): This is a combination of the Features for Accelerated Segment Test (FAST) for the detection of key points and the Binary Robust Independent Elementary Features (BRIEF) for the description of the features. By applying a Harris corner, the intensity-weighted centroid of the corner point is used to find the orientation from this corner point to the centroid. ORB has 32 dimensions.

2.2 Bag-of-Word Descriptor (BoW)

The algorithm was first proposed in text retrieval, where a BoW is used to represent a document by quantizing the number of occurrences of each word, given a predefined vocabulary. If two documents have more common words, then they are classified under the same topic. This idea has been transferred to the computer vision field, where the visual features are treated as words and are quantized using the visual vocabulary known as a codebook. BoW represents an image by a set of visual words, and it is a faster method of comparison than the direct matching of visual features with high dimensions.

The codebook can be constructed offline by using a predefined dataset of images. This is a robust approach against perceptual aliasing, where it is capable of evaluating the distinctiveness of each word based on the offline codebook. However, it is necessary to construct the offline codebook for model learning and dictionary computation, where a remarkable change in the environment could have a negative impact since a good codebook for one environment may not be useful in a different environment.

The RTAB-Map uses an online incremental BoW, where each word represents a visual feature. A randomized forest of four kd-trees (with Fast Library for Approximate Nearest Neighbors (FLANN)) is used to structure the BoW. Two visual features can be represented by the same word in the codebook if they pass the condition of the nearest neighbor distance ratio (NNDR) threshold [16].

2.3 Real-Time Appearance-Based Mapping (RTAB-Map)

The RTAB-Map is a state-of-the-art appearance-based loop closure detection with vSLAM [16, 17]. This solution provides a memory management approach and was inspired by the observation that humans can strongly remember locations where they spent most of their time [3, 4]. The RTAB-Map splits the memory into two parts: The first part, known as the working memory (WM), saves the new location as well as the most visited locations. The WM has a fixed memory size, where the loop closure detection is applied only on the WM, and the running time of the detection process can be controlled by choosing the length of the WM in proportion to the real-time operation. The second part of the memory is the long-term memory (LTM), where the rest of the locations are saved. Section 3 describes how the locations are registered in the map.

The RTAB-Map uses an online incremental BoW, where each word represents a visual feature. A randomized forest of four kd-trees (with FLANN) is used to structure the BoW. Two visual features can be represented by the same word in the codebook if they pass the NNDR threshold (set 0.8) [16]. The loop closure candidates are nominated using a Bayesian filter.

2.4 Bayesian Filter

A Bayesian filter is used in loop closure detection to estimate the similarities between the current location and the locations that have been visited. A decision is then taken to register the current location as a new location or as a loop closure to a previous location.

The Bayesian filter tries to estimate the full posterior probability $p(S_t|L^t)$ Eq. (1) for the current location L_t at time t, where S_t is a set of all the loop closure candidates for the location L_t. If the locations L_t and L_i are representing the same location, where $i \in [0, \dots, n]$ and n is the number of locations in the map, then the probability of the loop closure between L_t and L_i will be $S_t = i$. On the other hand, if the location L_t is a new location, then $S_t = -1$.

$$p(S_t|L^t) = \eta \underbrace{p(L_t|S_t)}_{Observation} \underbrace{\sum_{i=-1}^{t_n} \underbrace{p(S_t|S_{t-1}=i)}_{Transition} p(S_{t-1}=i|L^{t-1})}_{Belief} \tag{1}$$

where η is a normalization term and L^t is for locations in WM (the search space of the Bayesian filter), then $L^t = L_{-1}, \dots, L_t$. *The observation* is a likelihood function, $\mathfrak{L}(S_t|L_t)$ that is used to evaluate $p(L_t|S_t)$ as follows:

$$p(L_t|S_t = j) = \mathfrak{L}(S_t = j|L_t) = \begin{cases} \frac{\mu}{\sigma} + 1, & j = -1 \\ \frac{\mathfrak{s}_j - \sigma}{\mu}, & \mathfrak{s}_j \geq \mu + \sigma \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

where $\mathfrak{s}_j$ is the location matching using Eq. (3) for L_t to all locations, $S_t = j$, where $j = 1, \dots, t_n$, and the standard deviation σ is normalized by the mean μ for the non-null $\mathfrak{s}_j$.

The second part is *the belief* part, where *the transition* model is combined with the recursive part of the filter. The transition model, $p(S_t|S_{t-1} = i)$, is measured by the probability of the transition from one state $S_{t-1} = i$ to every possible state S_t as shown in Table 1 [2].

Location Matching: The similarity measurement, $\mathfrak{s}$, is used to find the match between locations in the features level as follows:

$$\mathfrak{s}(z_a, z_b) = \begin{cases} N_{\text{pair}}/N_{z_a}, & \textit{if} \quad N_{z_a} \geq N_{z_b} \\ N_{\text{pair}}/N_{z_b}, & \textit{if} \quad N_{z_a} < N_{z_b} \end{cases} \tag{3}$$

where N_{pair} is the number of matched word pairs between the signatures z_a and z_b, $a, b \in [0, \dots, n]$, N_{z_a} and N_{z_b} are the total number of words of the signatures z_a and z_b, respectively. If $\mathfrak{s}(z_a, z_b)$ passes the similarity threshold, then the two signatures z_a and z_b are matching.

Table 1 The transition model declaration

Current location L_t	Previous location L_{t-1}	Probability $p(S_t|S_{t-1} = i)$
New location	New location	$p(S_t = -1|S_{t-1} = -1) = 0.9$
New location	Loop closure to L_j	$p(S_t = -1|S_{t-1} = j) = 0.1$
Loop closure to L_i	New location	$p(S_t = i|S_{t-1} = -1) = 0.1/\text{No.locations}$
Loop closure to L_i	Loop closure to L_j	$p(S_t = i|S_{t-1} = j)$ Probability is defined as a discretized Gaussian curve ($\sigma = 1.6$) centered on j and the value is set recursively by starting from $i = j$ to the end of the neighborhood range

3 Location Registration

SLAM is able to construct a map while estimating the pose of the robot relative to this map. The map structure and the determination of the location register depend on the type of information that is collected from the environment by the robot. vSLAM uses a vision sensor, such as a camera, to construct a topological map by collecting visual landmarks from the movement.

Topological maps are compressed, consume less memory, and are stored in efficient data structures. The environment is modeled by topological maps by means of graphs, where each node holds the appearance description of the locations as well as other relevant information, and each edge indicates a traversed path between two locations [5].

The RTAB-Map with a topological map was used as a testing platform for the proposed algorithm, with the nodes representing the visited locations and containing the location signature. The signature is a set of visual features extracted from the image belonging to this location. The node has an ID and weight, where the ID represents the location age, and the weight is a frequently visited counter. The nodes can be connected using two types of links (edges). The first link is a neighbor that is near in time or in space, while the second link is a loop closure between two nodes representing the same location.

A location has to pass through four main stages as follows to be registered in the map:

1. The first stage is the sensory memory (SM) stage, where the image is used to extract the visual features, reduce the size, and keep the useful features. A new location (node) is created with the location signature.
2. The short-term memory (STM) is a first-in–first-out stack with a fixed size. The size is set based on the velocity of the robot and rate of capture. In this stage, a comparison is made between the new location and the last location in the stack using the similarity matching in Eq. (3), and if they pass the similarity threshold, then the two locations will be merged in one node. When the short-term memory (STM) stack is full, the first location that is saved in the stack will pass on to the next stage.
3. The working memory (WM) is the active part of the memory where a location is registered as a new location in the map or as a loop closure to a privacy location. The Bayesian filter, as in Eq. (1), is applied only on the locations that have been saved in the WM. The Bayesian filter will nominate the loop closure candidates for the current location and the probability that it is a new location. If the probability value is higher than the loop closure threshold (set 0.08 [16]), the current location will be registered as a new location or the current location will be registered as a loop closure to the highest candidate location.
4. When the WM reaches the size limit, the oldest location with the minimum weight will be transferred to the LTM while the rest of the locations will be retained.

4 Ensemble of Loop Closure Detection

The ensemble learning approach has the potential to solve the weaknesses in the previously mentioned approaches, where the ensemble has the ability to reduce the risk of a single decision and can deal with the lack of information by using multiple descriptors to generate a discriminative description.

Furthermore, the ensemble can be implemented in a multi-process scheme. The proposed solution uses the ensemble learning approach in the decision level for the loop closure detection, as shown in Fig. 2.

Three independent codebooks were generated using different visual features (SURF, SIFT, and ORB), and three signatures were assigned to each location. After that, the Bayesian filter used the location signatures with the appropriate codebook to nominate a candidate loop closure for each signature using Eq. (1). A proportion of the candidates were matched to the current location and the Bayesian filter provided a probability value for each signature that it could be a new location. The ensemble rule was used to combine the proportion of all the candidates and the probabilities of the new location.

The Mean Rule: is a simple and popular rule that is used for fusion, as shown in the equation below:

$$\mu_t(S) = \frac{1}{N} \sum_{n=1}^{N} p((S^n)_t | (z^n)^t) \tag{4}$$

where $(z^n)^t$ is the signature associated for the location L_t, at time t, and $n = 1, \ldots, N$ is the number of codebooks. The proportion of matching $p((S^n)_t | (z^n)^t)$ is computed using Eq. (1).

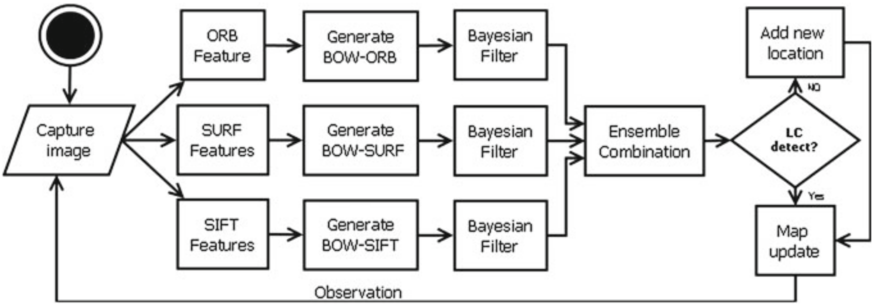


Fig. 2 Ensemble approach for loop closure detection

Fig. 3 Sample images from City-Center dataset

5 Experimental Setup

5.1 The Dataset

A public dataset was used to evaluate the capabilities and robustness of the proposed technique under various conditions.

CityCentre (CiC)[1]: The dataset contains 2474 images with a size of 640 × 480 acquired from two cameras (left and right). The images were captured outdoors along 2 Km of public roads at a rate of 0.5 Hz. The images included dynamic objects and were taken on a windy day in bright sunshine that showed the foliage and shadows in dynamic motion. For the current implementation, only one image was taken as the input; the images from the two cameras were merged into one image with a new size of 1280 × 480. The number of images totaled 1237 [6]. Figure 3 shows samples of images from the dataset.

5.2 Precision Recall

A Precision-Recall (PR) curve is a very popular evaluation approach which gives a more informative picture of the performance of an algorithm when highly skewed datasets are used (e.g., information retrieval, image retrieval, and loop closure detection) [1, 6, 16].

The main elements of PR curve are defined under the loop closure detection approaches as follows:

[1] Dataset available download on http://www.robots.ox.ac.uk/~mobile/IJRR_2008_Dataset/.

Precision (P) is defined as the number of true positive loop closure detections to the total number of detections as follows:

$$P = \frac{TP}{TP + FP} \tag{5}$$

Recall (R) is defined as the number of true positive loop closure detections to the number of ground true loop closures as follows:

$$R = \frac{TP}{TP + FN} \tag{6}$$

5.3 *Environment Parameters*

All the parameters and threshold values were set up according to the values reported in [16], where the threshold values of the NNDR, the similarity, and the loop closure were set at 0.8, 0.2, and 0.08, respectively, while the maximum size of the STM and WM were set to 30 locations and 377 locations, respectively. The source code for the RTAB-Map version 8.3 is available online [14]. The default setup was the visual features parameters, as reported in the Opencv.

6 Detection Results and Discussion

The proposed mean rule ensemble learning method with Bayesian filters can deal very well for detecting loop closure with scene sequences. This method clearly outperforms single descriptor and other loop closure systems for City-Center dataset. Table 2 shows the results on the single descriptors. Table 3 shows the result on the mean rule ensemble. From the experiment results, we observe that the performance

Table 2 Experimental Results with single descriptor on *CiC* dataset

Descriptors	Recall
SURF	86.36 ± 0.23
SIFT	85.02 ± 0.52
ORB	64.39 ± 0.26

Table 3 Experimental results with ensemble approach on *CiC* dataset

Descriptors	Recall
SURF + SIFT + ORB	87.92 ± 0.30

of the mean rule ensemble of Bayesian filter model significantly outperforms individual Bayesian filter model ($p < 0.05$). This is due to the ensemble algorithm can improve independent models most, if the model predictions are less correlated. Experiments also showed that combining strong and weak models with ensemble can retain the loop closure detection results. We believe that this is due to the ensemble can make a better trade-off between the different performances of the models. Besides, we also compared the ensemble method with other loop closure detection algorithms, namely FAB-MAP, PIRF-Nav2.0, and RTAB-Map. Our ensemble method gave the best performance of 87.92 % over other these three methods. These three methods gave 37 %, 80 %, and 81 % the performance results, respectively.

Besides the mean rule, we also did experiments on other ensemble learning methods, namely product rule and majority voting rule. However, these approaches gave a worst result of 7.6 % and 20.1 %, respectively. This is due to (a) very large probability estimation errors for all nodes of different keypoint descriptor models in the working memory of the topological map and (b) the number of identical or similar nodes for these models is not the same. These reasons might influence the final probability outputs of the ensemble learning algorithms.

In future work, we plan to focus on optimizing the important parameters in the RTAB-Map system and testing on different datasets. Besides, we are also currently studying methods for learning to weigh each independent Bayesian filter to improve the current performance result and investigating to possibly improve the product rule ensemble algorithm.

Acknowledgments The authors wish to thank to ETP-2013-053 grant for funding this project.

References

1. Abdullah, A., Veltkamp, R.C., Wiering, M.A.: Ensembles of novel visual keywords descriptors for image categorization. In: 2010 11th International Conference on Control Automation Robotics and Vision (ICARCV), pp. 1206–1211. IEEE (2010)
2. Angeli, A., Filliat, D., Doncieux, S., Meyer, J.A.: Fast and incremental method for loop-closure detection using bags of visual words. IEEE Trans. Robot. **24**(5), 1027–1037 (2008)
3. Atkinson, R.C., Shiffrin, R.M.: Human memory: a proposed system and its control processes. Psychol. Learn. Motiv. **2**, 89–195 (1968)
4. Baddeley, A.D.: Human memory: theory and practice. Psychology Press (1997)
5. Cortés, E.B.B.: Appearance-based mapping and localization using feature stability histograms for mobile robot navigation
6. Cummins, M., Newman, P.: Fab-map: probabilistic localization and mapping in the space of appearance. Int. J. Robot. Res. **27**(6), 647–665 (2008)
7. Fuentes-Pacheco, J., Ruiz-Ascencio, J., Rendón-Mancha, J.M.: Visual simultaneous localization and mapping: a survey. Artif. Intell. Rev. **43**(1), 55–81 (2015)
8. Garcia-Fidalgo, E., Ortiz, A.: State-of-the-art in vision-based topological mapping and localization methods
9. Garcia-Fidalgo, E., Ortiz, A.: Probabilistic appearance-based mapping and localization using visual features. In: Pattern Recognition and Image Analysis, pp. 277–285. Springer (2013)

10. Goedemé, T., Nuttin, M., Tuytelaars, T., Van Gool, L.: Markerless computer vision based localization using automatically generated topological maps. In: European Navigation Conference GNSS, Rotterdam (2004)
11. Goedemé, T., Nuttin, M., Tuytelaars, T., Van Gool, L.: Omnidirectional vision based topological navigation. Int. J. Comput. Vis. **74**(3), 219–236 (2007)
12. Gouda, W., Gomaa, W., Ogawa, T.: Vision based slam for humanoid robots: A survey. In: 2013 Japan-Egypt International Conference on Electronics, Communications and Computers (JEC-ECC), pp. 170–175. IEEE (2013)
13. Kawewong, A., Tongprasit, N., Hasegawa, O.: Pirf-nav 2.0: fast and online incremental appearance-based loop-closure detection in an indoor environment. Robot. Auton. Syst. **59**(10), 727–739 (2011)
14. Labbé, M.: Rtab-map version 8.3 the source code website. https://github.com/introlab/rtabmap/archive/0.8.3.tar.gz
15. Labbé, M., Michaud, F.: Memory management for real-time appearance-based loop closure detection. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1271–1276. IEEE (2011)
16. Labbe, M., Michaud, F.: Appearance-based loop closure detection for online large-scale and long-term operation. IEEE Trans. Robot. **29**(3), 734–745 (2013)
17. Labbe, M., Michaud, F.: Online global loop closure detection for large-scale multi-session graph-based slam. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), pp. 2661–2666. IEEE (2014)
18. Murillo, A., Sagüés, C., Guerrero, J.J., Goedemé, T., Tuytelaars, T., Van Gool, L.: From omnidirectional images to hierarchical localization. Robot. Auton. Syst. **55**(5), 372–382 (2007)
19. Ros, G., Sappa, A., Ponsa, D., Lopez, A.M.: Visual slam for driverless cars: a brief survey. In: Intelligent Vehicles Symposium (IV) Workshops (2012)
20. Wang, J., Yagi, Y.: Robust location recognition based on efficient feature integration. In: 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 97–101. IEEE (2012)
21. Wang, J., Yagi, Y.: Efficient topological localization using global and local feature matching. Int. J. Adv. Robot. Syst. **10** (2013)
22. Zhiwei, L., Xiang, G., Yanyan, C., Songhao, Z.: A novel loop closure detection method in monocular slam. Intell. Serv. Robot. **6**(2), 79–87 (2013)

Learning with Small Autonomous Robots

Joaquin Sitte, Ulf Witkowski and Reza Zandian

Abstract In this paper, we advance the thesis that educational robots can make a much larger contribution in the classroom than has hitherto been the case. However, to realise this potential, it is necessary to supply teachers with detailed guides for classroom robotic student activities for achieving specific curricular learning objectives. As illustration, we describe in detail three activities that use the capabilities of the inexpensive smartphone-based mobile robot Zorro developed by us.

1 Introduction

Constructing robots and making them perform some tasks attracts children of all ages and even many adults. Why this is so does not seem to be well understood yet. What we can say for sure is that for many playing with robots is fun. Efforts to take advantage of this attraction for the purpose of facilitating learning dates back at least to 1969 when Seymour Papert at MIT built and used the first-floor turtle robot[1] to visualise LOGO programming constructs to children. Papert saw small robots *as objects-to-think with* [4].

[1] A floor turtle is a real robot moving on the floor as opposed to the LOGO-simulated turtle, which moves on the computer screen to do line drawings.

J. Sitte
Queensland University of Technology, Brisbane, QLD 4000, Australia
e-mail: j.sitte@qut.edu.au

U. Witkowski (✉) · R. Zandian
South Westphalia University of Applied Science, Lübecker Ring 2,
59494 Soest, Germany
e-mail: witkowski.ulf@fh-swf.de; witkowski@fh-swf.de

R. Zandian
e-mail: zandian.reza@fh-swf.de

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_28

The LOGO programming language is meant to let children do something with a computer. In the process of doing, they learn the foundations of programming computers and analytic thinking. The turtle robots were little mobile vehicles remotely controlled from a computer by LOGO commands. Their shape was cylindrical with a hemispherical dome. They were called turtles in reference to the autonomous robots built in the late 1940s by neurophysiologist Grey Walter. The LOGO drawing instructions had been used to draw curves (turtle graphics) on a screen by incrementally moving an imaginary pen or a crawling creature (turtle). The turtle robots could physically visualise LOGO programs by moving according to the LOGO drawing instructions [1].

Technology has come a long way since 1969, and even if we still find many educational robots that resemble Papert's turtles, their capabilities have grown enormously and therewith their potential as *objects-to-think with*. Worldwide many schools offer extra-curricular activities in robotics such as participating in one of the many robot competitions and competitive team games such as robot soccer. The activities aim broadly at learning about science, technology, engineering and mathematics (STEM). However, for robotics to find a place in the regular curriculum, the learning objectives to which robotics contribute have yet to be fully identified. With the increased capabilities and increased complexity of educational robots comes a growing challenge on how to use small robots effectively in the classroom.

Flot and Shoop [3] consider robotics in education as a *content organiser* for the following topics:

- Math
- Engineering Process
  - Twenty-first Century Skill Sets

    Cooperation/Collaboration
    Teamwork
    Problem Solving
    Resource Allocation
    Time Management
  - Electronics
  - Scientific Methods
  - Technological Literacy
  - Communications, English and Language Arts
  - Persistence
- Computer Science

Out of this broad scope, teachers have to find the robotic activities for their students which will help them to achieve the specific learning objectives of the curriculum. Otherwise, the time students spend on robotic activities may serve learning objectives different from those of the primary subject. To effectively incorporate robotic activities to assist in achieving learning objective across the curriculum, the teacher needs to know what specific learning objectives can be supported with robotic experiments. For this task, the majority of teachers do not have the required specialist knowledge nor the time to acquire it.

So far, it is in computer science where most progress has been made in formulating learning objectives for robotic activities. Small mobile robots have been found helpful in the teaching of basic computer programming skills. Programming constructs such as conditionals and loops can be visualised by programming movement trajectories for a robot. This has been extended to other LOGO programming environments such as Smalltalk [2].

A robot is a computer with sensors and actuators to do things in the physical world.

As such, it can be an object-to-think with about the world around us. With their sensing, actuating and computing capabilities, small robots can be desktop laboratories for demonstrating the foundations of mechanics and electromagnetism, feedback control as well as image processing and computational intelligence. To illustrate how robot classroom activities can be designed to address specific learning objectives, we describe a set of three different learning objectives and the robot classroom activities to achieve them. In the next section, we describe the capabilities of the *Zorro* smartphone-based autonomous robot that we use for these activities. Sections 3, 4 and 5 describe the three activities and their learning objectives. Section 6 closes the paper with the conclusions.

The learning activities we describe are aimed at advanced primary students. The activities require no previous knowledge. For this reason, the robots' vision capability is not used in these activities. In the first two sample activities, the use of sensors is limited to the two incremental motor axis position encoders. In the third example, we use in addition additional magnetic field sensor. All activities require the robot to operate under program control. The activities can be adapted to different levels of programming skills of the targeted student group. For students without programming skills, complete programs can be provided where students can set the needed parameters through the smartphone's touch screen as well as displaying relevant measurements. Alternatively, when student has some programming skills, program skeletons can be provided where the students only have to modify some program segments.

2 Capabilities of the Zorro Smartphone Mobile Robot

The Zorro autonomous mini-robot consists of a mobile base that carries a smartphone that communicates with the mobile base over an USB connection. The Zorro mini-robot, shown in Fig. 1, is an inexpensive highly capable vision-guided autonomous mini-robot. An Android smartphone provides computing power, communication and camera [5] The Zorro mobility module (ZMM) gives the smartphone the actuators for moving around in space.

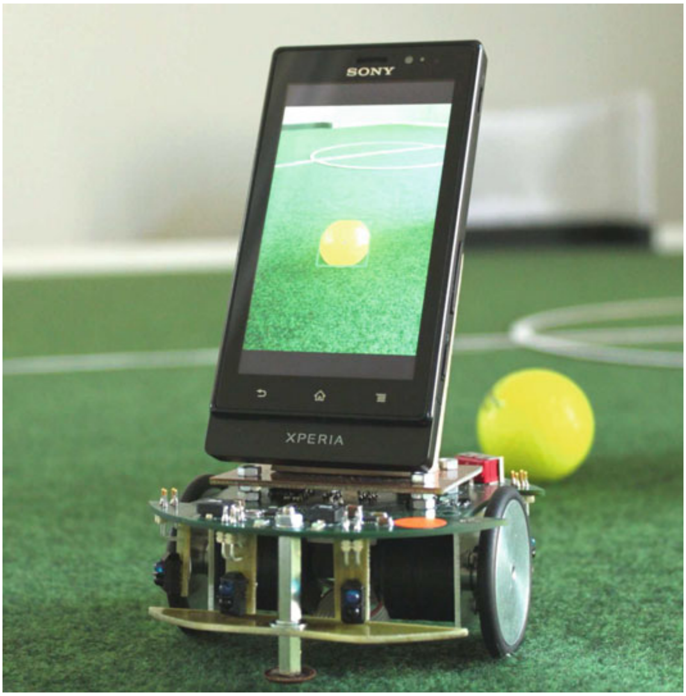


Fig. 1 The Zorro robot consisting of the mobility module and a smartphone

The ZMM is a two wheeled, differentially steered vehicle of cylindrical shape with a diameter of 100 mm. It has a socket on top to fit a smartphone that connects to the vehicle over USB. The ZMM has incremental position encoders on its motors, 8 infrared proximity sensors, a 3-axis accelerometer, gyroscope and compass (magnetometer). A microcontroller interprets motor commands and sensor reading commands sent by the smartphone to the ZMM over the USB connection. A battery pack provides power for the motors and the electronics on the module, while the smartphone is powered by its own batteries. The ZMM uses our own custom-made I/O board. A similar robot could be built with commercially available boards such as the IOIO® board or an Arduino® board.

3 Activity 1: Tell the Robot to Move to a Desired Location on a Table

The robots' mobility is the most basic capability that provides many opportunities for learning activities. For the first activity, we chose the learning of basic notions of distance and specification of the position of objects in 2D space with the help of a coordinate system. The traditional way of illustrating these concepts is with a ruler on a sheet of paper. With the help of a robot, the abstract problem of specifying a point on a plane is embodied in the physical task of directing a robot to move to a designated place within a flat surface area. The measuring and data recording capabilities of a small mobile robot enrich this experiment. An important difference of using a robot compared to the traditional ruler-based approach in this activity is that the cognitive functions that we perform without even noticing now become exposed by having to think about how the robot can do it.

3.1 Description

In this activity, students have to develop a method for directing a mobile robot to move from an arbitrary initial position to a specified target position on the table.

3.2 Learning Objectives

Basic concepts of analytical geometry are: specification of the position of points in 2D space, Cartesian and polar coordinate systems, Position vectors in 2D and finally Vector addition and subtraction.

3.3 Materials

The students have a mobile robot and an empty table or desktop. For this activity, the robot needs to be able to execute the following motion commands:

1. Move forward a given distance and
2. Turn on the spot by a given angle.

The robot can measure the distance travelled and the angle turned by reading the incremental motor axle position encoders.

3.4 Analysis of the Activity

The first problem to solve is finding a way of uniquely describing to the robot all available positions on the table. The robot then has to be able to travel to a specified position. To travel to the position, the robot has to follow a path. The student will discover that the path is a proxy for the position. If a single type of path can be found that will lead to any position by varying some parameters of the path, then the position of the point can be specified to the robot by the values of the parameters. These values become the coordinates of the position.

There are infinitely many ways to move from a place A to a place B. The simplest way is a straight-line movement from A to B. For this, the robot will have to know, first, in which direction to move and then how far. Note that in this activity, no other sensors will be used, and therefore, the robot is blind and cannot perceive where position B is.

The robot has to be told the direction and the distance to move. The two parameters, direction and distance, become the coordinates of B. A direction is measured as an angle with a reference direction (often chosen as north). Any position in table can now be specified by the parameters of the line from A to B. The problem is that when A changes so change the coordinates of B. The coordinates of B depend on the coordinates of A. As there are infinitely many positions for A, it means that there are infinitely many possible coordinates for B. What is needed is that each position on the table has unique coordinates. The solution to the dilemma is to pick one position A as a special reference position and specify the position of all points on the table by their coordinates relative to A. Such a reference point is called the origin. For instructing the robot to move from another position A′ to B, then the robot would need to know its position in relation to the origin A and with this information move to the origin and from there to B. By making use of trigonometry, the robot could avoid travelling to the origin and instead calculate the direction of B in relation to A′ and the distance from A′ to B. However, we cannot do this yet because the robot has no way to know his own position unless told by us. Before turning the attention to this problem, the student first needs to solve the problem of reaching any position from the origin.

When the robot is placed at the origin and given the coordinates of B, the robot does not know its own orientation, and hence, it cannot just turn the angle given by the coordinates of B and move forward the given distance to B. The robot needs initial help orienting it to a reference direction. The orientation of the robot is defined by the direction in which it will move when given the command to move forward by a given distance. The student can mark this direction on the body of the robot. Because the robot can measure how much it turns on executing a turning command and also how far it moved on executing a move forward command, the robot can keep track of its position and orientation from the time onwards when it started at the origin. At this stage in the activity, the robot can move to any position it is told on the table. Because it keeps track of its position as it moves, it can visit different positions in sequence without having to return to the origin, provided it can calculate the distance and direction to move for reaching a new position from its current position. At this time, the student will find it natural to represent the distance between two points as a vector.

The coordinates use so far are the polar coordinates (r, θ). An alternative trajectory for reaching any point B from the origin is to follow two successive straight-line paths that are not parallel. Most often, the two paths are chosen perpendicular to each other, for example, in the direction of the width and the length of the table. The parameters of the path are the distances, x and y, travelled in each direction. This method assigns each position on the table a pair of coordinates in a Cartesian coordinate system.

3.5 Procedure

To minimise the amount of programming, a program on the smartphone may be provided that allows the student to enter a sequence of displacements and rotations to be carried out by the robot. The program will also always display the current net displacement and orientation of the robot. The students should do at least the following:

1. Make the robot move from the origin to 10 or more different positions specified in polar coordinates and in Cartesian coordinates.
2. For each position make the robot return to the origin and measure with a ruler how much it misses the origin.
3. Make the robot visit several positions in sequence without returning to the origin. Here, the students will have to use vector algebra to calculate the necessary displacements.

According to the programming skills of the students, they may do all the required computations part of the programmed robot behaviour. Otherwise, they must do the computations separately.

3.6 Extensions

The activity can be extended to explore the notion of measurement error and error statistics. The capacity of the robot to record its sensor measurements enables it to collect error data and process it to calculate statistical parameters such as the mean, RMS error and error distribution curves. Basic error statistics can be collected, and the difference between accuracy and precision can be shown.

4 Activity 2: Robot Drag Race

The Zorro robot is pushed along by two electric DC motors. A DC motor turns when an electric current is made to flow through the motor. To make the robot move in a desired way, the right current has to be supplied at the right time. The only way to control the motion of the robot is through controlling the current. A technique for making a variable current flow through a resistive load, in our case a motor, is pulse width modulation (PWM). It consists of applying a constant voltage to the motor and switching the voltage on and off periodically at a high rate. The fraction of the time the voltage is on during one on–off period is called the duty cycle. The average current trough the motor is proportional to the duty cycle. With the appropriate electronics, the duty cycle can be set to any value between 0 and 100 %. Much can be learned about the motion of bodies by recording and analysing the movement of a robot in response to the applied motor currents.

4.1 Description

This activity is a desktop robot drag race with a twist. The robot has to reach the end of a straight race track in the shortest possible time. The twist is that it has to come to a halt exactly at the end of the track. The challenge is to find a sequence of PWM values that when applied by the robot will make it travel from the beginning to the end of the track as fast as possible.

4.2 Learning Objectives

Basic concepts of one-dimensional body motion dynamics are: time trajectories, the derived quantities of speed and acceleration, Discrete representation of time-dependent functions, Numerical computation of derivatives and finally Obtaining a qualitative understanding acceleration as the cause of motion change experienced by a body as preliminary to the study of Newton's laws.

4.3 Materials

The students have a mobile robot and an empty table or desktop. The race track is just a stretch of free space on the table, say, 1 m long. For this activity, the robot needs to be able to measure time intervals and displacement. The Zorro module provides distance travelled using the motor axis position encoders. The distance can be sampled at regular intervals, say every 20 ms. The distance values, in the form of encoder counts, collected by the smartphone over the duration of the race can then be transferred to a desktop computer for analysis. A simple way is to use a spreadsheet program to calculate differences between successive distance readings to compute and plot the speed as a function of time. Taking the difference between successive speed values gives the acceleration.

4.4 Procedure

First, the student will set a value for the motor current and its duration and will record the distance travelled along the track at regular time intervals resulting in a distance time trajectory. The student will plot the trajectory. For the trajectory, the speed will be calculated as the distance increases per time step divided by the duration of the time step. The speed curve (speed profile) is to be plotted for the whole trajectory. From there, the acceleration profile has to be calculated and plotted. This has to be repeated for several values of the PWM duty cycle and the length of time it is applied. By trial and error, the student has to find values and duration of the PWM signal so that the robot stops approximately at the end of the track. Once the student has become familiar with producing the diagram and in particular observing how the magnitude and duration of the PWM change the acceleration profile, the student is encouraged to change the PWM values during travel to achieve a shorter travelling time while still stopping at the end. These actions are all to be done under program control on the smartphone. According to the programming skills of the student group, they may be provided with a program template where they modify the part of the program that sets the PWM profile or they may be provided with a complete program where they can enter a PWM profile from the touch screen of the smartphone.

4.5 Analysis of the Activity

The smartphone controls the robot motors by sending a command to the ZMM to set a value for the duty cycle of the PWM. The current supplied to the motor is proportional to the duty cycle. In turn, torque generated by the motor is proportional to the current. The PWM signal remains at the set value until it is overwritten by a new value. Setting the value to zero effectively switches off the motor. Negative values for the duty cycle make the motor run in the reverse direction. Changing the duty cycle from zero to a nonzero value sets the robot in motion. As long as the motors deliver torque greater than the opposing friction, the robot will accelerate. The friction has three sources: the gearbox, motor friction and the friction of the wheels on the ground. The DC motor characteristics are such that as the speed of the motor increases, the torque diminishes.[2] Therefore, after some time, the robot will have gained enough speed to decrease the torque to be equal to the friction torque and the robot ceases to accelerate from there on. When the PWM duty cycles are set back to zero, only the friction torque remains and brakes the motors until they stop. To make the robot reach the end of the track, the magnitude of the PWM duty cycle has to be sufficient to set the motor in motion and the duration has to be

[2] This is because the motor also acts as a generator producing a voltage, the counter-electromotive force (EMF), that opposes the applied voltage, thereby diminishing the current through the motor.

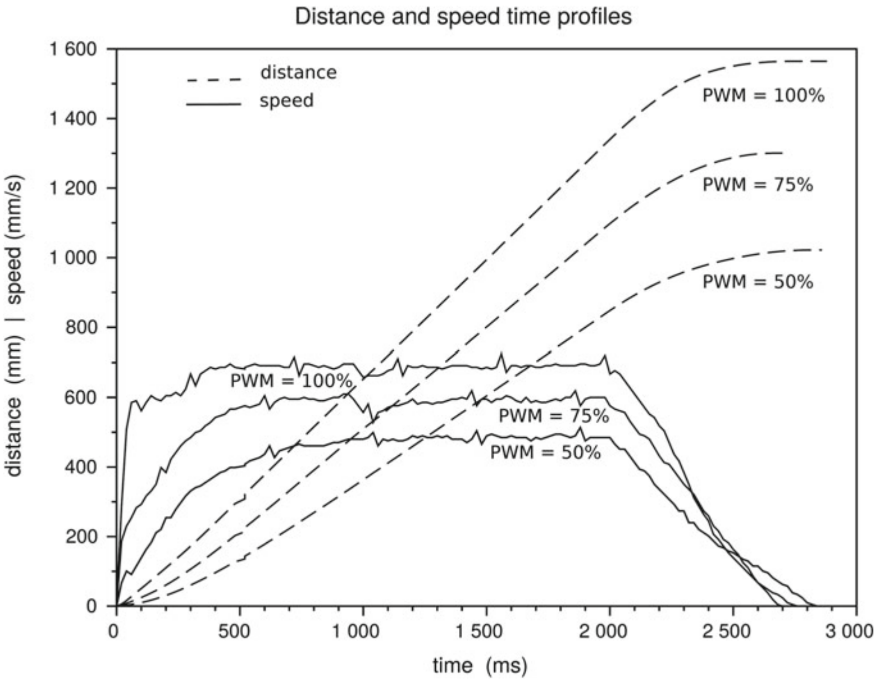


Fig. 2 Distance and speed time profiles for various PWM duty cycles (motor torques). The PWM is held constant for 2 s and then switched off. Friction stops the robot

long enough so that it is not stopped too early by friction. Many combinations of duty cycle value and duration will help to achieve this goal. However, the time to reach the end of the track will be different. Figure 2 shows the distance travelled, starting from rest, when the motors are switched on at constant PWM duty cycle for two seconds. The best combination can be found by the simple search strategy of starting from an initial guess increasing first the duty cycle and then the magnitude until the robot stops at the end of the track. From there on, the travel time may still be reduced by increasing the duty cycle and reducing the duration. Instead of reducing the duration of the initial duty cycle, a stronger braking may be applied by making the motor generate a reverse motor torque to add to the friction torque. It is helpful to plot the speed as a function of distance as shown in Fig. 3 instead of specifying a PWM profile as a function of time. It is also possible to set the PWM profile as a function of distance.

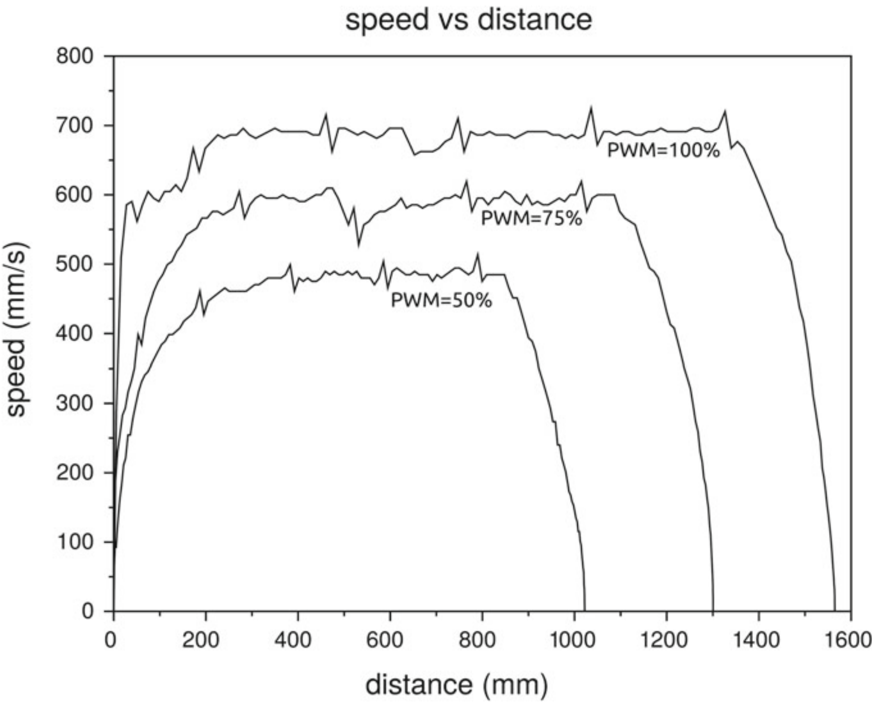


Fig. 3 Speed versus distance profiles for time trajectories shown in the previous figure

5 Activity 3: Show Magnetic Field Lines Around a Conductor

The inspiration for this activity is the classical experiment of visualising magnetic field lines with iron filings. In this experiment, iron filings are strewn on a sheet of white paper that covers a magnet. By slightly shaking the paper, the iron filings align with the local magnetic field revealing a streaked pattern suggestive of the magnetic field lines. A similar but somewhat more complicated experiment visualises the magnetic field lines around a current-carrying wire. A rectangular wire loop is connected to a current source (a battery with a current limiting resistor). The loop is positioned in a vertical plane and so that one side of the loop passes through a horizontal surface with iron filings. The horizontal surface is placed halfway on the side of the loop. This arrangement provides a vertical wire cutting through a horizontal surface so that half of the straight part of the wire loop is above the surface and half is below the surface. When a current flows in the wire, the iron filings align in concentric circles around the wire. This experiments shows the basic phenomenon of electromagnetism in its simplest form.

5.1 Description

The experiments with iron filings described above are simple ways to reveal the existence of magnetic field that are normally invisible to us. However, these experiments are unsuitable for a quantitative measurement of the magnetic fields. The Zorro robot has a magnetic field sensor like those used as digital compasses in smartphones and other consumer devices. The sensor allows the measurement of the magnetic field vector (magnitude and direction). In combination with the robots' capability of measuring its displacement in a plane, the robot can map a magnetic field in a plane. In this activity, the robot will move on a horizontal board around a vertical current-carrying wire that passes through the middle of the board, as described in the next section. While moving, the robot will sample the magnetic field vector and record it together with the measured position of the sample point.

5.2 Learning Objectives

Principles of electromagnetism are: Currents cause magnetic fields; The magnetic field is a vector field; Shape of the magnetic field in a plane perpendicular to a linear conductor; Right-hand rule for the relation between the direction of the current, the position in space and the direction of the magnetic field at that position. Related skills for doing this task are: measuring and representing vector quantities.

5.3 Materials

Ideally, we would use a 700 mm by 700 mm plywood board with a small hole in the centre for inserting a 1-m-long straight and stiff single-core electric cable. To provide enough stiffness, the diameter of the copper wire core should be around 2 mm. The cable is to be glued in position so that half of it protrudes above the board. The ends of the cable are to be connected to the DC current source by flexible cables with alligator clips. Care must be taken that upper supply cable does not hang down close to the straight wire. The plywood board is to be put on a support high enough that the lower part of the wire is clear of the ground.

On the software side, the robot has to be programmed to follow a sampling trajectory and storing the position and magnetic field measurements for posterior offline evaluation.

5.4 Procedure

The robot is to move on the board in the surroundings of the wire to sample the magnetic field. From the magnetic field measurements at the sample points along the path, a two-dimensional map of the magnetic field has to be constructed. The location of each sample point is to be calculated from the motor position encoders. The choice of the sampling path can facilitate the construction of the map or make it more difficult. The magnetic field sensor is mounted near the perimeter at the front centre of the ZMM. The magnetic field is strongest near the wire; therefore, the sensor should be able to come as close as possible to the wire. This can be achieved by making the robot face the wire.

However, this position does not allow the robot to move sideways while keeping the sensor close to the wire. Instead, the robot can move in the radial direction away from the wire and sample the magnetic field in the radial direction. At a certain distance from the wire, the robot may turn 90° and follow an arc of a circle centred at the wire.

After following the arc for say 30°, the robot turns 120° in the opposite direction so that it faces the wire again. From there, it moves towards the wire sampling the magnetic field. This can be repeated until 360° around the wire are covered. The sampling should be repeated with the direction of the current in the wire reversed. The curves for these different radial profiles of the magnitude and direction of the magnetic field should be compared. The students will discover that the radial magnetic field profiles are all very much the same, reflecting the radial symmetry of the situation. Once the sampled magnetic field vectors have been placed on a map, the student must identify trajectories where the magnetic field vector always points in the direction of motion of the robot, and these are the field lines. According to the student's skill level, they could be asked to match the variation in magnitude of the magnetic field with a simple analytical expression and also to find the relation between the direction of the current, the position vector of magnetic field sample point and the magnetic field vector.

5.5 Analysis of the Activity

This activity uses the robot to detect the invisible magnetic field around a current flowing in a wire. The activity will extend the students' previous experience with magnets to understand magnetism as originating from currents. Physicist and physics teacher have devised many classroom demonstrations for showing the effects caused by the interaction of currents and magnetic fields. These classroom demonstrations try to be visually striking for those who see it first time and thus make a lasting change in their interpretation of observed phenomena. In such demonstrations, the opportunity for direct engagement by students is often limited by either the effort to set up the experiment or the difficulties in extending the experiment to a quantitative investigation of the displayed phenomenon. In this activity, the robot provides the medium for a quantitative investigation of the causal relation between current and magnetic field. The robot allows the collection of sufficient data, with a moderate effort, for a quantitative description of the phenomenon.

6 Conclusion

Many educators in primary to tertiary education have searched for ways to use the interest that mobile robots arise in young and not so young people to motivate learning in the areas of science, technology, engineering and mathematics. Many ways have been found to do so as evidenced by the robotic competitions, robotic programs for schools and supporting educational robotic products on the market. A number of studies have shown that engaging students in building and programming mobile robots is a powerful motivator for learning. Still, most educational robotic activities take place in an extra-curricular setting. In this way, only a small fraction of the student population participates in such activities. A larger portion of the student population would participate if robotic activities could be linked to specific curricular learning objectives. This in turn requires that teachers know how to design robot activities that motivate students to achieve the learning objectives in their curricula. Most teachers will not have the means develop the appropriate robotic activities, and currently, there is not enough ready-made material they can draw on. Creating teacher support material requires the cooperation between teachers and roboticists. In this paper, we provide three examples of robotic activities aimed at achieving the stated learning objectives with the intention to motivate other educators and robotic researchers to contribute in the production of such educational material. The example activities are for learning with robots about other subjects instead of learning on robots about robotics. In these activities, the robot provides a single integrated multifunctional tool for uncovering the causal relations in basic phenomena.

The robot's capability for sensor data capture, preprocessing and storage allows for filtering and error compensation that are much harder to achieve with traditional experimental set-ups. The robot allows the student to engage directly in an investigation of greater depth and with less effort than in conventional experimentation. The student's effort will be rewarded by a feeling of discovery and accomplishment upon successful completion.

References

1. cyberneticzoo.com: 1969 the logo turtle, Seymour Papert, Marvin Minskyetal. http://cyberneticzoo.com/cyberneticanimals/1969-the-logo-turtle-seymourpapert-marvin-minsky-et-al-american/. Accessed 11 June 2015
2. Ducasse, S.: Squeak, Learning Programming with Robots. Apress (2005)

3. Flot, J., Shoop, R.: Foregrounding math, engineering, and computer science using robotics. In: Presentation given at the Technology Education and Engineering Association of Pennsylvania Annual Conference (2013)
4. Papert, S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books (1993). https://books.google.co.nz/books?id=AP8WBQAAQBAJH. Sekane, Z.S.: Localizing Objects during Robot SLAM in Semi-Dynamic Environments. In: IEEE Proceedings of the 2008 International Conference on Advanced Intelligent Mechatronics. ASME (2008)
5. Tetzlaff, T., Zandian, R., Drueppel, L., Witkowski, U.: Smartphone controlled robot platform for robot soccer and edutainment. In: Kim, J.H. (ed) Robot Intelligence Technology and Applications, vol. 3, pp. 505–518. Advances in Intelligent Systems and Computing, Springer (2015)

Design and Implementation of Double Passing Strategy for Humanoid Robot Soccer Game

Ping-Huan Kuo, Ya-Fang Ho, Te-Kai Wang and Tzuu-Hseng S. Li

Abstract The goal of this paper was to accomplish a technical challenge of double passing soccer game for humanoid soccer robots in RoboCup competition. Using only a vision sensor, the control strategies for the technical challenges of humanoid league in RoboCup are designed and presented. The vision system includes the color space setting, the object recognition, a simplified mean shift algorithm, and the target position derivation. Vision system works on the tasks of object recognition, which includes the goal, landmark poles, and the interval of two black poles. The computational time is reduced greatly by the mean shift algorithm and that time can be utilized to do other control strategies. With the proposed control strategies, humanoid robots can successfully complete the RoboCup double passing task. The successful experiment results demonstrate the feasibility and effectiveness of the proposed foot–eye coordination control scheme.

Keywords Double passing strategy · Foot–eye coordination control scheme · Humanoid robot soccer game

P.-H. Kuo · Y.-F. Ho · T.-K. Wang · T.-H.S. Li (✉)
aiRobots Laboratory, Department of Electrical Engineering,
National Cheng Kung University, Tainan City, Taiwan, ROC
e-mail: thsli@mail.ncku.edu.tw

P.-H. Kuo
e-mail: coll22000@hotmail.com

Y.-F. Ho
e-mail: makinosakuya@hotmail.com

T.-K. Wang
e-mail: greatmutaxx@gmail.com

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_29

1 Introduction

Many researches of robots have developed in the world for decades of years. Different kinds of robots are designed for corresponding specific uses. It is believed that the development of robots has mainly been in the areas of their interacting with humans and helping them with their needs. So a lot of methods have been proposed for human–robot interaction and communication [1–4]. Since the hardware computational ability improves fast with the size of the devices become smaller and smaller, building a robot for our own is not a difficult thing anymore. The goal of this paper was to design and implement a team (aiRobots-V) of humanoid robots for robot soccer games. A humanoid robot is a highly integrated system including mechanism design, passive sensors, vision system, power supply system, communication system, and algorithms in software programming. The robot soccer game presents a dynamic and complex environment. It could be a challenging platform for multi-agent research, involving many topics such as motion generation [5–9], motion control, image processing algorithm, localization, and collaboration strategy planning. They aim to promote artificial intelligence (AI), robotics, and related fields. The ultimate goal of the RoboCup [10] project is as follows: By the year 2050, develop a team of fully autonomous humanoid robots that can play against the human world champion soccer team and win the game. We attend these competitions in order to enhance and stimulate our technique and development of the humanoid robot. At the same time, we can learn and try to make strides by competing with other teams from different countries. We will also attend the challenge games the 3 on 3 soccer games of RoboCup 2010 humanoid league.

Besides, the robots need to be autonomous and intelligent during the soccer games. Thus, strategies of many cases have to be taken into consideration. Strategies are also programmed in the embedded computer. Robots will communicate with each other via wireless network to accomplish the collaboration strategies. The role of each robot will switch dynamically by itself to reach the best efficiency for offense and defense. The aim of this paper was to develop a fast processing vision system and robust strategy system for not only humanoid robot soccer competition but also challenge games of double passing.

2 Vision System

The vision system used in this paper includes two essential parts: object recognition and simple mean shift algorithm, which are described in the following subsections.

2.1 Object Recognition

In the soccer playing field, there are five important objects: ball, landmark pole, goal, black pole, and interval space between the black poles. These need to be recognized for the technical challenge since the goals and landmark poles have been changed in RoboCup 2010, and the interval between the black poles is important for the dribbling event. The goal, landmark poles, and interval detections are presented as follows.

2.1.1 Goal Recognition

After the object with the same color goal is segmented by the YUV lookup table, the following steps are applied to check whether it has the characteristics of a goal.

Step 1 The width and height of the goal must be wider and taller than a set threshold.

Step 2 The goal must be situated on the game field, so the color under the goal probably belongs to the field or the lines. It is necessary to check this condition. Two places will be checked: the lower right corner and the lower left corner. For each corner, there is a 3 × 3 window. Each pixel will be scanned and checked to see whether it is the color of the game field or a white line or something else. An illustration of the goal-checking windows is shown in Fig. 1a. However, the view may be occluded by other robots during the game time, as shown in Fig. 2. Since the robots must be mostly black according to the rules of the RoboCup, the black color also has to be checked in those two windows.

Step 3: Because of the shape of the goal, there should not be any color belonging to the goal. There is a 5 × 5 window in the middle of the goal to scan and check the color in that window to make sure it is not the color of the goal. It is illustrated in Fig. 1b.

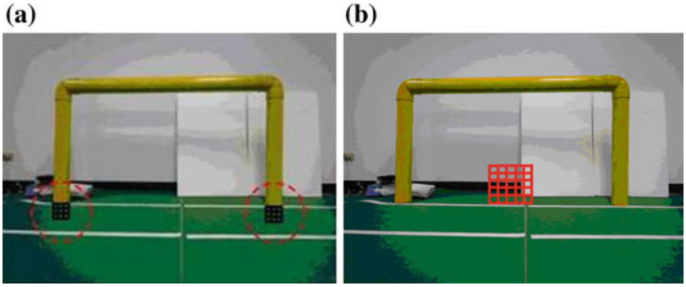


Fig. 1 Illustration of the goal-checking windows of **a** 3 × 3 **b** 5 × 5 pixels

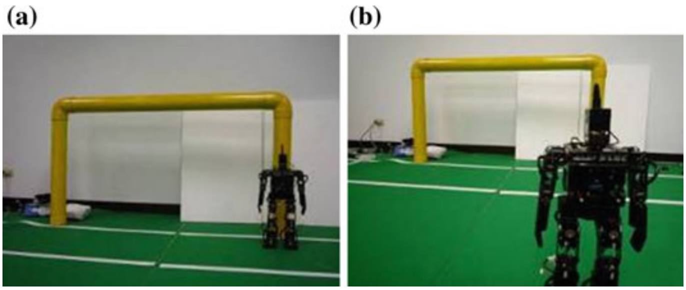


Fig. 2 Examples of a view of a goal occluded by another robot

In steps 2 and 3, every pixel in the window is scanned and a reward function is adopted. If the scanned pixel meets the expectation, the score goes up; otherwise, the score goes down. The reward function can be described as follows:

$$\begin{cases} G_{\text{score}} = G_{\text{score}} + \alpha_G, & \text{if the pixel fits the right color} \\ G_{\text{score}} = G_{\text{score}}, & \text{otherwise} \end{cases} \tag{1}$$

where G_{score} is the total reward and α_G is the reward value; the recognition succeeds if $G_{\text{score}} > \beta$. Otherwise, the recognition fails. β is a threshold defined by the user. The results of the goal recognition are shown in Figs. 3 and 4. Nevertheless, when the robot stands very close to the goal, it can only see part of the goal. Figure 5 gives an example of a partial view of the goal.

In this case, the robot will be asked to raise its head, and then, the view will be clearer. It is able to determine which side of the goal the robot sees by sampling of the goal and calculating the vector, as shown in Fig. 6. After the color of the goal is segmented out, a vertical sampling from the left to the right of the image will be done as shown in Fig. 6a and e show. Then, the vector is calculated by:

$$V_g = \begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} x_R \\ y_R \end{bmatrix} - \begin{bmatrix} x_L \\ y_L \end{bmatrix} \tag{2}$$

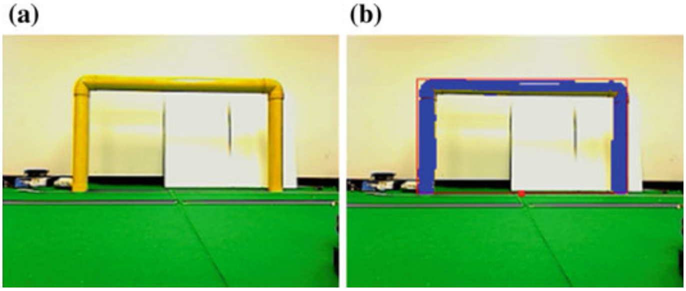


Fig. 3 The result of the goal recognition. **a** The original image, **b** result of the goal recognition

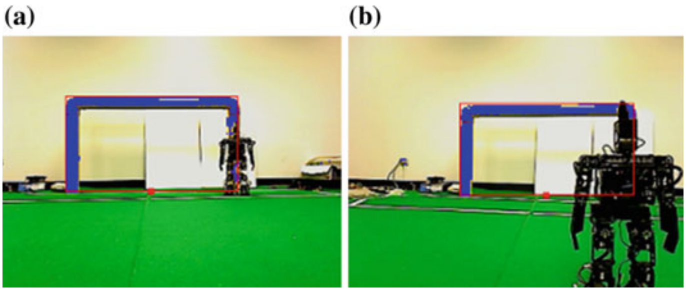


Fig. 4 Goal recognition with occlusion of another robot. **a** Occlusion of a robot (far), **b** occlusion of a robot (near)

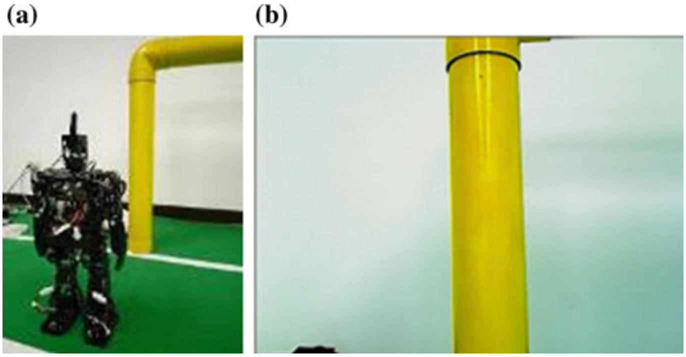


Fig. 5 An example of a partial view of the goal. **a** The relative positions of the robot and goal, **b** view of a robot

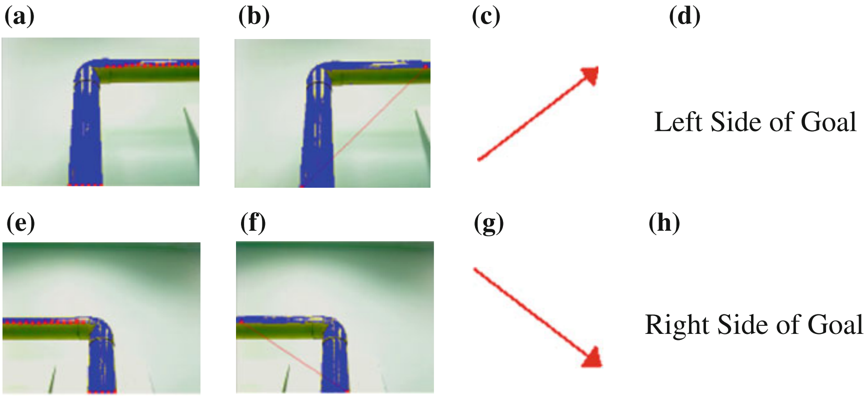


Fig. 6 The recognition result for a partial view of the goal. **a** Sampling points, **b** calculating the vector, **c** the vector, **d** result, **e** sampling points, **f** calculating the vector **g** the vector, **f** result

where (x_R, y_R) is the coordinate of the rightmost sampled point, (x_L, y_L) is the coordinate of the leftmost sampled point, and $V_g(x_g, y_g)$ is the result vector. If the vector points to the upper right corner, this means that the robot sees the left side of the goal. If the vector points to the lower right corner, this means that the robot sees the right side of the goal.

2.1.2 Interval Detection

In the double passing event, there are two more black poles of 30 cm in radius and 90 cm in height. One pass must go through the interval of the two black poles. The interval is about 120 cm in width. Figure 7a shows the two black poles. The detection of the interval is via a sampling method, and the method is divided into several steps.

Step 1 Do the sampling vertically, and the sampling interval is 10 pixels. A group of sample points will be collected. Figure 7b demonstrates the sampling.

Step 2 Check from the first point. If the distance between this point and the next point is long enough and the vector points to the bottom, one black pole is found. Keep scanning the points until there is a vector that orients the opposite way, and then one black pole is detected. All the points will be checked in this step. The distance is given by

$$D_i = |y_i - y_{i-1}|, \quad i = 1 \sim 32 \tag{3}$$

where D_i is the distance between ith point and $(i-1)$th point and y_i represents the coordinate value of the y-axis. Since the sampling interval is 5 pixels, the difference in the x-axis from every point to the next one will be constant. It is not necessary to take the x-axis into consideration. The computation is reduced in this way. The vector is given by

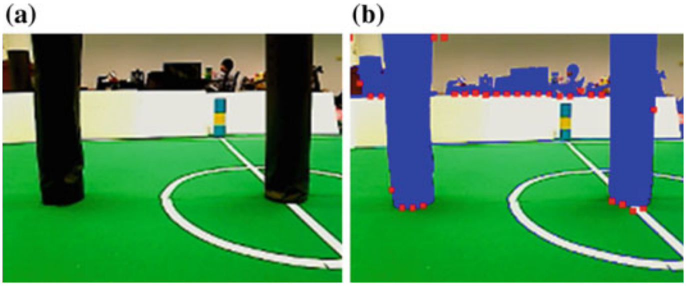
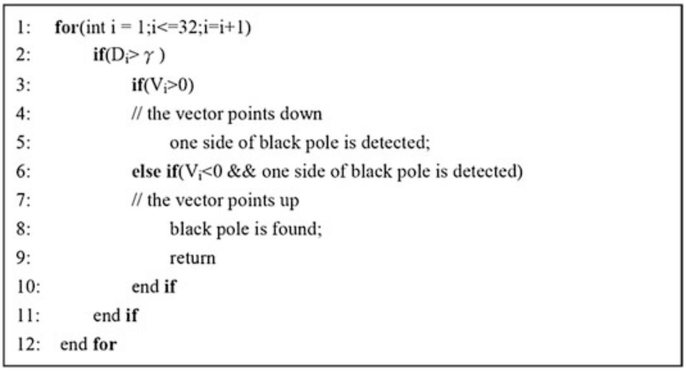


Fig. 7 **a** The position of the black poles in a double passing game. **b** The sampling points of the *black color*



```
for(int i = 1;i<=32;i=i+1)
    if(D_i> γ )
        if(V_i>0)
        // the vector points down
            one side of black pole is detected;
        else if(V_i<0 && one side of black pole is detected)
        // the vector points up
            black pole is found;
            return
        end if
    end if
end for
```

Fig. 8 Pseudo code of step 2

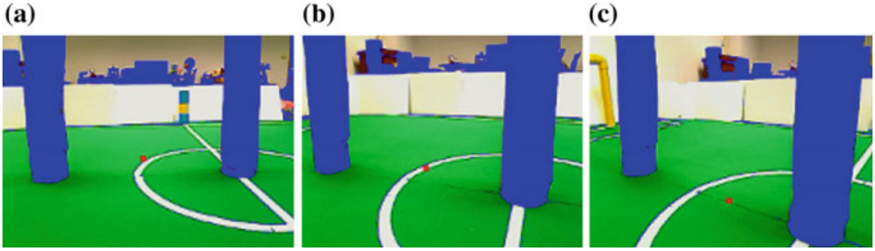


Fig. 9 Results of interval detection of two black poles from different views

$$V_i = y_i - y_{i-1}, \quad i = 1 \sim 32 \tag{4}$$

where V_i is the vector from $(i - 1)$th point to ith point. As denoted in (4), the x-axis is not taken into consideration. The pseudo code of step 2 is shown in Fig. 8.

Step 3 If there are two black poles found in Step 2, the width of their interval can be calculated. The center of the interval is then pointed out. The interval detection is completed, and the result is shown in Fig. 9.

2.2 Simple Mean Shift Algorithm

After the recognitions are done, the next step is to keep tracking the target. There are many papers that discuss different kinds of methods or algorithms for tracking. But actually, the target is not that big, so it is not necessary to always search the whole image. Besides, searching the entire image for a small target is a waste of time and increases the computation of the system. So a mean shift algorithm is adopted to solve this problem. The mean shift algorithm has been widely discussed

[11–15]. The algorithm only searches the region of interest, and the size of the region is adaptive to the variation in size of the target. This section shows how the algorithm works.

2.2.1 Principal Concept of Mean Shift

The main idea and intuitive description of mean shift algorithm can be illustrated as shown in Fig. 10. The red dots shown in Fig. 10 mean the global data distribution. The algorithm aims to find the maximum distribution of the data. In the case of the target tracking in this research, red dots are considered as pixels whose color in YUV channel is similar to the color of the target. The main difference is that the data in our case will be concentrated but not scattered because every target has its distinguishing color organization. For example, if there is only one orange ball on the game field, it is impossible to see the scattered orange color distributed on the field. The data distribution is shown in Fig. 11. Another different point is that, unlike in most cases, the shape of the region is square in this research. The square shape is convenient for searching digital image data.

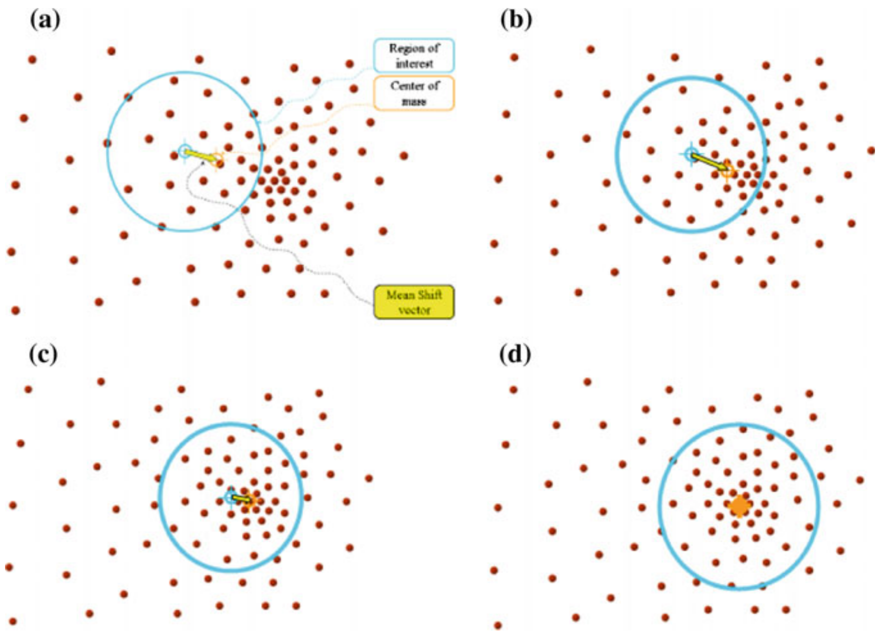


Fig. 10 The principal concept of mean shift algorithm. **a** Calculate the center of mass (COM) of the data in the region of interest, **b** shift the region of interest to the COM and then calculate the new COM, **c** keep shifting the region of interest as in (**b**), **d** the shifting procedure is completed when COM is just the center of the region

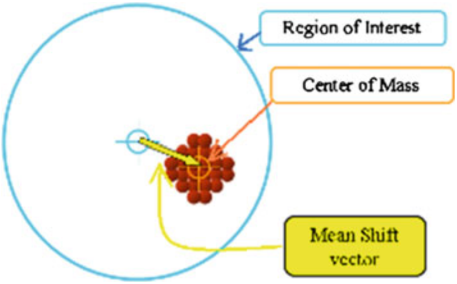


Fig. 11 The data distribution in the case of the ball target

2.2.2 Simplified Mean Shift Algorithm

The first step is to initialize the region of interest. In the beginning, scanning all the pixels of an image is needed because the data only gather together somewhere but are not scattered in the whole image. We can say that the initial region of interest is the whole image. When the target is found, the region is then determined and will keep shifting with the target until the target is lost. If the target disappears in the region of interest, the procedure goes back to search the entire image. A diagram of a simplified mean shift algorithm is shown in Fig. 12. The center of mass $C(x_c, y_c)$ is computed by

$$X_c = \frac{\sum_{i=1}^{n} x_i}{n}, \quad \forall x_i \neq 0 \tag{5}$$

$$Y_c = \frac{\sum_{i=1}^{n} y_i}{n}, \quad \forall y_i \neq 0 \tag{6}$$

where x_i and y_i are the coordinate values of the pixels of the target on the image plane and n is the total number of pixels of the target. Then, the mean shift vector $V_m(x_m, y_m)$ is given by

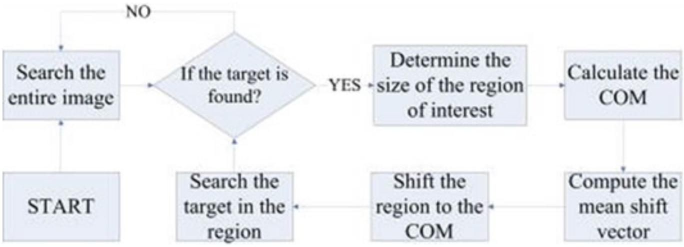

NO
Search the entire image
If the target is found?
YES
Determine the size of the region of interest
Calculate the COM
START
Search the target in the region
Shift the region to the COM
Compute the mean shift vector

Fig. 12 Diagram of a simplified mean shift algorithm

$$V_m = \begin{bmatrix} x_m \\ y_m \end{bmatrix} = \begin{bmatrix} X_c - X_{\text{region}} \\ Y_c - Y_{\text{region}} \end{bmatrix} \tag{7}$$

where $\left(X_{\text{region}}, Y_{\text{region}}\right)$ is the center of the region of interest. So the new center of region $\left(X'_{\text{region}}, Y'_{\text{region}}\right)$ is as follows:

$$\begin{bmatrix} X'_{\text{region}} \\ Y'_{\text{region}} \end{bmatrix} = \begin{bmatrix} X_{\text{region}} + x_m \\ Y_{\text{region}} + y_m \end{bmatrix} \tag{8}$$

The size of a target that is shown on the captured image varies when the distance from the robot and target changes. A fixed size of a searching region is not reasonable. So the size of the region or, more specifically, the side length of the square will be adaptive to the size of the target. The side length is adjusted according to the size of the target. The adaptive mechanism is given by

$$L_{\text{region}} = k \times S_{\text{target}} \tag{9}$$

where L_{region} is the side length of the region and the S_{target} is the size information of the target. S_{target} may be different from other targets. For instance, S_{target} is the radius in the case of the ball and the width in the case of the pole and goal. k is a related coefficient. With the adaptive region, the algorithm is more efficient and time-saving. The results are shown in Fig. 13. The tracked target in this case is a ball. There are two squares in every picture in Fig. 13.

The outer square is the region of interest, and the inner square is the boundary of the target after the recognition is successful. It is obvious that the side length of the region of interest varies according to the area of the target. When the ball is far from the robot, the region becomes smaller; otherwise, it becomes bigger. The efficiency is raised in this way.

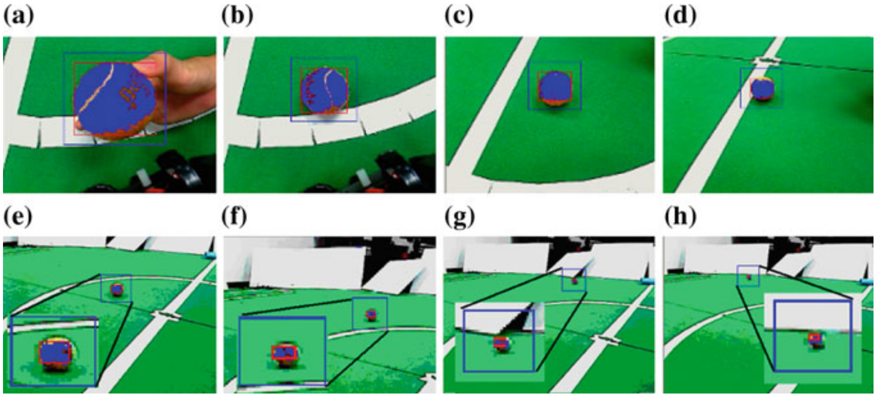


Fig. 13 The result of a simplified mean shift algorithm

3 Control Strategy

The control strategy is divided into two independent processes because there are two players and their tasks are a little different. In order to achieve the strategy, target tracking by the head of the robot is required because many procedures in the strategy need to track the target, walk toward it, and calculate the distance from it. So, once the current target is found, keeping it in the image all the time is important. Here, the fuzzy logic controller (FLC) implemented in [16] is adopted to solve the problem. The FLC is done by controlling the two motors as pan and tilt on the head. In this way, the following strategies can run smoothly. The flowchart strategy of robot Player A is shown in Fig. 14a. As shown in Fig. 14a, the process of a control strategy for Player A is partitioned into three parts. Process 1 is to make the first pass to Player B and then walk toward the left pole until the distance from Player A is less than a certain length and turn back to face the game field to wait for the ball to be passed back. Process 2 is to pass the ball through to the interval between the two black poles, walk toward the goal until the distance from the goal is less than a certain length, and then wait for the ball to be passed back. One of the steps in process 2 is to dribble the ball in the direction of the goal. This step can make the second pass of Player A easier according to the many times of experiments. The third process is to make a goal when the ball is passed back from Player B, just as in a normal soccer game.

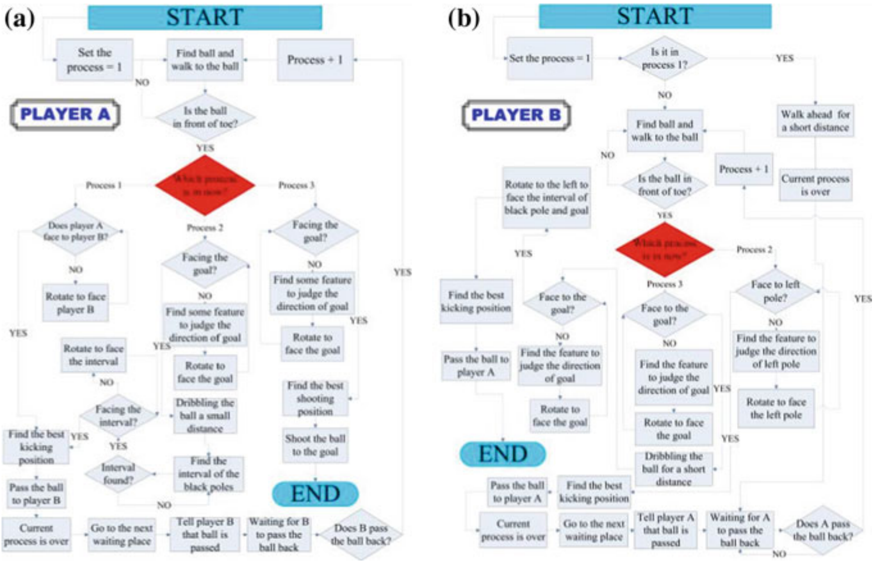

(a)
START
Set the process = 1
Find ball and walk to the ball
Process + 1
NO
PLAYER A
Is the ball in front of toe?
YES
Which process is in now?
Process 1
Process 2
Process 3
Does player A face to player B?
NO
Rotate to face player B
YES
Rotate to face the interval
NO
Facing the interval?
YES
YES
Interval found?
NO
Find the best kicking position
Pass the ball to player B
Current process is over
Go to the next waiting place
Tell player B that ball is passed
Waiting for B to pass the ball back
Does B pass the ball back?
Facing the goal?
NO
Find some feature to judge the direction of goal
YES
Rotate to face the goal
Dribbling the ball a small distance
Find the interval of the black poles
Facing the goal?
NO
Find some feature to judge the direction of goal
YES
Rotate to face the goal
Find the best shooting position
Shoot the ball to the goal
END
YES
(b)
START
Set the process = 1
Is it in process 1?
YES
NO
PLAYER B
Find ball and walk to the ball
NO
Is the ball in front of toe?
YES
Process + 1
Walk ahead for a short distance
Current process is over
Which process is in now?
Process 2
Process 3
Rotate to the left to face the interval of black pole and goal
YES
Find the best kicking position
Pass the ball to player A
Face to the goal?
NO
Find the feature to judge the direction of goal
Rotate to face the goal
Face to the goal?
NO
Find the feature to judge the direction of goal
YES
Rotate to face the goal
YES
Dribbling the ball for a short distance
Face to left pole?
NO
Find the feature to judge the direction of left pole
Rotate to face the left pole
YES
END
Pass the ball to player A
Find the best kicking position
Current process is over
Go to the next waiting place
Tell player A that ball is passed
Waiting for A to pass the ball back
Does A pass the ball back?
NO

Fig. 14 Flowchart for the double passing strategy of the robots. **a** Player A and **b** Player B

4 Experiment Results

The full procedure for double passing is demonstrated in Figs. 15, 16, 17, 18, and 19. Figure 15 shows the first pass from robot Player A to robot Player B. The first kick is very slight because the two players are very close to each other. Figure 16

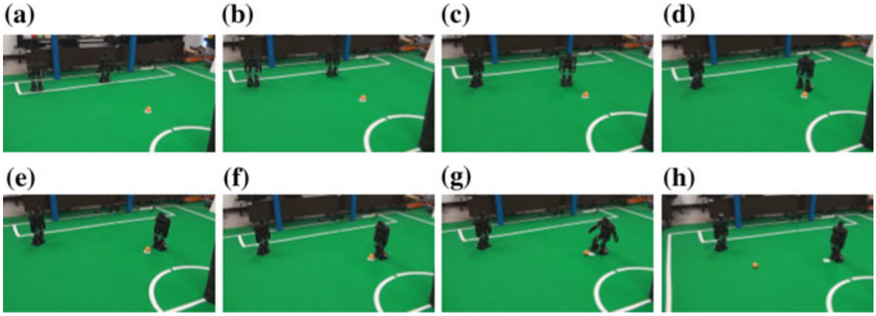


Fig. 15 The first part of the result of the double passing strategy

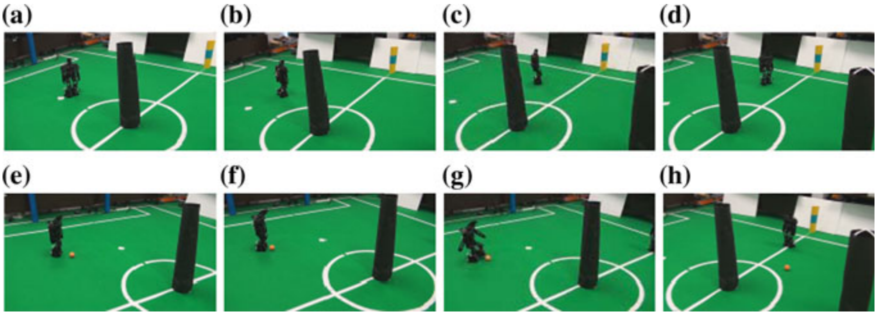


Fig. 16 The second part of the result of the double passing strategy

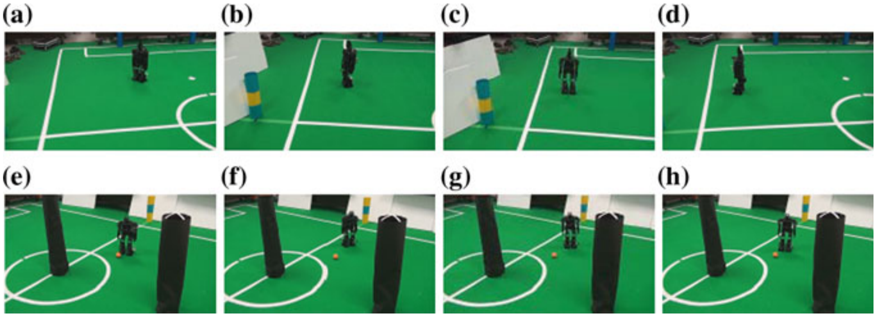


Fig. 17 The third part of the result of the double passing strategy

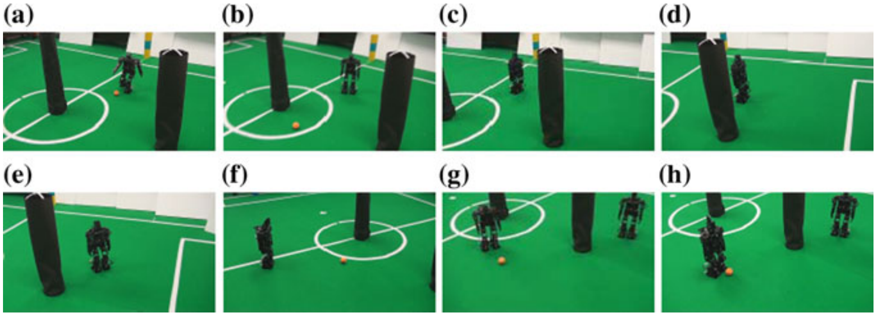


Fig. 18 The fourth part of the result of the double passing strategy

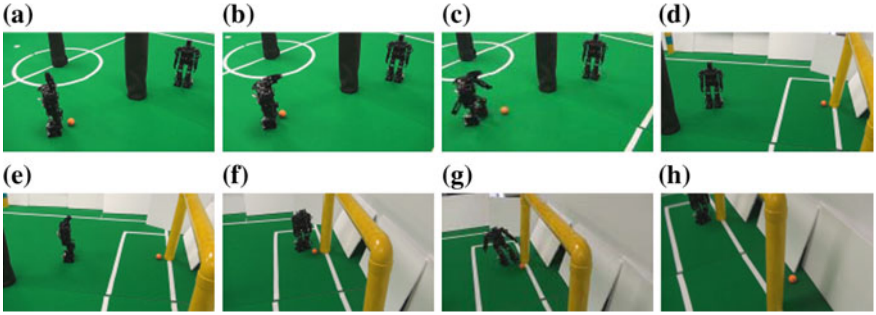


Fig. 19 The last part of the result of the double passing strategy

shows the second pass from Player B to A. The strength of the kick is stronger than in the first pass. The strength is not as strong for shooting the goal because the ball must be controlled in the field. Figures 17 and 18 demonstrate the third pass from Player A to B. The strength of the kick is like the second pass to maintain the ball inside the field. Figure 19 shows the last task of the double passing. Player A shoots the ball into the goal. After the ball is in the goal, the two robots will stay at their position. Finally, the double passing is then completed after the last shot.

5 Conclusions

This paper has proposed a complete and systematic vision system and foot–eye coordination control for humanoid soccer robots. The major objects on the field can all be recognized by different methods. A simplified mean shift algorithm is proposed for target tracking, and it searches the region of interest instead of always searching the entire image. In addition, the area of the region is adaptive according to a change in size of the target. It can save more computation time than the original mean shift algorithm by the adaptive searching region. Finally, the foot–eye coordination control strategies are presented for the double passing challenges. The control strategies allow the robots to successfully implement the technical challenge of the humanoid league of the RoboCup. For the dribbling challenge, the proposed control scheme provides a complete and effective strategy for robots to accomplish the tasks. Additionally, the proposed foot–eye coordination control scheme is used in humanoid soccer robots, which entered the humanoid league of the RoboCup in 2010 and won second place in the technical challenge of RoboCup 2010.

Acknowledgment This work supported by Ministry of Science and Technology of Taiwan, R.O. C, under Grants MOST 103-2221-E-006-252, MOST 104-2221-E-006-228-MY2, and the aim for the Top University Project to the National Cheng Kung University (NCKU) is greatly appreciated.

References

1. Stiefelhagen, R., Ekenel, H.K., Fugen, C., Gieselmann, P., Holzapfel, H., Kraft, F., Nickel, K., Voit, M., Waibel, A.: Enabling multimodal human-robot interaction for the Karlsruhe humanoid robot. IEEE Trans. Robot. **23**, 840–851 (2007)
2. Spexard, T.P., Hanheide, M., Sagerer, G.: Human-oriented interaction with an anthropomorphic robot. IEEE Trans. Robot. **23**, 852–862 (2007)
3. Lee, J.M., Choi, J.S., Lim, Y.S., Kim, H.S., Park, M.: Intelligent and active system for human-robot interaction based on sound source localization. In: Proceedings of International Conference on Control, Automation and Systems, ICCAS, pp. 2738–2741 (2008)
4. Rivera-Bautista, J.A., Ramirez-Hernandez, A.C., Garcia-Vega, V.A., Marin-Hernandez, A.: Modular control for human motion analysis and classification in human-robot interaction. In: Proceedings of 5th ACM/IEEE International Conference on Human-Robot Interaction, HRI, pp. 169–170 (2010)
5. Su, Y.-T., Chong, K.-Y., Li, T.-H.S.: Design and implementation of fuzzy policy gradient gait learning method for walking pattern generation of humanoid robots. Int. J. Fuzzy Syst. **13**(4), 369–382 (2011)
6. Li, T.-H.S., Su, Y.-T., Lai, S.-W., Hu, J.-J.: Walking motion generation, synthesis, and control for biped robot by using PGRL, LPI and fuzzy logic. IEEE Trans. Syst. Man, Cybern. B **41**(3), 736–748 (2011)
7. Kuo, P.-H., Ho, Y.-F., Lee, K.-F., Tai, L.-H., Li, T.-H.S.: Development of humanoid robot simulator for gait learning by using particle swarm optimization. In: Proceedings of 2013 IEEE International Conference on System, Man, Cybernetics, pp. 2683–2688 (2013)
8. Hong, Y.-D., Park, C.-S., Kim, J.-H.: Stable bipedal walking with a vertical center-of-mass motion by an evolutionary optimized central pattern generator. IEEE Trans. Ind. Electron. **61** (5), 2346–2355 (2014)
9. Li, T.-H.S., Kuo, P.-H., Ho, Y.-F., Kao, M.-C., Tai, L.-H.: A biped gait learning algorithm for humanoid robots based on environmental impact assessed artificial bee colony. IEEE Access. **3**, 13–26 (2015)
10. RoboCup. http://www.robocup.org/
11. Pooransingh, A., Radix, C.A., Kokaram, A.: The path assigned mean shift algorithm: a new fast mean shift implementation for colour image segmentation. In: Proceedings of 15th IEEE International Conference on Image Processing, ICIP, pp. 597–600 (2008)
12. Wang, H., Yang, B., Tian, G., Men, A.: Object tracking by applying mean-shift algorithm into particle filtering. In: Proceedings of 2nd IEEE International Conference on Broadband Network and Multimedia Technology, IC-BNMT, pp. 550–554 (2009)

13. Li, Y.H., Pang, Y.G., Li, Z.X., Liu, Y.L.: An intelligent tracking technology based on Kalman and mean shift algorithm. In: Proceedings of Second International Conference on Computer Modeling and Simulation, ICCMS, pp. 107–109 (2010)
14. Yafeng, Y., Hong, M.: Adaptive mean shift for target-tracking in FLIR imagery. In: Proceedings of Wireless and Optical Communications Conference, WOCC, pp. 1–3 (2009)
15. Liu, Y., Peng, S.: A new motion detection algorithm based on snake and mean shift. In: Proceedings of Congress on Image and Signal Processing, CISP, pp. 140–144 (2008)
16. Su, Y.T., Hu, C.Y., Li, T.H.S.: FPGA-based fuzzy PK controller and image processing system for small-sized humanoid robot. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, SMC, pp. 1039–1044 (2009)

Development of Motion Management System for the Robot Soccer Using Multiple Humanoid Robots

Jae-Man Lee, Hye-Yeun Chun, Yusupov Dilmurod, Keong-Seok Ko and Tae-Koo Kang

Abstract We address the motion management system to apply it to the robot soccer game. The soccer system has a vision system, a host computer which manages the strategies and position control of the soccer robots, and a communication system between a host computer and three soccer robots. The player robots can take 32 motions according to the game situation. To avoid the obstacles such as the opponent player robots, we proposed the novel motion control method which consists of CM and PL-AMS. The CM decides the avoid direction considering the position and size of the obstacles. PL-AMS selects the appropriate avoidance motion by the fuzzy logic. In addition, we have the strategy for the game situation such as the offense and defense by using the seven subregion divisions. Moreover, to prevent the ping-pong effect, we provided the buffer region between subregions.

1 Introduction

Robot soccer is a good example of a multi-agent system in a real-world application which can be moderately abstracted. Multi-agent systems deal with research subjects such as cooperation protocol by distributed control, effective communication and fault-tolerance while having efficiency of cooperation, adaptation, and robustness, and being in real-time.

Soccer robot control can be divided into three control schemes. The first scheme is a remote-brainless soccer robot system in which a host computer controls the robots by commanding the robots motion. The second scheme is a vision-based one which controls the robots by processing information using the vision with a

J.-M. Lee (✉) · H.-Y. Chun · Y. Dilmurod · K.-S. Ko · T.-K. Kang (✉)
Department of Information and Communication Engineering, Sangmyung University,
Cheonan-si, Chungcheongnam-do 330-720, Republic of Korea
e-mail: ejm303@naver.com
URL: http://ittech.smuc.ac.kr

T.-K. Kang
e-mail: tkkang@smu.ac.kr

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_30

supervisor processor and sends commands to robots directly. The other scheme is a robot-based one, wherein each autonomous robot can make a decision based on information from its sensors and, if needed, can send information to others. Robots are designed for autonomous operation [1].

Among the three schemes, only the remote-brainless soccer system has been dealt with, in this paper as it is easy to develop, though the robots are designed for the robot-based system. A control structure, several behaviors, and actions are developed for a soccer robot. Modified zone defense as a basic strategy and several special strategies for fouls are applied to multiple humanoid robots for ANDROSOT'15 (Robot World Cup Soccer Tournament). A soccer game of the ANDROSOT played by 3 robots from each team is used as a test bed for the multi-agent systems. In Sects. 2 and 3, we described the soccer robot system and control structure in detail.

2 Soccer Robot Systems

There are various methods including the hardware, such as CPU, actuators, sensors, and software, such as control algorithm, strategies to make a soccer system for ANDROSOT which consists of multiple humanoid robots per a team. Therefore, we have to select the appropriate hardware and software among the various methods.

In this section, we focus on the overall architecture of the soccer robot system and the soccer robot system. The specific configurations and functions are explained in detail as follows.

2.1 Architecture of Area Soccer Robot System

Figure 1 shows the overall architecture of the soccer system for ANDROSOT. As shown in Fig. 1, the soccer robots having motion mechanism, a vision system, a host computer which manage the strategies and position control of the soccer robots according to the offense or defense situation, and a communication system between a host computer and three soccer robots are needed for robot soccer game.

2.2 Soccer Robot System

Figure 2 illustrates a robot system we used in the soccer game. As shown in Fig. 2b, the soccer robot has 34 cm × 18 cm × 12 cm dimension and 17 DOF [3]. It involves a 2-axis gyro sensor, an IR sensor, and a tilt sensor to detect the

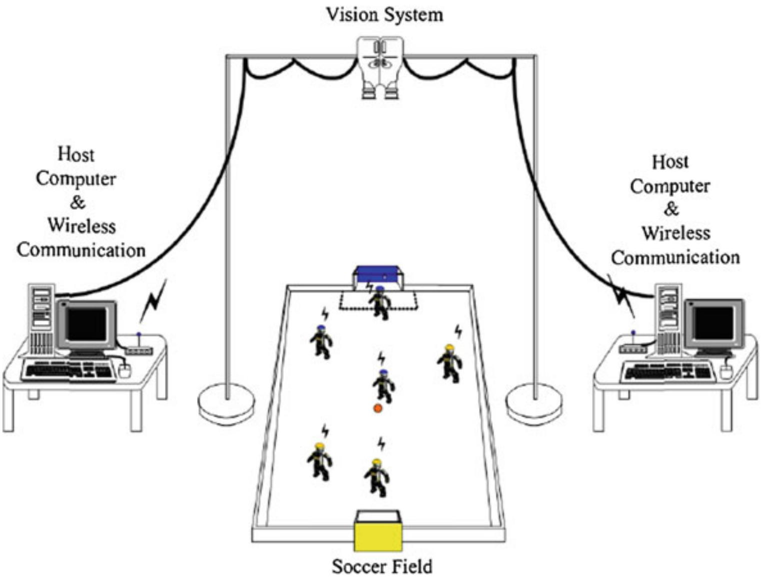


Fig. 1 Overview of the soccer system including a vision system [2]

inclination fall down of the robot body. To control the servo motor, we used the MR-C3024FX (Fig. 2a) which can control 24 servo motors and has various control interface, such as a gyro sensor, a Bluetooth module, and a remote control.

Basically, each robot system can make over 120 motions and in the case of our soccer game, we use the 32 various motions including the block motions for a

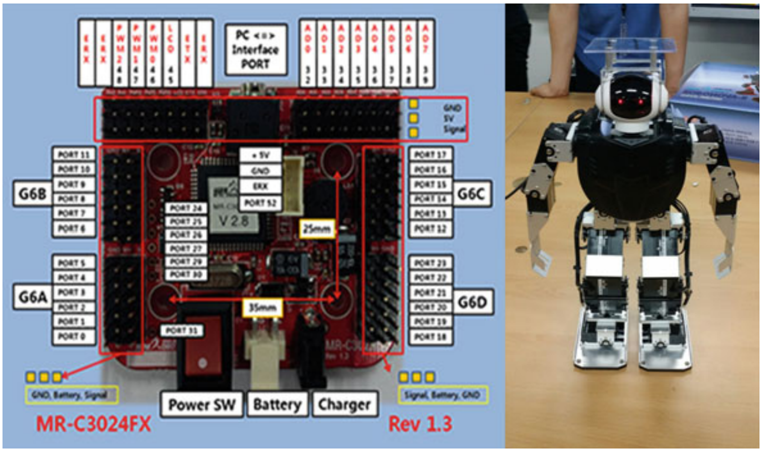


Fig. 2 A soccer robot system and its controller

goalkeeper robot. Each robot system selects the appropriate motion corresponding to the game situation.

3 Motion Control and Path Planning Algorithm

In this section, we deal with the control algorithms for the obstacle avoidance method and the optical path generation. Detailed methods for the control structure are described as follows.

3.1 Motion Control for Obstacle Avoidance

To avoid collisions between our player robots and the obstacles such as the opposite player robots, we need to determine the avoidance direction and the avoidance motion of our player robots. Therefore, we proposed novel method called Complex Measure (CM) and the Fuzzy Logic-based Avoidance Motion Selector (FL-AMS) to solve the given problem.

3.1.1 Decision of Avoidance Direction Using Complexity Measure (CM)

To calculate the avoidance direction, we use the environment information such as obstacle's position and size as the input data. To decide the avoidance direction, we propose the complexity measure (CM) as follows:

$$CM = \begin{cases} \sum_{l=1}^{n} k_l + 2^{(\pi/2 - \theta_l)} + w_l, & \theta_l > 0 \\ -\sum_{l=1}^{n} k_l + 2^{(\pi/2 + \theta_l)} + w_l, & \theta_l < 0 \\ \sum_{l=1}^{n} 2k + w_l^k \sum_{l=1}^{n}, & \end{cases} \tag{1}$$

In (1), the CM is determined by three factors: distance from the robot, width of the obstacle, and angle from the robot to the obstacle. From the CMs, we can get the final avoidance direction as

$$Direction = sign(\sum_{all\,obstacle} CM) = \begin{cases} Direction < 0, & \text{turn left} \\ Direction > 0, & \text{turn right} \end{cases} \tag{2}$$

The avoidance direction is decided by the summation of all CMs for all obstacles. In (2), the CMs for obstacles located on the left side of the robot have negative values, and the CMs for obstacles located on the right side of the robot have positive values. Therefore, if the direction is a positive value, then the robot turns to the left, and if the direction is a negative value, then the robot turns to the right.

3.1.2 Decision of Avoidance Motion Using Fuzzy Logic-Based Motion Selector

After we decide the avoidance direction for the obstacle, avoidance motion has to be selected depending on the environment information. Avoidance motions have to be varied under width of obstacle and open space that a robot can avoid. Therefore, we propose the fuzzy logic-based avoidance motion selector (FL-AMS) [4].

In order to perform the fuzzification, a width difference between an obstacle width and a robot width w_i and space difference between avoidance space width and robot width S_i are set up as the input variables. The term sets of the input and output variables are selected as follows:

$$T(w_i) = \{A_1, A_2, A_3, A_4, A_5\} = \{NL, NS, Z, PS, PL\}, \tag{3}$$

$$T(s_i) = \{B_1, B_2, B_3, B_4, B_5\} = \{NL, NS, Z, PS, PL\}, \tag{4}$$

$$T(f_i) = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7\} = \{ST, SS, SF, RS, RF, TS, TF\}, \tag{5}$$

where fuzzy sets $A_1, A_2, A_3, A_4,$ and A_5 are, respectively, denoted as negative large (NL), negative small (NS), zero (Z), positive small (PS), and positive large (PL) for the input variable w_i. Fuzzy sets $B_1, B_2, B_3, B_4,$ and B_5 are also, respectively, denoted as negative large (NL), negative small (NS), zero (Z), positive small (PS), and positive large (PL) for the input variable d_i. Fuzzy sets $C_1, C_2, C_3, C_4, C_5, C_6,$ and C_7 are, respectively, denoted as Stop (ST), Slip step + Slip step (SS), Slip step + Forward step (SF), Rotation step + Slip step (RS), Rotation step + Forward step (RF), Turning step + Slip step (TS), and Turning step + Forward step (TF) which are the combinations of two motions in Slip step (S), Forward step (F), Rotation step (R), and Turning step (T) excluding the Stop motion (ST) for the output variable f_i.

The partitions and the shapes of the membership functions are shown in Fig. 3, where triangular-type membership functions and fuzzy singleton-type membership function are, respectively, used for the input and output variables. As shown in Fig. 3, each input is divided into five subsets, so that two inputs will construct 25 fuzzy rules and output value of fuzzy logic determining the walking motion type is constructed in seven membership functions. The fuzzy rule base used for the proposed FL-AMS system is described in Table 1, where each rule can be described by

$$\begin{aligned}
&\text{Rule} \quad R(j_1, j_2): \\
&\quad \text{IF } w_i \text{ is } A_{j_1} \text{ and } d_i \text{ is } B_{j2}, \text{ then } f_i \text{ is } C_{f(j_1, j_2)} \\
&\quad\quad j_1 \in \{1,2,3,4,5\}, j_2 \in \{1,2,3,4,5\}, f(j_1, j_2) \in \{1,2,3,4,5,6,7\}
\end{aligned} \tag{6}$$

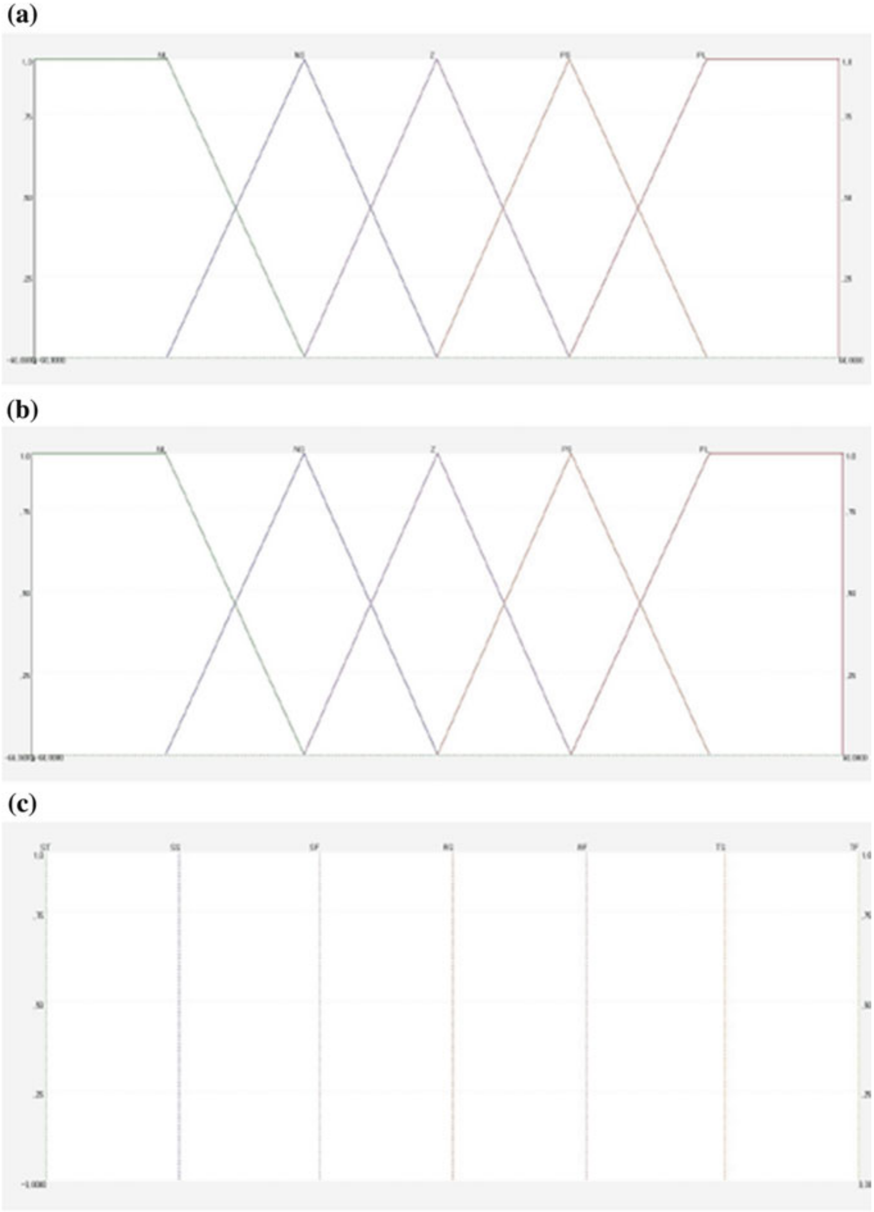


Fig. 3 Membership functions of FL-AMS: **a** input variable w_i, **b** input variable w_i, and **c** output variable w_i

Table 1 Fuzzy rule for FL-AMS

f_i		S_i				
		NL	NS	Z	PS	PL
w_i	NL	ST	SS	SF	SF	SF
	NS	ST	SS	SF	SF	SF
	Z	ST	RS	RF	RF	RF
	PS	ST	RS	RF	RF	RF
	PL	ST	TS	TF	TF	TF

To determine the final output of the FL-AMS, we use the weighted average method described by

$$f_i = f(w_i, s_i) = \frac{\sum_{j_1=1}^{j_1=5} \sum_{j_2=1}^{j_2=5} u(j_1, j_2) \cdot v(C_{f(j_1, j_2)})}{\sum_{j_1=1}^{j_1=5} \sum_{j_2=1}^{j_2=5} u(j_1, j_2)} \tag{7}$$

where $v(C_{f(j_1,j_2)})$ is the crisp value of the fuzzy set, $C_{f(j_1,j_2)} \cdot u(j_1,j_2)$ is the fire strength of the rule $R(j_1,j_2)$ and can be described by

$$u(j_1, j_2) = \min(\mu_{A_{j_1}}(w_i), \mu_{B_{j2}}(s_i)) \tag{8}$$

Based on the width difference between obstacle width and robot width w_i and space difference between avoidance space width and robot width s_i, seven evaluation values $f_i, i \in \{1, 2, 3, 4, 5, 6, 7\}$ for seven avoidance motions can be determined by the FL-AMS. The walking motion type consists of seven motions, ST, SS, SF, RS, RF, TS, and TF which are the combinations of two motions in sidestep walking (S), forward walking (F), rotation walking (R), and turning (T) excluding the stop motion (ST). Table 1 shows the fuzzy rule for the walking motions.

3.2 *Determination of Optimal Path*

To reach the objective point, the optimal path generation from current location to target location is necessary. We can examine this problem in terms of efficiency and safety.

The path planning method can be largely divided into two parts: (1) Pint-To-Point (PTP) and (2) Complete Coverage (CC). PTP path planning is searching the optimal path from a start point to an end point through points. The objective of PTP path planning method is that we find the path having efficiency in terms of minimum energy consumption or risk reduction. On the other hand, the objective CC path planning is that a robot covers the all sections as far as possible for a given area. Therefore, we use the PTP path planning method because it is suitable for our purpose playing the soccer game.

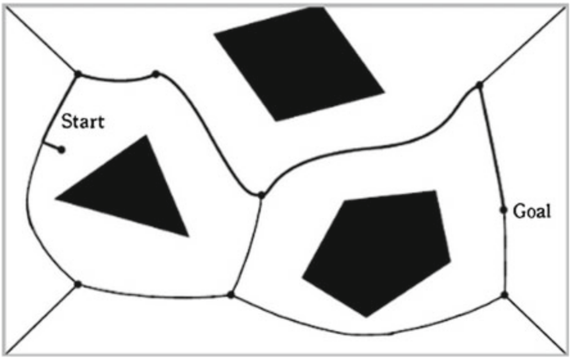


Fig. 4 Voronoi diagram-based path planning method

In addition, an algorithm conducting the PTP path planning is needed. To achieve this purpose, we consider algorithms in terms of efficiency and safety. In terms of efficiency, the robots playing the game can move using the relatively short course to reach the target position under offense or defense situations. In terms of safety, the robot in the soccer game does not foul like colliding with the robots of the opposite team.

To satisfy the above-mentioned conditions, we use a Voronoi diagram to generate the optimal path. Figure 4 shows an example of the Voronoi diagram-based path planning. In mathematics, a Voronoi diagram is a partitioning of a plane into regions based on distance to points in a specific subset of the plane. That set of points (called seeds, sites, or generators) is specified beforehand, and for each seed, there is a corresponding region consisting of all points closer to that seed than to any other. These regions are called Voronoi cells.

Moreover, the PTP path planning can be used in the free-ball situation such as free-kick, goal kick, and penalty kick due to the non-interference property of the Voronoi diagram between the robots belonging our team and those of the opposite team.

3.3 Strategies for the Game Condition

The basic strategy for the soccer robots was the zone defense or offense. However, the applied strategies were needed to the complicated situations in the game.

Each robot has its own role such as a striker, defender, and a goalkeeper. Firstly, we divided the ground into the three parts according to its role. At the beginning of development, the concept of zone defense was that the robot selected an appropriate action among basic actions if only the ball is located in its own area and the other robots did not select any actions because the ball was out of their areas. But, this concept had two problems: (1) motion conflict of nearby two robots when the ball is located in one of the boundary areas and (2) ping-pong effect [5]. The corresponding formation will be formed according to the position of the ball, which

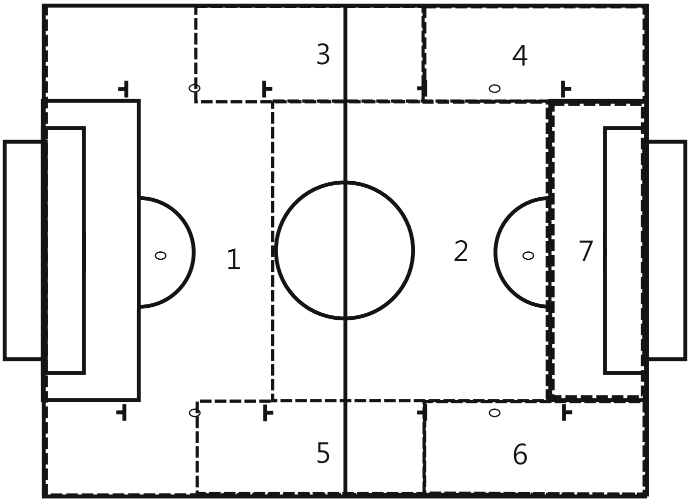


Fig. 5 Subregions for formation and play mode

causes a serious problem when the ball moves back and forth between two regions. The player robots will keep changing formation and consume their energy unnecessarily.

To solve the problem, we made a play mode and a formation through the field division according to the position of the ball. The play mode consists of offense, defense, and free-ball mode and depends on the relative position among the positions of our opposite robots and the location of the ball except the free-ball mode.

Player robots change the formation according to the position of the ball within seven subregions as shown in Fig. 5. If the ball locates within the defender goal area (region 2, 4, 6, and 7), then the major mission of the robots in these regions is that kick the ball out of defender goal area. When the ball is close to the opponent goal area or the position of our player robot with respect to the ball position is closer than that of the opponent player robot, we regard this situation as the attack mode and form the attack formation. As mentioned above, to prevent the ping-pong effect, we provide the buffer region between subregions, which keep the formation of the team although the ball moves back and forth between two subregions.

4 Conclusion

We address the motion management system and its application to the robot soccer game. The soccer system has a vision system, a host computer which manages the strategies and position control of the soccer robots, and a communication system between a host computer and three soccer robots. The player robots can take 32 motions according to the game situation.

To avoid the obstacles such as the opposite player robots, we proposed the novel motion control method which consists of CM and PL-AMS. The CM decides the avoidance direction considering the position and size of the obstacles. PL-AMS selects the appropriate avoidance motion by the fuzzy logic.

In addition, we have the strategy for the game situation such as the offense and defense by using the seven subregion divisions. Moreover, to prevent the ping-pong effect, we provide the buffer region between subregions. In the demonstration tournament, we played games with two periods of up to seven minutes (or until a goal was scored), and on average 4–5 goals were scored in each match, which was essential for us to achieve a spectacular game.

References

1. Kim, J.-H., Shim, H.-S., Kim, H.-S., Jung, M.-J., Choi, I.-H., Kim, J.-O.: A cooperative multi-agent system and its real time application to robot soccer. Int. Conf. Robotic Autom. **1**, 638–643 (1997)
2. www.fira.net/
3. http://minirobot.co.kr/
4. Kang, T.-K., Choi, I.-W., Park, G.-T., Lim, M.-T.: Local environment recognition system using modified SURF-based 3D panoramic environment map for obstacle avoidance of a humanoid robot **10** 1–13 (2013)
5. Wu, S.L., Tung, K.C., Liou, Y.R., Lin, W.H., Wu, M.H.: A multi-agent algorithm for robot soccer games

Small-Size Robot Platform as Test and Validation Tool for the Development of Mechatronic Systems

Philipp Kemper, Thomas Tetzlaff, Ulf Witkowski, Reza Zandian, Michel Mamrot, Stefan Marchlewitz, Jan-Peter Nicklas and Petra Winzer

Abstract The development of mobile robots or in general autonomous mobile systems is a complex task. These systems are characterized by a set of sensors and actuators as well as software running on computing hardware. Depending on the foreseen tasks of designing a mobile system, the overall complexity of the design may range from a few integrated components to several tens of sensors and actuators including distributed information processing mechanism. Besides system modeling and simulations, the design process involves system testing by performing several experiments. A difficult part in this testing phase is to identify the relation between the observed system behavior and the underlying implemented hardware component or part of software. With our test environment and a specific validation method, an iterative design cycle can be enabled that helps to speed up the design process as well as achieve the system behavior that is defined by the specification.

P. Kemper · T. Tetzlaff · U. Witkowski (✉) · R. Zandian
South Westphalia University of Applied Science, Lübecker Ring 2,
59494 Soest, Germany
e-mail: witkowski.ulf@fh-swf.de; witkowski@fh-swf.de

P. Kemper
e-mail: kemper.philipp@fh-swf.de

T. Tetzlaff
e-mail: tetzlaff.thomas@fh-swf.de

R. Zandian
e-mail: zandian.reza@fh-swf.de

M. Mamrot · S. Marchlewitz · J.-P. Nicklas · P. Winzer
University of Wuppertal, Gaußstraße 20, 42097 Wuppertal, Germany
e-mail: mamrot@uni-wuppertal.de

S. Marchlewitz
e-mail: marchlew@uni-wuppertal.de

J.-P. Nicklas
e-mail: nicklas@uni-wuppertal.de

P. Winzer
e-mail: winzer@uni-wuppertal.de

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_31

1 Introduction

Challenge of the design of mechatronic systems, for example autonomous robots, is their inherent complexity. Usually, the overall function of a robot is realized by implementing and combining appropriate sensors, actuators to enable interaction with the environment, and software code that realizes sensor data processing and the robot's behavior. There are different approaches on how to design a mechatronic system. Zheng et al. [1] have performed a survey to recognize common design methods of mechatronic systems. According to the paper, common techniques are sequential design model, V-model, and hierarchical design model. A comparison between the models is performed by the authors and concluded that the mechatronic systems are basically formed by domain-specific components, which requires multidisciplinary collaboration among mechanical, electrical, and software engineers. Casner et al. [2] proposed a novel design approach for mechatronic systems based on multidisciplinary design optimization. Their approach is an extension of V-model including the modeling and the optimization process which aims to improve the prototype design. Furthermore, Qamar et al. [3] discuss a model-integrated approach which combines system-level modeling and domain-specific modeling in order to support coevolution of design models (electronics, mechanical, and software). They concluded that the proposed integration infrastructure makes it possible to combine the system model and domain-specific design models within the development process of the products. Another model-based approach is proposed by Heikkilã et al. [4]. In this approach, they use a knowledge base (KB) decision support system to simplify decision making for designing autonomous robotic systems. In this technique, already available, performed, and well-established models are stored in a database, which can be used to find the most suitable design solutions and software services.

In this paper, we are focusing on resource-efficient robot platforms. These platforms use minimum resources in terms of number of sensors, actuators, and processing devices to realize a desired behavior. The motivation is to have a cost-efficient system as it is necessary for wide usage of robot platforms in our daily life, e.g., vacuum cleaners, lawn mower, glass cleaners, and elderly assisting. Even relatively simple robot devices as the mentioned vacuum cleaners have to provide a good performance to be accepted by the users. A typical challenge for vacuum cleaners is to reach all areas inside a room or a flat in a short time. Therefore, it is useful that robots follow a navigation strategy that is well organized, e.g., to navigate a meander pattern, in contrast to driving randomly. Considering the operation of the targeted robot platform, this type of systems can be described as an autonomous mechatronic system (AMS). Main challenges of designing autonomous mechatronic systems such as the resource-efficient robot platforms are caused by the interdependencies needed to be considered. Another important impact is the fact that every system depends on its environment [5]. In the case of AMS, the environment can be very complex and is classified in static (barriers like walls), semidynamic (mobile barriers like chairs), and dynamic (moving humans) obstacles

[6]. An additional difficulty arises as there is a need to develop a robot for the consumer market at low cost. Due to the growing complexity and the increasing number of involved components and requirements, the costs of individual tests increase. One possible solution to decrease the cost of testing is to automate the generation, execution, and evaluation of tests.

Today, there are a few approaches to design AMS with a focus on robotics. Steinbauer et al. [7] suggest using a modularized design approach to overcome this complex challenge. The authors propose to apply the technique according to the desired physical functions of the robots. The communication between the individual modules can be enabled by predefined interfaces. In summary, [7] focuses on the modeling especially on the hardware configuration of the robot. [8] describes a formal development process based on OOADM (object-oriented analysis and design methodology) using UML (unified modeling language). The suggested method based on UML consists four phases: requirements analysis, system analysis, design and coding, and finally testing. UML diagrams (case diagram, activity diagram, and class diagram) are used to model the system to support the system analysis phase as well as the design phase. In summary of the foregoing methods to design and test robots, it can be noted that there are several research activities and innovative approaches to support the development process; nevertheless, a comprehensive methodology does not exist. Therefore, the development of a usable method to test and validate the design of autonomous mechatronic systems is necessary. A key part to develop this method is the development of a hardware platform as well as a test bench to support the design and validation process.

The paper is organized as follows: In Sect. 2, our test platform consisting of a modular mobile robot platform and the tele-workbench environment is introduced. In addition, an indoor tracking approach based on UWB communication modules is discussed. Section 3 explains our approach for testing and validation of a designed robot platform and robot behavior. Performed example experiments are given in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Test Environment and Data Recording

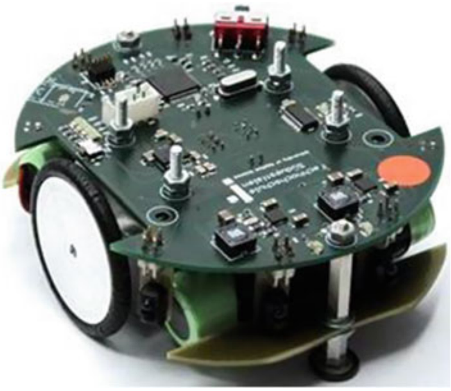
The test environment consists of a mobile robot platform and the tele-workbench. Depending on the task(s) of the mechatronic system, it is possible to perform single-robot or multirobot experiments. One option to evaluate the performance of a robot simply is to observe visually an experiment. However, with this approach, there is no option to derive performance parameters. In addition, it is usually difficult to identify the underlying reason of an observed specific behavior of the robot, i.e., the mapping from a behavior to a sensor, to actuator, or to piece of software usually cannot be done. By recording an experiment, the available data can be analyzed in detail and it will be possible to identify the reason for the (unexpected) behavior. To have numerous data available for a detailed analysis, a video of the experiment is recorded. In order to complement the video data, also several sensor data and status data of the robot are recorded. Data are transmitted via a wireless connection to a data server and saved together with a time stamp to be able to superimpose the different types of data with video data. In the following subsections, a robot platform, as well as the tele-workbench infrastructure, is introduced.

2.1 AMiREsot Robot Platform

The robot platform that is used on the tele-workbench platform has a diameter of 10 cm (see Fig. 1). This small size has been chosen to be able to perform robot experiments with moderate space requirements. Due to the small size, neither special lab space is required nor need special safety regulations. In addition, the platform meets the size regulations of the AMiREsot robot soccer league [9] that enables valuable options for the robot platform in teaching and edutainment [10]. The robot follows a modular concept. In its basic configuration, a base PCB is used that integrates a microcontroller (STM32F103RBT6) and several sensing options (8 IR sensors, accelerometer, gyroscope, compass sensor, and incremental encoder of the motors). Two DC motors (Faulhaber, 6 V, 2619S006S, 22:1) are used to drive the platform. Maximum speed of the robot is about 0.5 m/s. The microcontroller is used to capture all sensor data of the base platform, as well as to control the motor speed and to provide the interface for extensions. Power supply is realized by two Li-ion accumulators (3.7 V, 2.6 Ah) resulting in an operating time of several hours.

By adding additional boards to the baseboard of the robot, the sensing and processing options can be increased. For example, an FPGA (field programmable gate array) module with integrated camera is developed to support parallel image processing. Another option is to extend the robot by attaching a smartphone to the

Fig. 1 Small-size AMiREsot robot platform with a diameter of 10 cm



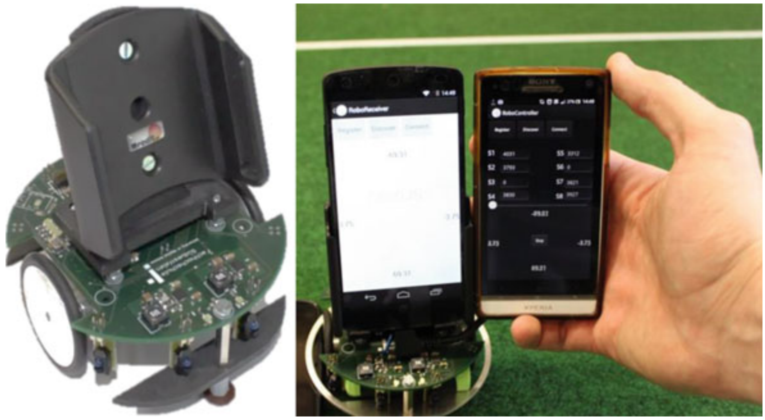


Fig. 2 Smartphone integration into the robot platform via a standard socket (*left*) and wireless remote access for system setup and data recording via Wi-Fi or Bluetooth (*right*)

base platform. For this, a mounting option via smartphone socket is provided (Fig. 2 left).

The communication is realized by a USB interface connecting the microcontroller and the phone. A bidirectional communication is supported, i.e., the smartphone can read sensor data from the baseboard and it is possible to set motor speed or other parameters inside the microcontroller. By having the smartphone integrated into the platform, an onboard high-quality image or video capturing and processing can be performed. The smartphone can also be used for implementation of desired robot behavior. Due to the available computing power and development tool, a bunch of methods can be used to implement easily a behavior like an autonomous environment exploration. The communication option of a smartphone can be used to integrate the robot platform into a Wi-Fi network or to establish point-to-point communication based on Bluetooth or Wi-Fi. In Fig. 2 (right), an ad hoc communication using Wi-Fi is depicted. Users are able to remotely control the robot platform via a smartphone, or observe the robot behavior based on displayed sensor data and even the transmitted camera data. In the past, we have used the robot platform with integrated smartphone to develop robot soccer algorithms [10].

2.2 *Tele-Workbench Environment*

The tele-workbench platform (Fig. 3) is intended to automate the execution of experiments in single-robot and multirobot applications and to facilitate concurrently an easy analysis on the resulted data by providing a visualization of occurring events during the experiment. Even though testing and validation of a designed robot platform can be done through simulation, still a real experiment is necessary

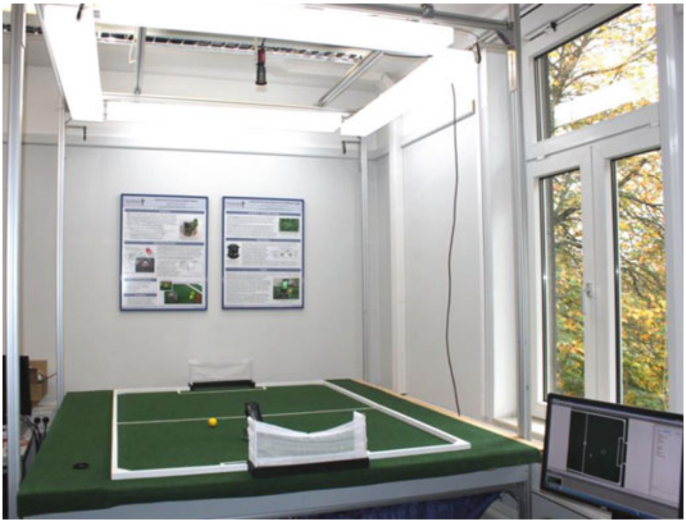


Fig. 3 Tele-workbench platform with video recording option (camera located over the field, live video displayed and recorded at PC, and robot data optionally included)

considering the fact that simulation simplifies the real situation and produces conclusive results [11].

Our tele-workbench is intended to enable onsite or remote users to set up, to execute, and to analyze experiments involving one or many AMiREsot robot platforms. In order to have as much as possible data from an experiment, support of data handling and processing has to be provided. The tele-workbench records the experiment using a camera mounted on top of the area. Pictures or videos of the camera (model AVT MANTA, 1″ CMOS, 2048 × 2048 pixel) are transmitted via Gigabit Ethernet to a video server for storing or processing the video data. The setup of the tele-workbench environment is shown in Fig. 3. The maximum size of the operational area is 2 m by 2 m. This area can be separated into smaller areas, or a robot soccer pitch can be placed on the top of area as depicted in Fig. 3. In parallel to the video transmission, sensor data and status of the robot during an experiment are transmitted to a data server. By adding time stamps, these data can be synchronized with the video data to enable detailed experiment analysis. The data storage concept is depicted in Fig. 4.

Depending on the required information of the robot, selected sensor data of the robot is transmitted to the data server. Examples are speed of the motors, data from the IR sensors, data from accelerometer or gyroscope, or internal state information of the robot extracted from the so-called software routine. To send the data, a wireless Bluetooth or Wi-Fi connection can be used; however, Bluetooth requires less hardware resources and it is simple to use because it runs in SPP mode that realizes a UART from the microcontroller’s point of view. One option to track the robots is to identify the robots in the video data. For this, a color patch is used on

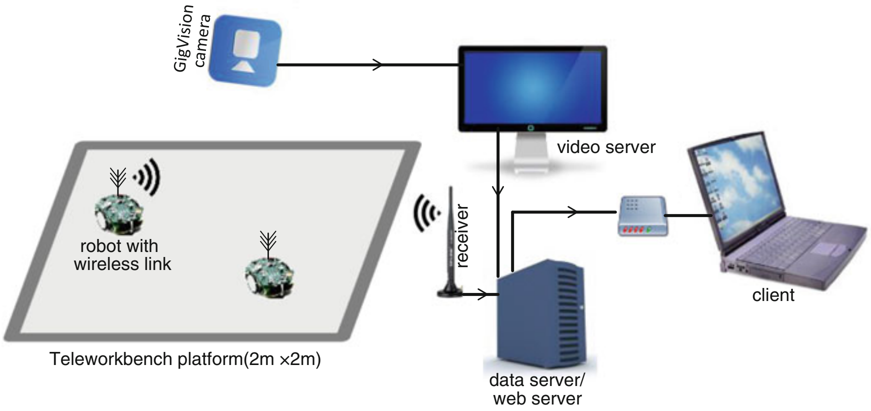


Fig. 4 Data flow of the tele-workbench environment including video recording and wireless data transmission from the robots for the purpose of sensor and status data capturing

top of the robots to mark them, or more structured pattern is used to be able to distinguish different robots and to extract the orientation of a robot. In more complex environments, the visual robot tracking is completed by a radio-based tracking as explained in the next section.

2.3 Robot Tracking Based on UWB Communication

A state-of-the-art option to track the robots during an experiment is to extract the robot position and orientation based on the video data. Particularly when the robots are marked with a specific pattern, this type of tracking can be performed at high accuracy. However, if the environment is more complex and robots are occluded, robots may not be visible by the camera. For these cases, a radio-based localization technique is used in our tele-workbench environment. The approach is to apply multiple time-of-flight measurement technique between communication modules. Multiple fixed modules are mounted in close vicinity of the tele-workbench platform. One module is mounted on the top of a robot to act as a module node. With known positions of the fixed nodes, the mobile node is able to calculate its position after performing a series of point-to-point time-of-flight distance measurements. The ranging principle is depicted in Fig. 5 left. To calculate the position of the robot on the two-dimensional tele-workbench surface, a minimum of two fixed nodes and two measurements is required. As an example, the determined distances r1 to r4 between the robot and two infrastructure nodes are shown in Fig. 5. These distances define the circles with the radius of r1 to r4. The intersection of the circles results in extracting the position of the robot. The more the distance measurements are performed, the more accurate the position of the robot can be calculated. As additional

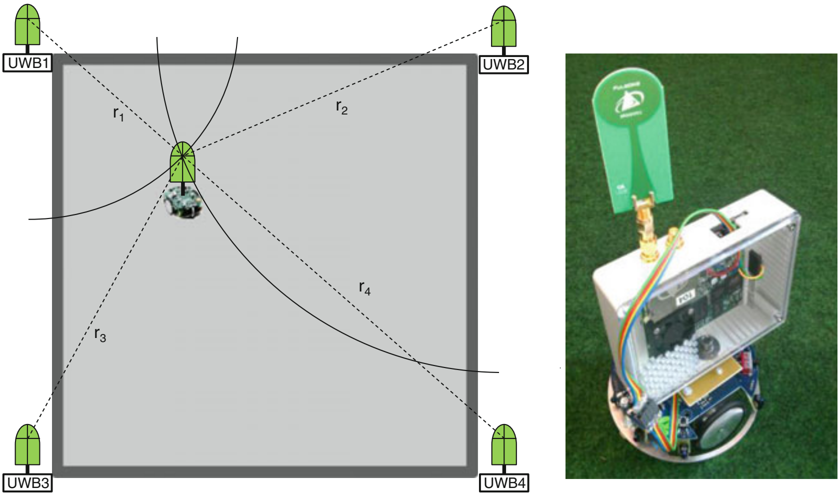


Fig. 5 Fixed reference nodes for position calculation are placed at the corners of the tele-workbench platform (*left*), UWB node from time domain, and its robot integration (*right*)

data processing, a filter technique can be applied to the measured distances and the derived robot position. Taking these calculations into account, an accuracy of ±4 cm can be achieved based on the UWB modules. By fusing these data with tics from the incremental encoder and applying odometry techniques, a higher accuracy can be expected, but this is part of the future work.

As depicted in Fig. 5, the robot is equipped with a fully functional UWB module. Therefore, the robot can perform time-of-flight measurements in the fixed UWB modules at the tele-workbench. This allows the robot to determine its position autonomously. The time-of-flight measurement with succeeding trigonometry results in x and y coordinates of the robot on the tele-workbench. With this basic approach, the orientation of the (static) robot cannot be determined. However, there are two reasonable options to determine the orientation of the robot using a single-antenna system. When the robot is moving, it has to calculate its positions consecutively. The difference of two positions defines a movement vector that also indicates the orientation of the robot. The second option is to place the UWB antenna not centered at the robot and to evaluate distances between the robot's antenna and the fixed antennas when the robot is turning. In future tests, the achievable precision of different options will be analyzed. In order to reduce the complexity, in the first tests, the differential vector method is used to determine the orientation.

3 Test and Validation Approach for Autonomous Robots/Applied Modeling of a Robot Platform

The first approach to develop a general methodology of testing and validation of the design of autonomous robots consists of a five-step iterative process. Starting with the modeling of the targeted AMS (Step A), test parameters can be derived based on the generated model (Step B). After that, the test bench can be described and configured (Step C). Furthermore, the data acquisition and processing must be implemented and executed (Step D). The last step is to apply the test and check whether the requirements were met (Step E). Based on the degree of fulfilling the requirements, actions to improve the design are derived (Step E). The steps A to E are visualized in Fig. 6.

Besides the shown hardware setup and first fundamental tests using the described methodology (Fig. 6), also a suitable approach to model the targeted robot platform is presented. The approach of modeling and handling of the AMS and the environmental complexity, which is characterized by the number and

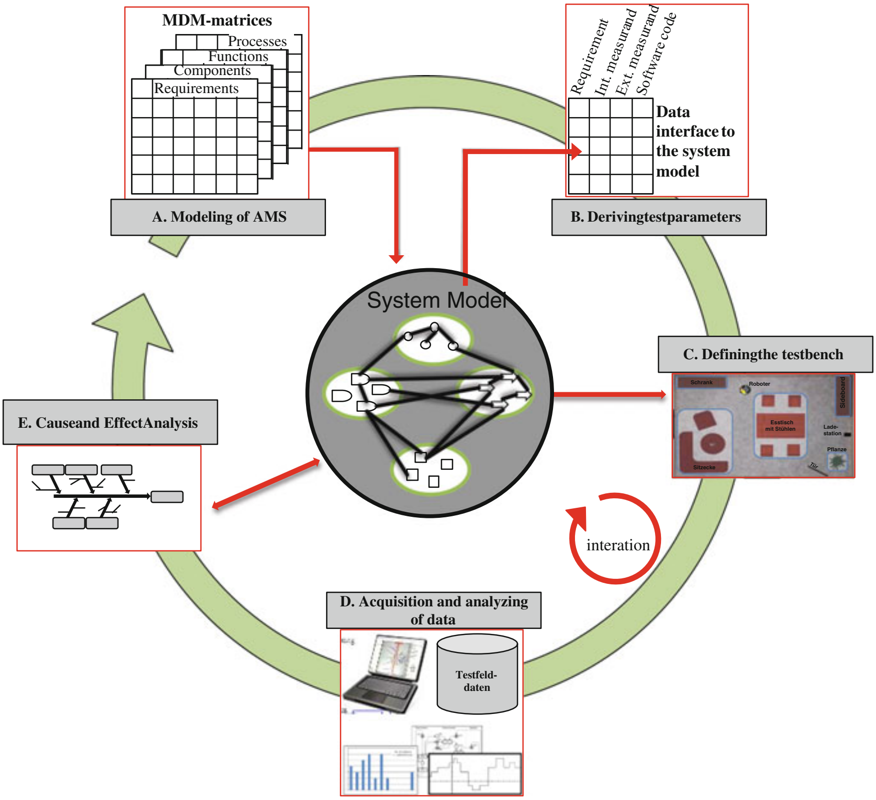


Fig. 6 Methodology to test and validate the design of AMS in an iterative process

diversity of elements of an AMS, is based on the systems engineering (SE) approach [12, 13]. SE is a general approach for problem solving which combines a procedure for problem solving with an interlinked system model [14, 15]. To establish such a model, different methodical approaches were postulated: the MDM (multiple domain matrix), the DSM (design structure matrix), the DMM (domain mapping matrix), or the DeCoDe method (demand compliant design) [16, 17]. To develop a unified system model of an AMS, the DeCoDe method was already used in several approaches [13, 18]. The DeCoDe method uses four limited views (Fig. 7):

- Requirements, for example stakeholder requirements for an AMS. These can be derived, for example, by use cases. A requirement for an AMS can be described by the example behavior “autonomous navigation.”
- Functions, which are solution neutral and realize the requirements, e.g., “change electric energy into kinetic energy.”
- Processes, which realize functions by using components as resources. For a control process, e.g., hardware devices (components) and software code are needed.
- Components of the AMS itself, e.g., motor and wheels.

The four views will be created by the information of relevant partial models such as systems and performance specifications (requirements), block diagrams (functions), flowchart (processes), and CAD drawings (components). However, software is a major issue for AMS. Therefore, the DeCoDe model was enhanced to model software in a unified way [17]. While the hardware is described as a component, the software is integrated into the DeCoDe model as an attribute of the processes. The entire software code is decomposed into process-specific parts. Therefore, the

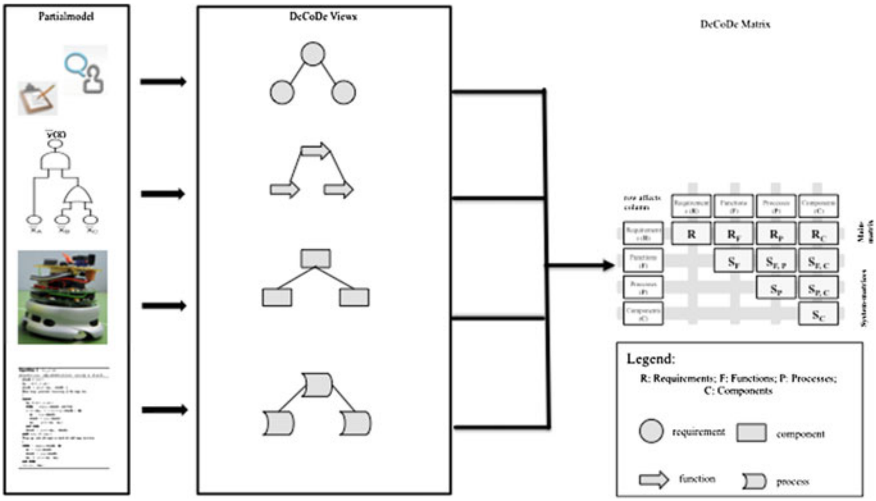


Fig. 7 DeCoDe model and matrices as a unified system model approach for AMS

software for each process is depicted as a process description. Therefore, the process describes a part of the desired behavior of the AMS. An observable failure can be directly assigned to the responsible software code. Inter alia, this is needed to allocate reasons for failures while testing.

4 Robot Experiments and System Performance

To simplify first tests, two UWB nodes are placed at the corners of the tele-workbench (see Fig. 5, modules UWB1 and UWB2). These two nodes allow the robot to determine its own position on the platform by using the UWB node that is fixed on the robot. By time-of-flight measurements, the UWB module mounted on the robot determines the distances to the UWB models placed at the corners of the platform. Robot's coordinate (*x*, *y*) is calculated using the trigonometric functions.

Figure 8 shows the recording and robot's behavior of one test. The left picture shows the initial situation with the desired path of the robot. The picture in the middle shows the orientation determination of the robot, and the right picture shows the resulting path and the end position of the robot. The right picture also depicts that the robot does not match the target exactly. This can be improved by taking more position measurements into account while driving toward the target direction. Figure 9 gives information about the speed of each motor of the robot during the experiment. These data have been additionally recorded when capturing the video. In the first 50 received pulses, the orientation phase can be seen. Both motors turn forward with equal speed and stop to perform the orientation measurement. Afterward, the right motor drives backward and the left motor forward in order to turn the robot and to align it to the destination point. When the robot faces the

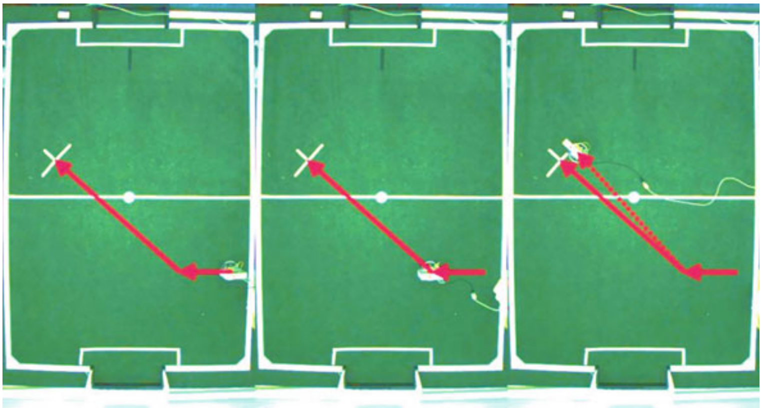


Fig. 8 Three snapshots of a video taken during a test scenario (sequence: start position, detection of robot's orientation, and turn and drive toward the destination)

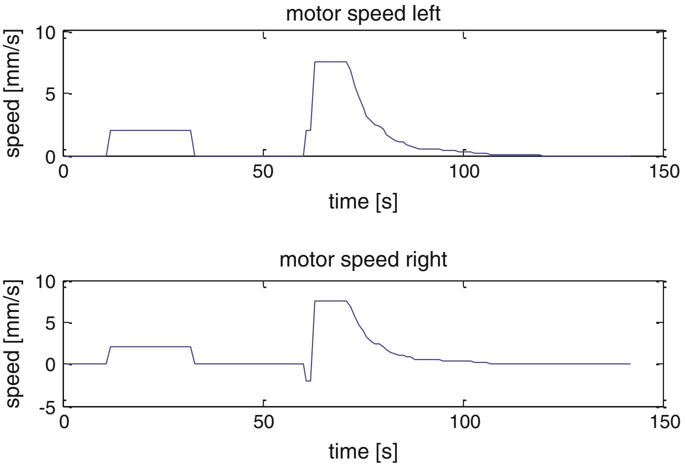


Fig. 9 Motor speed of *left* and *right* motors. Recorded sequence: orientation in the first 50 s, turn toward the target, and control speed in time segment between 50 and 100 s

target, it moves forward and reduces the speed when it is close to the target point to prevent an overshoot.

The applied tests demonstrate a basic functionality of the targeted robot platform. In most cases, the robot reaches an end position relatively close to the predefined target point. With the help of the tele-workbench, it is possible to analyze the movement and the internal data of the robot in detail in a synchronized way. Based on the test results, the main cause for not precisely reaching the target was identified: Small deviations in the rotation of the robot lead to an angle error, this results in a target vector with small error, and therefore, the robot finally missed the target. This was caused by the internal evaluation of the accelerometer data. The accelerometer data are needed to be integrated to estimate the current orientation of the platform. By decreasing the time step, it is possible to improve the estimation of the angle and could improve the robot performance. Another approach is to fuse data from the accelerometer and from the incremental encoder.

5 Conclusion

The design of complex mobile mechatronic systems requires an appropriate environment for device testing. Main objective of the test system is to validate the behavior of the system and to identify the problems if they occur. Our approach is to observe the robot's behavior based on video recording and to additionally save several status data during the experiment. By using a mapping algorithm followed by the DeCoDe approach for modeling the target platform, it is possible to identify the reason for a wrong behavior of a robot. It can be identified if the problem is based on, e.g., a specific sensor limitation (low resolution, wrong sample time) or on a piece of software that, for example, performs a wrong mapping of sensor data in response to motor actions. In order to develop a validation tool that can be used for the design process, a small-scale robot platform and an experiment environment have been used: the tele-workbench and the AMiREsot robot platform. As a first validation example, the navigation task of a mobile robot has been considered as it is necessary for robot vacuum cleaners. In future works, the tele-workbench will be extended to be able to record several data in parallel and to match the data with the video stream. An automated feature extraction should help to detect unwanted situations and to ease the mapping from the data space to the source of the problem that can be on hardware or software level.

References

1. Zheng, C., Bricogne, M., Le Duigou, J., Eynard, B.: Mechatronic design process: a survey of product data model. In: 24th CIRP Design Conference, vol. 21, pp. 282–287 (2014)
2. Casner, D., Renaud, J., Houssin, R., Knittel, D.: A novel design approach for mechatronic systems based on multidisciplinary design optimization; TRIZ Future 2014, Lausanne Suisse (2014)
3. Qamar, A., Wikander, J., During, C.: Designing Mechatronic Systems: A Model-Integration Approach, In: Culley, S.J.; Hicks, B.J.; McAloone, T.C.; Howard, T.J., Lindemann, U. (eds.) ICED11, 18TH International Conference on Engineering Design, pp. 145–156 (2011)
4. Heikkilä, T., Dalgaard, L., Koskinen, J.: Designing Autonomous Robot Systems—Evaluation of the R3-COP Decision Support System Approach. Matthieu ROY. SAFECOMP 2013—Workshop DECS (ERCIM/EWICS Workshop on Dependable Embedded and Cyberphysical Systems) of the 32nd International Conference on Computer Safety, Reliability and Security, Toulouse, France (2013)
5. Zhang, D., Gao, Z.: From component, structure to system: towards the design applicability of robots based on new methodology. In: IEEE 32nd Chinese Control Conference, Xian (2013)
6. Zhou, H., Sekane, S.: Localizing objects during robot SLAM in semi-dynamic environments. In: IEEE Proceedings of the 2008 International Conference on Advanced Intelligent Mechatronics, ASME (2008)
7. Steinbauer, G., Mühlenfeld, A., Koholka, R., Röhrer, R., Hartle, H., Wotawa, F.: Modulbasiertes Hardware-Design für autonome Mehrzweck-Roboter. Technische Universität Graz, Graz (2003)
8. Wongwirat, O., Hanidthikul, T., Vuthikulvanich, N.: A formal approach in robot development process using a UML model. In: 10th International Conference on Control, Automation, Robotics and Vision Hanoi, Vietnam (2008)
9. The AMiREsot Robot Soccer Tournament. http://www.fira.net/contents/data/AmireSotDescription.htm. Accessed June 2015
10. Tetzlaff, T., Zandian, R., Drüppel, L., Witkowski, U.: Smartphone controlled robot platform for robot soccer and edutainment. In: Proceedings of 3rd International Conference on Robot Intelligence Technology and Applications (RiTA 2014), China (2014)
11. Mondada, F., Franzi, E., Ienne, P.: Mobile robot miniaturization: A tool for investigation in control algorithms. In: Proceedings of the Third International Symposium on Experimental Robotics, pp 501-513. Tokyo, Japan (1994)
12. Meyer, C.M.: Integration des Komplexitätsmanagements in den strategischen Führungsprozess der Logistik. Haupt (2007)

13. Mamrot, M., Marchlewitz, S., Nicklas, J.-P., Winzer, P.: Using systems engineering for a requirement-based design support for autonomous robots. In: IEEE International Conference on Systems, Man, and Cybernetics, pp. 3146–3151. San Diego, CA, USA, 5–8 Oct 2014
14. Sage, A.P., Rouse, W.B.: Handbook of Systems Engineering and Management. Wiley, Hoboken (2009)
15. Winzer, P.: Generic Systems Engineering. Ein methodischer Ansatz zur Komplexitätsbewältigung. Springer-Vieweg Verlag, Berlin, Heidelberg (2013)
16. Kreimeyer, M., Lindemann, U.: Complexity Metrics in Engineering Design. Managing the Structure of Design Processes. Springer Verlag, Heidelberg (2011)
17. Sitte, J., Winzer, P.: Demand compliant design. IEEE Trans. Syst. Man Cybern.-Part A: Syst. Humans **41**(3) (2011)
18. Mamrot, M., Marchlewitz, S., Nicklas, J.-P., Winzer, P-. Tetzlaff, T., Kemper, P., Witkowski, U.: Model based test and validation support for autonomous mechatronic systems. In: IEEE System, Man and Cybernetics, Accepted paper for publication (2015)

Soft Robotics Technology and a Soft Table for Industrial Applications

Zhicong Deng, Martin Stommel and Weiliang Xu

Abstract Soft robotics is an expanding new research field. This article presents a state-of-the-art review on soft robotics including its research directions, key characteristics, materials, design and fabrication techniques. Although many biomimetic soft robots have been developed, a few of these have industrial applications. This article proposes a soft XY machine table for the purpose of object manipulation. The proposed table combines the concepts from three areas: *soft robotics*, *object manipulation* and *industrial application*. The surface of proposed table is entirely soft and embedded with inflatable air chambers. Surface deformation is generated by inflating these chambers. One object manipulation approach is to generate travelling waves on the deformable surface.

Keywords Soft robotics · Soft table · Object manipulation

1 Introduction

The traditional robots or ‘hard’ robots are well known for their abilities to perform repetitive tasks with both high speed and precision in a well-defined environment [1]. The word ‘hard’ reflects on the physical structures and components that construct the robot. Soft robotics is a relatively new research direction in the robotics research field. These soft robots generally have no hard skeleton structures. Their soft bodies

Z. Deng (✉) · W. Xu
Department of Mechanical Engineering, The University of Auckland,
Auckland, New Zealand
e-mail: zden012@aucklanduni.ac.nz

W. Xu
e-mail: p.xu@auckland.ac.nz

M. Stommel
School of Engineering, Computer and Mathematical Sciences,
Auckland University of Technology,
Auckland, New Zealand
e-mail: mstommel@aut.ac.nz

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_32

are deformable allowing them to change their body shape with unlimited degree of freedom (DOF). This flexibility allows them to adapt to complex environments without using an explicit 3D-model of the scene. The compliant nature of the soft body enables these robots to produce smooth and natural motions comparable to human and animal movements. Collisions with the soft body are harmless to humans and the environment.

This article comprises two sections. The first section is a state-of-the-art review on soft robotics. The current research direction, key characteristics, material, design and fabrication techniques are discussed in detail. The second section of this article proposes a novel soft XY machine table for the purpose of object manipulation.

2 State-of-the-Art Review on Soft Robotics

2.1 Biological Inspirations

Many soft robots are inspired by soft animals and soft body parts found in nature [2–10]. These include elephant trunk, caterpillar, earthworm, octopus and star fish. Soft animals typically have no skeleton to support their own body. As such, these animals are commonly found in the ocean or underground where the surrounding environment can support their body weight [3]. The softness of their body provides them certain survival advantages. Their flexible bodies allow them to quickly adapt to the surrounding environment such as avoiding predators by moving into tight spaces. Soft robots exploit these characteristics of soft animals to gain advantages over traditional hard robots.

Biomimetics is one major research direction for soft robotics. The objective of developing these biomimetic robots is to reproduce the motions of soft animals which in turn deepen the understandings of such animals [11]. These concepts inspired from soft animals are transferred to robotic applications. One example is the GoQBOT that mimics the caterpillar's rolling escape motion [12]. Another example is the soft robot arm inspired by the octopus [13]. Marchese, Onal and Rus developed a soft robotic fish capable of rapid escape as seen in real fish [5]. Another area of research interest for soft robotics is their applications in medical studies. The human swallowing motion was emulated by a soft swallowing robot [14, 15]. Menguc et al. had developed a soft wearable suit for lower limb which was capable of sensing motion and collecting biomechanics measurements [16]. A Braille display device was developed using soft actuators [17].

2.2 Key Characteristics

The appealing characteristic of soft robots is their ability to change their body shape. The elastic material used for the body enables multiple DOF movements such as morphing and rolling [18]. This high body flexibility inherited from the material allows locomotive robots to navigate through unknown environments by adapting their motion without making detailed 3D measurements. Deformation is possible in any direction on the soft body and thus creating countless DOF. However, these infinite possibilities make it difficult to define mechanical states for the robot.

The softness of the body introduces compliance into the system, enabling the ability to reproduce animal-like motion. This compliant nature adds a sense of 'self-correction' into the system which enables environmental adaptation that hard robots do not have [19]. This body softness also reduces the damage caused by collision with the human body which enables safe human–robot interaction [20]. The soft body possesses resistance against damage in certain aspects that hard robots do not have [21]. One example is the multigait robot which has a resistance to high-force and low-strain damage such as falling rocks [22]. Soft robots generally have low resistance to sharp cutting damage.

Nonlinearity in the system is another important characteristic of soft robots. The mechanical properties of the soft material change with deformation which leads to non-uniform properties across the body. This is a major challenge for modelling the physical system. Trimmer had attempted to use this nonlinearity as an advantage to generate adaptive and robust control for the soft robot [19]. Generally, soft robots are lightweighted due to the absence of the hard skeleton and supporting structures seen in hard robots. This lightweighted characteristic suggests a lower energy consumption for soft locomotive robots compare to heavier hard robots. The movements of soft robots are potentially more energy efficient. However, the lack of hard structures makes it difficult to transmit actuation through the soft body [23]. There are no standard components to be selected and purchased for soft robots. Every soft robot is custom-built for its own application. The lack of standardization leads to design difficulty as no guideline exists for an effective design. The cost of the robot depends on the soft material and fabrication method chosen.

Certain soft robots have special features that distinct themselves from the others. Morin et al. had developed a soft robot that was capable of changing its body colour and surface temperature [24]. Colour fluids flowed into the micro-fluidic channels embedded in the soft body to apply camouflage. Another example is a hybrid of hard and soft robots which was developed to combine the attractive features of both sides [25]. This hybrid had the fast locomotion with wheels of hard robots and the soft gripping capability of soft robots.

2.3 Soft Materials and Soft Actuation

There are various potential materials for soft robots to achieve the desired softness and compliance. These materials can be classified into two categories: soft materials used for the robot body and soft materials used as soft actuators. Silicone elastomer or silicone rubber is the most common soft material used for the body of soft robots [7, 8, 12, 15, 16, 19, 26–28]. This material can be fabricated into different shapes and structures with mould and casting processes. Its elastic characteristic enables large storage of potential energy. This characteristic was exploited by the soft robot powered by explosions [29]. Electrical sparks were used to trigger a methane combustion inside the soft body which produced a rapid body expansion. The soft body was not damaged by this internal explosion. A composite elastomer material was developed to achieve lower density with higher strength [30]. The robot constructed from this material had larger weight support capability and higher resilience to harsh environments.

The use of soft actuators is a viable solution to the problem of force transmission through the soft body. These actuators can be embedded into the soft body to provide actuation or even become the body itself. Smooth and natural motion can be generated with actuator's compliant nature. Shape memory alloy (SMA) is a popular smart material used in soft robotics applications. It is capable of producing high mass-specific force with low strain [3, 31]. The drawback is the nonlinearity induced by the change of electrical resistance with temperature change [12]. The common actuation method with SMA is in the form of coils or springs [2, 4, 7, 10, 13, 18–20, 32, 33]. The coils contract when triggered by an electrical current and generate pulling forces towards the centre at both ends of the coil. A peristaltic motion can be generated through phased contractions and relaxations of SMA springs [14].

Electro-active polymer (EAP), also referred to as artificial muscle is a major category of smart material that can be used for both the soft body and actuator. It can combine the body structure, soft actuation and sensory feedback in one material [1]. The material has high flexibility and damage tolerance [31]. Motion is generated in response to electric field. EAP can be classified into two types: the ionic and non-ionic types [34]. The non-ionic type also referred to as the dielectric type produces a larger strain and has a faster response time compared to the iconic type at the cost of high actuation voltage [1]. Dielectric elastomer is a popular non-ionic EAP. This material has a low density and good compliance [35]. It is capable of producing high mass-specific power in silent operation but requires high voltage and a rigid frame [3]. The material is generally configured as membrane layers to be used as soft actuator [36, 37]. Multiple layers can be stacked together to enhance the torque, power and mass [38–40]. The self-sensing ability enables the material to be used as deflection sensor [41]. Ionic polymer metal composite (IPMC) is another EAP material that can be used as soft actuator [32]. The material bends with low voltage [42]. A worm-like robot was developed using IPMC as the soft body material [43].

The use of inflatable structures with a pneumatic system is another approach to achieve soft actuation. This combination is capable of generating a relatively high force and large displacement with the drawbacks of high energy consumption and a complex pneumatic system [3]. Other advantages include low operating temperature and high compliance. Numerous soft actuators developed with this approach achieved large bending angles [44, 45]. Pneumatic artificial muscle (PAM) uses inflatable chambers with a pneumatic system. The muscle cell contracts in length with inflation. The high compliance of this soft actuator makes it favourable for applications involving human interaction [46]. PAM was also used as the body material for a soft robot [47, 48]. Embedding inflatable chambers into the soft body can also achieve soft actuation. The inspiration comes from plant cells which use its internal pressure to control its structure properties and shape [1, 49]. Soft robots embedded with inflatable chambers are able to directly deform their body [15, 22]. Martinez et al. had developed robotic tentacles that were capable of generating 3D motion [28]. The tentacles were controlled by micro-channels embedded in the soft body.

2.4 Sensory Feedback

Sensory feedback is a difficult task for soft robotics as the deformation can occur in any direction. The challenge is to decide which parameters to measure and the measurement method. Only relevant and accurate information should be collected to provide useful feedbacks. Visual feedback is one potential sensing approach for soft robots. The resultant motion can be monitored on an external image or video capture device. Deformable tactile sensor is a promising solution to achieve sensory feedback. This is biologically inspired by soft animals which have sensing elements embedded in their skin [50]. These deformable sensors can be attached to the soft surface or embedded into the soft body and directly measures the deformation. Artificial skin is an example of deformable sensor. Park, Chen and Wood had developed a soft artificial skin which consisted of micro-channels that were filled with EGaIN, an alloy of gallium and indium in liquid state that changes its electrical resistance with deformation [51].

2.5 Design, Modelling and Fabrication

The traditional approach to analyse a soft body is to approximate it as a rigid body and apply rigid body principles. Finite element analysis (FEA) is one approach to model the deformation of soft structures [7, 8, 27]. FEA is useful to guide the design and optimization of deformable structures. The computational time required for each deformation simulation increases with the structural complexity of the soft robot. Another approach is the lumped dynamics modelling technique which decompose the soft body into smaller rigid bodies that are linked together by linear and torsional springs and dampers [52]. This is a less accurate model compared to FEA but computationally more efficient. These approaches are not ideal for soft body analysis as certain assumptions in the rigid body principles may not apply to soft materials. A dynamic simulation framework for soft material was developed by discretizing an object into small interconnected voxels [53]. The framework used the VoxCAD software to provide visual appearance for the simulation. Another toolkit was developed as an open source platform containing numerous tools to assist the design and simulation of soft robots [54]. Evolutionary algorithm is another approach to optimize the design of soft structures if the design goals can be specified [55].

Nonlinearity is a key characteristic to consider for theoretical modelling. Deng, Wang, Wakimoto and Kawashima researched techniques based on statistical learning theory to achieve nonlinear modelling [45]. The use of artificial neural networks is a promising approach to model nonlinearity. Inputs and output data can be obtained from experiments and feed into the network to approximate the relationship function. Predictions of given situations can be obtained once the network is trained. Various researches had demonstrated the capabilities of echo state networks to control the soft robotic arm inspired by octopus [56, 57].

Morphological computation is a design and control concept which aims to outsource portion of the control from the central controller to the body or the mechanism itself [58–60]. Simple actuations can generate complex motions which are partly controlled by the morphology in the mechanism [61]. Iida and Pfeifer demonstrated an example robot of self-stabilization without sensory feedback [62]. This concept is suitable for underactuated system which is seen in numerous soft robots. The concept is also related to reservoir computing which considers the soft body as a reservoir for computational resources [63]. The nonlinearity and elastic body dynamics of soft robots are not seen as drawbacks, but rather as a potential computational resource to be exploited. Information theoretical analysis was one effective approach to characterize the soft system and could be used to guide the development of the controller for soft robots [64].

As no standard components exist for soft robots, fabrication methods vary between different soft robots and can occasionally be expensive [65]. Unlike hard robots that can be separated into modules and manufactured independently, the soft body needs to be fabricated as one module for effective force transmission. At present, the most commonly used fabrication method is mould and casting. Soft lithography is the fabrication method used for the Multigait robot [22]. This method is an advanced moulding technique for micro- and nanofabrication of elastomeric polymers [66]. It is able to create complex structure and patterns with high precision.

3 Proposal of a Soft XY Machine Table

3.1 Research Motivation and Existing Researches

Although the main research direction of soft robotics is biomimetic robots, we believe that soft robots have unique characteristics that are highly attractive for industrial applications: they are lightweight, suitable for human–robot interaction, and they prevent damaging forces on the objects being handled. In the following, we will discuss a soft XY machine table that demonstrates the potential of soft robotics for industrial applications. To this end, we combine concepts from three areas: *object manipulation*, *soft robotics* and *industrial application*. The proposed table is capable of object manipulation on its surface.

Several studies have been done on the topic of object manipulation on a surface. The WaveHandling system developed by Festo Company created waves on a surface to transport objects [67]. The soft surface layer was actuated by multiple small linear actuators underneath. The wave produced was more towards discrete waves rather than a smooth continuous travelling wave. Another example of object manipulation on a surface is a soft machine controlled by a Braille display [68]. The machine demonstrated the movement of a round object from one position to another with the use of inflatable cells on the surface. The resultant motion was more of a state transition rather than a continuous motion. Shape display devices can also be used for object manipulation on a surface. The inFORM system was able to manipulate objects by changing the geometry of its surface which was constructed with many actuating pins [69, 70].

Previous work had been done to develop a robotic system for ovine offal sorting [71]. The soft peristaltic table in the system was devised to generate peristaltic motions to move organs to a state which was easier for organ separation. A probabilistic automaton modelling approach was developed to aid the control of the soft table [72].

3.2 Soft XY Machine Table

We propose a soft XY machine table for the purpose of object manipulation. Objects placed on top of the proposed table can be manipulated on the XY plane. This manipulation includes position control and orientation control on the XY plane. From the actuation methods discussed in the previous section, the use of inflatable soft structure with a pneumatic system is chosen. This combination is able to generate high force and large displacement which is ideal to support objects and create deformations of reasonable size. The surface layer of the proposed table is made of a single piece of soft material embedded with multiple inflatable air chambers. Surface deformation can be directly generated by pressurizing these air chambers as illustrated in Fig. 1. A potential application is a sorting table which can sort items for industrial production lines. The soft surface ensures safe human interaction which is important in industrial environments. This softness inherited from material also enables intrinsic soft handling of objects. This is ideal for handling delicate objects such as vegetables, fruits, animal organs and integrated chips. The handling damage is minimized without the use of specific control algorithms. It is also possible to use the deformable surface of the table as a 3D shape display. 3D landscape map or even sea wave simulation can be displayed.

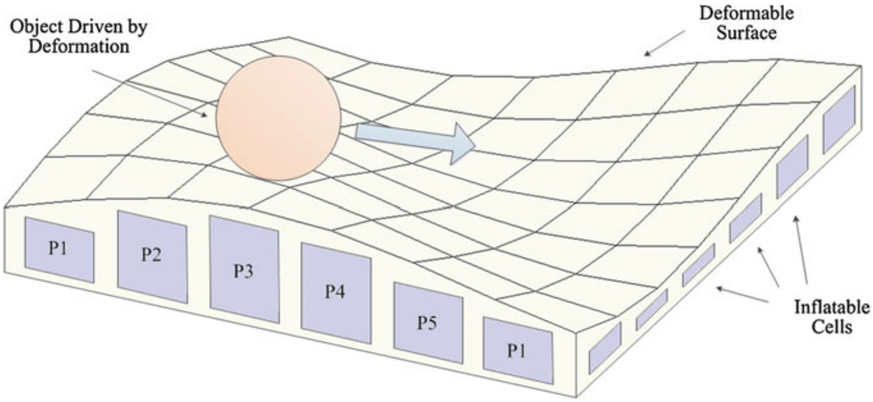


Fig. 1 A conceptualization of the proposed soft table

One object manipulation approach is to generate travelling waves on the deformable surface to provide a surface profile which initiates object movement. A travelling wave can be generated by inflating the embedded air chambers with a phase difference. As the travelling wave approaches the object, it raises the object to an unstable state of higher potential energy and provides a slope profile. This slope profile forces the object to an equilibrium position of lower potential energy. As the travelling wave moves across the surface, the equilibrium position for the object shifts along and thus achieving object manipulation. This is a passive actuation approach as the main driving force is gravity. Low surface friction is required for an efficient transportation. This approach is different to previous work which used peristalsis actuation and shaking as the driving forces for object movement. No shaking is used in the current design. As the proposed table is capable of deformation of numerous DOF, the object manipulation approach is not limited to the travelling wave method.

Multiple objects can be manipulated simultaneously as different travelling waves can be generated on different regions of the soft table. Provided the table size is large enough, the travelling waves can be completely independent of each other and the object manipulations are isolated to its own region. Smooth and continuous waves are desired to aid the object movement and this can be achieved with a high correlation between air chambers. A strong correlation between two chambers suggests the inflation of one air chamber will affect its neighbouring air chambers. This correlation can be in the form of chamber structure, air flow or correlation provided by the material. Different chamber designs can be evaluated to achieve an optimal performance for object manipulation. The design of the deformable surface is relatively simple and a mould and casting process can be applied to fabricate the soft surface.

Theoretical models are required to understand and operate the proposed soft table. These models aim to model the physical system and guide the prototype design. The output of the modelled system can be predicted when given input or conversely; the

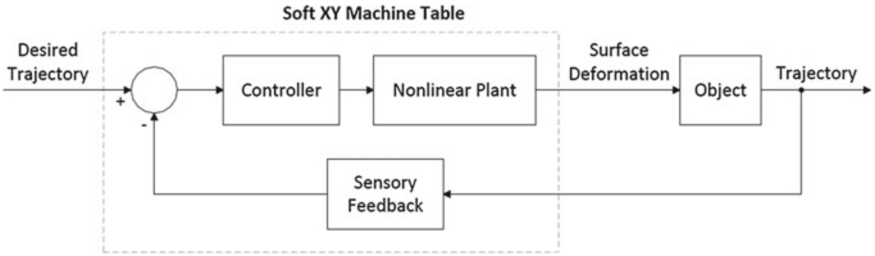


Fig. 2 Proposed control flow diagram for object manipulation

required input can be calculated when given the desired output. Theoretical modelling can be separated into two components for the case of the proposed table: object response and deformation generation. The object response component models the object dynamics in response to surface deformation. The object movement can be predicted when input deformation is given. The deformation generation component models the nonlinear plant and is used to predict the resultant surface deformation when given system inputs such as chamber pressures. There are various challenges in modelling the proposed soft table. Nonlinearity in the system is a major barrier in establishing a model for deformation generation. Both the soft material properties and the air chamber structure affect the surface deformation, making it difficult to establish a one-to-one relationship between the input and output of the system.

The general control flow of the object manipulation process is as follows. The user enters the desired object trajectory through a user interface. This command is processed by the controller of the table, and the required deformation is generated on the table surface. The object moves in response to the deformation and thus object manipulation is achieved. This flow can be expressed in the form of a closed-loop control system by introducing sensory feedback as illustrated in Fig. 2. The controller determines the deformation required based on the deformation generation model. Visual feedback can provide information regarding the object movement for high-level planning. Tactile sensors can be attached to the deformable surface to provide deformation information for low-level actuation control. The sensory system can be further developed to provide functions such as object detection and recognition. Object information such as shape, size and weight can be calculated based on deformation feedback. The surface deformation generated can then be adjust for different object sizes and shapes to improve the efficiency of object transportation.

4 Conclusion

Soft robots have many appealing characteristics compared to traditional hard robots. This article discussed the current research direction, key characteristics, materials, design and fabrication techniques of soft robotics. Design and modelling difficulties regarding soft robots were also presented. This article proposes a novel soft XY machine table for the purpose object manipulation. The soft table demonstrates the potential of soft robotics in industrial applications. The proposed table provides safe human interaction and soft handling of objects. One object manipulation method is to create travelling waves on the deformable surface. Visual and tactile sensory feedback can be included to the soft table to form a closed-loop control system. Theoretical models for object response and deformation generation will be developed in the future. A prototype of the table will be built to validate the manipulation concept.

References

1. Trivedi, D., Rahn, C.D., Kier, W.M., Walker, I.D.: Soft robotics: biological inspiration, state of the art, and future research. Appl. Bionics Biomech. **5**, 99–117 (2008)
2. Gao, F., Wang, Z., Wang, Y., Wang, Y., Li, J.: A prototype of a biomimetic mantle jet propeller inspired by cuttlefish actuated by SMA wires and a theoretical model for its jet thrust. J. Bionic Eng. **11**, 412–422 (2014)
3. Kim, S., Laschi, C., Trimmer, B.A.: Soft robotics: a bioinspired evolution in robotics. Trends Biotechnol. **31**, 287–294 (2013)
4. Mao, S., Dong, E., Jin, H., Xu, M., Zhang, S., Yang, J., Low, K.H.: Gait study and pattern generation of a starfish-like soft robot with flexible rays actuated by SMAs. J. Bionic Eng. **11**, 400–411 (2014)
5. Marchese, A.D., Onal, C.D., Rus, D.: Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators. Soft Robot. **1**, 75–87 (2014)
6. Margheri, L., Laschi, C., Mazzolai, B.: Soft robotic arm inspired by the octopus: I. from biological functions to artificial requirements. Bioinspiration Biomimetics **7** (2012)
7. Menciassi, A., Gorini, S., Pernorio, G., Dario, P.: A SMA actuated artificial earthworm. In: Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA), vol. 4, pp. 3282–3287. IEEE (2004)
8. Suzumori, K., Endo, S., Kanda, T., Kato, N., Suzuki, H.: A bending pneumatic rubber actuator realizing soft-bodied manta swimming robot. In: 2007 IEEE International Conference on Robotics and Automation, pp. 4975–4980 (2007)
9. Tsukagoshi, H., Kitagawa, A., Segawa, M.: Active hose: an artificial elephant's nose with maneuverability for rescue operation. In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, vol. 3, pp. 2454–2459 (2001)
10. Umedachi, T., Trimmer, B.A.: Design of a 3D-printed soft robot with posture and steering control. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 2874–2879. IEEE (2014)
11. Pfeifer, R., Lungarella, M., Iida, F.: The challenges ahead for bio-inspired 'soft' robotics. Commun. ACM **55**, 76–87 (2012)
12. Lin, H.T., Leisk, G.G., Trimmer, B.A.: GoQBOT: A caterpillar-inspired soft-bodied rolling robot. Bioinspiration Biomimetics **6** (2011)
13. Laschi, C., Cianchetti, M., Mazzolai, B., Margheri, L., Follador, M., Dario, P.: Soft robot arm inspired by the octopus. Adv. Robot. **26**, 709–727 (2012)
14. Miki, H., Okuyama, T., Kodaira, S., Luo, Y., Takagi, T., Yambe, T., Sato, T.: Artificial-esophagus with peristaltic motion using shape memory alloy. Int. J. Appl. Electromagnet. Mech. **33**, 705–711 (2010)
15. Chen, F.J., Dirven, S., Xu, W.L., Li, X.N.: Soft actuator mimicking human esophageal peristalsis for a swallowing robot. IEEE/ASME Trans. Mechatron. **19**, 1300–1308 (2014)

16. Menguc, Y., Park, Y.L., Martinez-Villalpando, E., Aubin, P., Zisook, M., Stirling, L., Wood, R.J., Walsh, C.J.: Soft wearable motion sensing suit for lower limb biomechanics measurements. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 5309–5316 (2013)
17. Lee, S., Jung, K., Koo, J., Lee, S., Choi, H., Jeon, J., Nam, J., Choi,H.: Braille display device using soft actuator. Smart Struct. Mater. (2014)
18. Trimmer, B.A., Lin, H.T., Baryshyan, A., Leisk, G.G., Kaplan, D.L.: Towards a biomorphic soft robot: Design constraints and solutions. In: Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics, pp. 599–605 (2012)
19. Trimmer, B.A.: New challenges in biorobotics: incorporating soft tissue into control systems. Appl. Bionics Biomech. **5**, 119–126 (2008)
20. Sugiyama, Y., Hirai, S.: Crawling and jumping of deformable soft robot. In: Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 4, pp. 3276–3281 (2004)
21. Martinez, R.V., Glavan, A.C., Keplinger, C., Oyetibo, A.I., Whitesides, G.M.: Soft actuators and robots that are resistant to mechanical damage. Adv. Funct. Mater. **24**, 3003–3010 (2014)
22. Shepherd, R.F., Ilievski, F., Choi, W., Morin, S.A., Stokes, A.A., Mazzeo, A.D., Chen, X., Wang, M., Whitesides, G.M.: Multigait soft robot. Proc. Nat. Acad. Sci. **108**(51), 20400–20403 (2011)
23. Steltz, E., Mozeika, A., Rembisz, J., Corson, N., Jaeger, H.M.: Jamming as an enabling technology for soft robotics. In: Proceedings of the SPIE: Electroactive Polymer Actuators and Devices, pp. 764225–764229 (2010)
24. Morin, S.A., Shepherd, R.F., Kwok, S.W., Stokes, A.A., Nemiroski, A., Whitesides, G.M.: Camouflage and display for soft machines. Science **337**, 828–832 (2012)
25. Stokes, A.A., Shepherd, R.F., Morin, S.A., Ilievski, F., Whitesides, G.M.: A hybrid combining hard and soft robots. Soft Robot. **1**, 70–74 (2014)
26. Steltz, E., Mozeika, A., Rodenberg, N., Brown, E., Jaeger, H.M.: JSEL: Jamming skin enabled locomotion. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5672–5677 (2009)
27. Ilievski, F., Mazzeo, A.D., Shepherd, R.F., Chen, X., Whitesides, G.M.: Soft robotics for chemists. Angewandte Chemie **123**, 1930–1935 (2011)
28. Martinez, R.V., Branch, J.L., Fish, C.R., Jin, L., Shepherd, R.F., Nunes, R., Suo, Z., Whitesides, G.M.: Robotic tentacles with three-dimensional mobility based on flexible elastomers. Adv. Mater. **25**, 205–212 (2013)
29. Shepherd, R.F., Stokes, A.A., Freake, J., Barber, J., Snyder, P.W., Mazzeo, A.D., Cademartiri, L., Morin, S.A., Whitesides, G.M.: Using explosions to power a soft robot. Angewandte Chemie—Int. Ed. **52**, 2892–2896 (2013)
30. Tolley, M.T., Shepherd, R.F., Mosadegh, B., Galloway, K.C., Wehner, M., Karpelson, M., Wood, R.J., Whitesides, G.M.: A resilient, untethered soft robot. Soft Robot. **1**, 213–223 (2014)
31. Mattar, E.: A survey of bio-inspired robotics hands implementation: new directions in dexterous manipulation. Robot. Auton. Syst. **61**, 517–544 (2013)
32. Guo, S., Shi, L., Ye, X., Li, L.: A new jellyfish type of underwater microrobot. In: International Conference on Mechatronics and Automation (ICMA), pp. 509–514 (2007)
33. Lee, N., Lee, Y.H., Chung, J., Heo, H., Yang, H., Lee, K.S., Ryu, H., Jang, S., Lee, W.: Shape-changing robot for stroke rehabilitation. In: Proceedings of the 2014 conference on Designing interactive systems, pp. 325–334. ACM (2014)
34. Choi, H.R., Jung, K., Ryew, S., Nam, J.D., Jeon, J., Koo, J.C., Tanie, K.: Biomimetic soft actuator: design, modeling, control, and applications. IEEE/ASME Trans. Mechatron. **10**, 581–593 (2005)
35. Anderson, I.A., Gisby, T.A., McKay, T.G., O'Brien, B.M., Calius, E.P.: Multi-functional dielectric elastomer artificial muscles for soft and smart machines. J. Appl. Phys. **112**, 041101 (2012)
36. Anderson, I.A., Hale, T., Gisby, T., Inamura, T., McKay, T., O'Brien, B., Walbran, S., Calius, E.P.: A thin membrane artificial muscle rotary motor. Appl. Phys. A: Mater. Sci. Process. **98**, 75–83 (2010)

37. Poole, A.D., Booker, J.D., Wishart, C.L., McNeill, N., Mellor, P.H.: Performance of a prototype traveling-wave actuator made from a dielectric elastomer. IEEE/ASME Trans. Mechatron. **17**, 525–533 (2012)
38. Carpi, F., Salaris, C., De Rossi, D.: Folded dielectric elastomer actuators. Smart Mater. Struct. **16**, S300–S305 (2007)
39. Lotz, P., Matysek, M., Schlaak, H.F.: Fabrication and application of miniaturized dielectric elastomer stack actuators. IEEE/ASME Trans. Mechatron. **16**, 58–66 (2011)
40. Nguyen, C.T., Phung, H., Nguyen, T.D., Lee, C., Kim, U., Lee, D., Moon, H., Koo, J., Choi, H.R.: A small biomimetic quadruped robot driven by multistacked dielectric elastomer actuators. Smart Mater. Struct. **23**, 065005 (2014)
41. Brochu, P., Pei, Q.: Advances in dielectric elastomers for actuators and artificial muscles. Macromol. Rapid Commun. **31**, 10–36 (2010)
42. Popa, I.: A multi-link kinematics model for microrobots with artificial muscle structure. In: ARA Annual Congress Proceedings, pp. 143–149 (2014)
43. Arena, P., Bonomo, C., Fortuna, L., Frasca, M., Graziani, S.: Design and control of an IPMC wormlike robot. IEEE Trans. Syst. Man Cybern. Part B: Cybern. **36**, 1044–1052 (2006)
44. Shapiro, Y., Wolf, A., Gabor, K.: Bi-bellows: pneumatic bending actuator. Sens. Actuators, A: Phys **167**, 484–494 (2011)
45. Deng, M., Wang, A., Wakimoto, S., Kawashima, T.: Characteristic analysis and modeling of a miniature pneumatic curling rubber actuator. In: 2011 International Conference on Advanced Mechatronic Systems (ICAMechS), pp. 534–539 (2011)
46. Nakamura, T., Shinohara, H.: Position and force control based on mathematical models of pneumatic artificial muscles reinforced by straight glass fibers. In: 2007 IEEE International Conference on Robotics and Automation, pp. 4361–4366 (2007)
47. Mangan, E.V., Kingsley, D.A., Quinn, R.D., Sutton, G.P., Mansour, J.M., Chiel, H.J.: A biologically inspired gripping device. Ind. Robot **32**, 49–54 (2005)
48. Hirayama, Y., Suzuki, K., Nakamura, T.: Development of a peristaltic pump based on bowel peristalsis verification of the basic characteristic considered change of motion patterns. In: 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 748–753 (2011)
49. Pagitz, M., Pagitz, M., Hühne, C.: A modular approach to adaptive structures. Bioinspiration Biomimetics **9**, 046005 (2014)
50. Boxerbaum, A.S., Chiel, H.J., Quinn, R.D.: A new theory and methods for creating peristaltic motion in a robotic platform. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 1221–1227 (2010)
51. Park, Y.L., Chen, B.R., Wood, R.J.: Design and fabrication of soft artificial skin using embedded microchannels and liquid conductors. IEEE Sens. J. **12**, 2711–2718 (2012)
52. Rieffel, J., Saunders, F., Nadimpalli, S., Zhou, H., Hassoun, S., Rife, J., Trimmer, B.: Evolving soft robotic locomotion in PhysX. In: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, pp. 2499–2504. ACM (2009)
53. Hiller, J., Lipson, H.: Dynamic simulation of soft multimaterial 3D-printed objects. Soft Robot. **1**, 88–101 (2014)
54. Holland, D.P., Park, E.J., Polygerinos, P., Bennett, G.J., Walsh, C.J.: The soft robotics toolkit: shared resources for research and design. Soft Robot. **1**, 224–230 (2014)
55. Hiller, J., Lipson, H.: Automatic design and manufacture of soft robots. IEEE Trans. Robot. **28**, 457–466 (2012)
56. Kuwabara, J., Nakajima, K., Kang, R., Branson, D.T., Guglielmino, E., Caldwell, D.G., Pfeifer, R.: Timing-based control via echo state network for soft robotic arm. In: The 2012 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2012)
57. Li, T., Nakajima, K., Pfeifer, R.: Online learning for behavior switching in a soft robotic arm. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 1296–1302 (2013)

58. Pfeifer, R., Iida, F., Gómez, G.: Morphological computation for adaptive behavior and cognition. In: International Congress Series, vol. 1291, pp. 22–29. Elsevier (2006)
59. Paul, C.: Morphological computation: a basis for the analysis of morphology and control requirements. Robot. Auton. Syst. **54**, 619–630 (2006)
60. Rieffel, J.A., Valero-Cuevas, F.J., Lipson, H.: Morphological communication: exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. J. R. Soc. Interface **7**, 613–621 (2010)
61. Pfeifer, R., Iida, F.: Morphological computation: connecting body, brain and environment. Jpn. Sci. Mon. **58**, 48–54 (2005)
62. Iida, F., Pfeifer, R.: Self-stabilization and behavioral diversity of embodied adaptive locomotion. In: Embodied Artificial Intelligence. Springer (2004)
63. Nakajima, K., Hauser, H., Kang, R., Guglielmino, E., Caldwell, D.G., Pfeifer, R.: A soft body as a reservoir: case studies in a dynamic model of octopus-inspired soft robotic arm. Front. Comput. Neurosci. **7** (2013)
64. Nakajima, K., Li, T., Sumioka, H., Cianchetti, M., Pfeifer, R.: Information theoretic analysis on a soft robotic arm inspired by the octopus. In: 2011 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 110–117 (2011)
65. Lipson, H.: Challenges and opportunities for design, simulation, and fabrication of soft robots. Soft Robot. **1**, 21–27 (2013)
66. Xia, Y., Whitesides, G.M.: Soft lithography. Ann. Rev. Mater. Sci. **28**, 153–184 (1998)
67. http://www.festo.com/cms/en_corp/13136.htm. Accessed 17 March 2014
68. Mosadegh, B., Mazzeo, A.D., Shepherd, R.F., Morin, S.A., Gupta, U., Sani, I.Z., Lai, D., Takayama, S., Whitesides, G.M.: Control of soft machines using actuators operated by a braille display. Lab Chip **14**, 189–199 (2014)
69. Follmer, S., Leithinger, D., Olwal, A., Hogge, A., Ishii, H.: inFORM: dynamic physical affordances and constraints through shape and object actuation. UIST (2013)
70. Leithinger, D., Follmer, S., Olwal, A., Ishii, H.: Physical telepresence: shape capture and display for embodied, computer-mediated remote collaboration. In: Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology. ACM (2014)
71. Stommel, M., Xu, W., Lim, P.P.K., Kadmiry, B.: Robotic sorting of ovine offal: discussion of a soft peristaltic approach. Soft Robot. **1**, 246–254 (2014)
72. Stommel, M., Xu, W.: Optimal, efficient sequential control of a soft-bodied, peristaltic sorting table. IEEE Trans. Autom. Sci. Eng. (2015)

Golf-Playing DARwIn-OP: A Theoretical Approach

Mauricio Gomez, Bakytgul Khaday, Adalberto Gonzalez, Ahmad Esmaeili and Eric T. Matson

Abstract This paper presents a theoretical approach for a humanoid robot performing autonomously in a golf setting. The main concentration of this research is on using vision and mathematical calculations to perform and evaluate the success of each test. Most of the problems encountered in making a humanoid perform complex actions like playing golf arise from the constraints in their features and implementations. Parts of the solution comprise linear interpolation and angle calculation. Challenges faced and solved include some that will lead to a better understanding of human–robot interaction.

1 Introduction

A few decades ago, to talk about robots performing daily activities was looked upon as almost impossible or only in a science fiction movie. Scenarios have widely changed, for instance, robots are becoming usual in our environment. They can accomplish a wide variety of works in many areas including entertainment,

M. Gomez (✉) · B. Khaday · E.T. Matson
Computer and Information Technology Department, Purdue University,
West Lafayette, IN, USA
e-mail: mgomezmo@purdue.edu

B. Khaday
e-mail: bkhaday@purdue.edu

E.T. Matson
e-mail: ematson@purdue.edu

A. Gonzalez
Mechanical Engineering, Purdue University, West Lafayette, IN, USA
e-mail: gonza252@purdue.edu

A. Esmaeili
Computer Engineering Department, Iran University of Science
and Technology, Tehran, Iran
e-mail: aesmaeili@iust.ac.ir

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_33

household, and hazardous workplace. Particularly, humanoid robots are very handy because they have human-like body parts and can be utilized to perform actions that only humans can do. Researching and building humanoid robots help us understand human nature in-depth and implement human capabilities in robots. One example of a humanoid is DARwIn-OP (dynamic anthropomorphic robot with intelligence—open platform) [1].

In this paper, DARwIn-OP humanoid robot was researched and deployed in playing golf. Even human beings need plenty amount of practice to become better players. Playing golf requires lots of techniques including mathematics, force application, and angle selection. Using these techniques, this article provides an approach on how DARwIn-OP can be tamed to play golf.

The purpose of this research is to showcase all the capabilities of humanoid robot DARwIn-OP and to share the implementation techniques with other researchers.

Most of the problems that can be found while making a humanoid able to perform complex actions like playing golf arise from the constraints in their features and implementations. In DARwIn-OP's case that we have used in this paper, some of the limitations are as follows:

- Limitations in its vision that makes it difficult to recognize wide ranges of colors or even to tell different colors from each other.
- Lack of tools that help measure angles and distances.
- Limitations in the movement of the joints and their speed.
- Constraints with capabilities in controlling the balance of robot in various circumstances.

Among these constraints, some were solved by trial and error, i.e., setting different values to the controlling parameters of the robot and choosing the one that leads to the best solution, some others were neglected due to the lack of physical parts or software libraries, and finally, the rest were solved by mathematical modeling.

This paper is organized as follows: Sect. 2 is dedicated to the review of some related works and it presents prior knowledge of this research. In Sect. 3, the extended implementation methods of this study are discussed. Section 4 presents a comparison between the performance of the robot and a human. And finally, this paper concludes the proposed method and the future work.

2 Background and Related Work

Humanoid used to simulate sports as humans do is a new trend of research. One of the reasons that many researchers in robotics field focus on this kind of robots is their human-like shapes and set of movements. Still many problems exist to overcome when using humanoids to emulate human's daily activities and sports practices. In the field of humanoids practicing sports, there are several focuses. Some sports, such as playing soccer, attract more attention in humanoid research. In humanoid soccer playing case, the main driver of the research trends is the RoboCup competitions, in which the ambitious goal is final victory of a fully autonomous humanoid robot soccer players' team over the humans' team consisting of the most recent World Cup champions by mid-twenty-first century [2]. Efforts in this sport are of a wide variety, given that humanoid robots are still under development. Some of recent efforts are mentioned next. Iterative learning algorithms to intercept a ball are shown in [3] where the focus is to let robots to learn the same way humans do, that is to test and evaluate or iterative way. Other research was held to understand how likely a robot will be allowed to play against humans in soccer matches [4]. These studies survey the issues on human–robot interaction from the view of human safety in a soccer field environment. The same work mentions that soon humans will be only spectators.

Other field that has gained interest of researchers due to competitions held around the world is robot fights, also known as sumo bot. However, some competitions start from the building of the robot, followed by a competition of individual performance, which finishes with a sumo-style fight [5]. The design of two classes of semiautonomous competitive robots to compete with each other in games of sumo wrestling and soccer is shown in [6]. There are some studies to make some humanoids like DARwIn-OP able to play bowling [7]. Ping pong is another sport that has gained attention. Fields like learning techniques are applied in table tennis in [8, 9]. Optimal momentum compensation method and a position-based impedance control scheme are used to generate body motion so that the robot absorbs the recoil force and ensures successful hit to the ball for a humanoid playing table tennis as presented in [10]. Many other researches are done in other sports; however, they are not mentioned here due to lack of space.

Golf is not an exception for robots designed either to play the game, also known as edutainment robots, or to enhance human experience in this sport. Edutainment, in general words, is the act to learn in a fun way [11]. The design and development of a golf ball picking robot design to reduce the human intense labor required for that task is presented in [12]. Also was designed and implemented a prototype of a vision-guided golf ball-collecting mobile robot in [13]. These works differ from ours in the sense that their robots are developed to pick up and collect the balls at the end of the day, letting humans be able to continue playing with those balls next day, while in our work, the humanoid robot plays golf with a small set of golf.

Other recent effort in golf robotics is in terms of improvement for training systems to let humans be able to boost their skill in swinging the golf ball. That is the case of a design of a 6-DOF (degrees of freedom) robot manipulator, which controls the position and orientation of the golf club grip point [14]. This kind of research differs from ours in the sense that it is a portion of a robot, not a complete humanoid, and the fact that it is more focused on the increase of humans' experience, instead of studying the humanoid performance itself.

Specifically in the performance of robots playing golf, a study of the development of an autonomous robotic system for playing mini-golf is shown in [15]. This work presents a robot that moves autonomously using 2 DC motors controlled by a remote PC using fuzzy logic system. It uses global camera vision that serves as the eyes of the robot to identify the environment completely. This work is different from the work presented in this paper, given that in our experiments, a completely autonomous system was utilized. The autonomous system includes that the vision is entirely provided by the camera built in the DARwIn-OP robot.

An autonomous robotic system for playing mini-golf was developed in [16]. The system was built using a Yaskawa robot fitted with two arms, and software was programmed in C++ language to control it. At that time, they mentioned in their work that no work had been reported on a system to play mini-golf. This specific work differs from the one we developed since they did not use a complete humanoid to make the work. Their system uses two scanners in x and y coordinates to locate the ball and two arms as actuators. Hence, its complete system is not in a completely autonomous and stand-alone humanoid robot, like DARwIn-OP, which we used in our experiments.

3 Implementation

In order to create a complete scenario where our humanoid robot could be able to play a golflike game, we bought a set of golf. Humans use this kind of set as a distraction while they go to void, in medical terms. The set acquired is shown in Fig. 1. And its size and dimensions fit to the robot we used for the tests. The ball was painted in blue color. We made a distance calibration for the ball and the hole. To determine the distance that the ball and the hole are from the position of the robot, we used a linear interpolation for different distances.

The humanoid used to run our experiments was DARwIn-OP, of which all kinematics are explained in [17]. We applied changes to the program of soccer provided within the DARwIn-OP system.

Fig. 1 Golf set used for the tests

3.1 Overall Process

The complete process of our robot to play golf is shown in a diagram in Fig. 2. It is important to explain all the process as follows. Firstly, we used the ball finding/tracking function that comes with DARwIn-OP programming libraries. These libraries work with the HSV (hue, saturation, and value) color model as stated in [7]. Hence, all objects were assigned in different colors for the robot to be able to identify them. We modified the parameters of the angle of vision to locate the ball that is different from the normal position of soccer mode that comes with the development SDK in the robot. The specific ranges of angle we used for the ball location are horizontal or also known pan [5, −14] and vertical or tilt [5, −5]. We also created the movement sequences for lateral step and angle shift to adjust body position to be able to drive the ball to the hole and the swing motion.

The ball was painted in blue and the hole in red color. The complete process is shown in Fig. 2. It starts with the ball tracking program segment. Find the ball, walk toward the ball, and put the ball in angle to hit are activities with the angles mentioned above which are included in the first program segment. Immediately afterward, the program continues to locate the hole with only head movements. If the hole is not in his vision spread, the robot moves 45° to the right, trying to keep the ball in the same angle, and it looks again for the hole. Latter verification of hole and 45° of motion in the vision spread is repeated until the hole is found. When the location of the hole is established by the different angles of pan and tilt of DARwIn-OP's head, some calculations are performed to obtain the angle between the imaginary adjacent line from the ball and the hole to the position of the ball and the line from the robot to the ball. The goal for the latter angle is to set the result

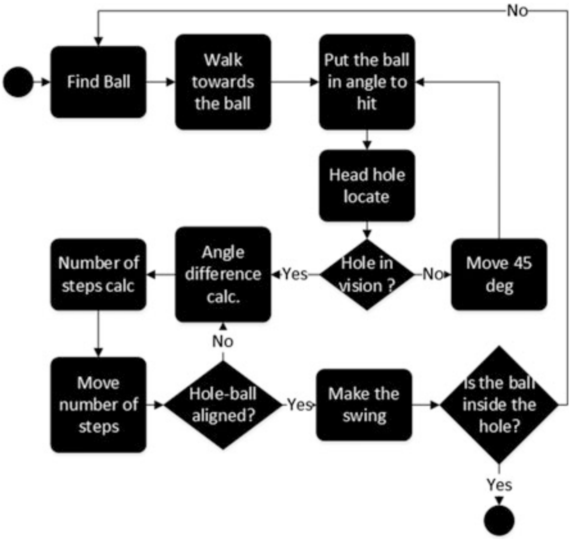


Fig. 2 Autonomous golf-playing DARwIn-OP system diagram

within a range of 95° and 125°. The reason for this angle, not being a right angle, is that the swing that the robot performs is not parallel to the robot position. If this angle is higher than the range mentioned above, a calculation of number of steps is performed to set the angle in within that specific range. A movement sequence of the robot to change angle of his position toward the line between the hole and the ball was created. These sequences of reducing the angle or turning right were created with an 11° reduction each time it is run. A movement sequence was created for the left turning or angle increase between the robot, the ball, and the hole, which increases the angle in 11°. When the steps calculated are executed, DARwIn-OP proceeds to verify whether the angle is within the range or if the hole and ball are aligned. In case not, the process of verification and angle calculation is run until alignment is achieved. When the hole and the ball are aligned, the robot proceeds to make the swing. After hitting the ball, it verifies whether the ball went inside the hole. When the ball is not inside, the overall process has to be run again. When the ball is inside the hole, the process is considered accomplished or finished.

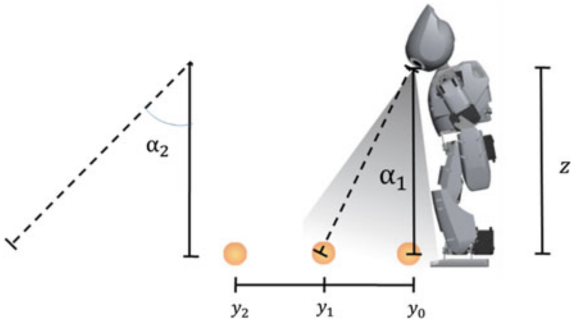
3.2 Mathematical Models

There were two different mathematical models applied in the solution that are explained as follows.

Linear Interpolation The first mathematical model used is a linear interpolation utilized to determine the distance that exists between the ball and the robot and the hole and the robot. The way it was used is shown in Fig. 3. The same figure shows that the ball located in y_2 is not in his sight view; however, moving his neck in tilt direction can be adjusted to locate the ball in the center of his view. The values obtained generated two databases, one for the ball and one for the hole. They contain the combinations of distance and pixel size range for each distance.

The distances taken into account were all moving along the y-axis. In other words, the value only changes on y and x value will always be 0. These are intermediate values that are used as inputs to make a linear interpolation for the size

Fig. 3 Relation between ball/hole distance to the robot and pixel size



of the ball when the robot stops walking toward it. Normal linear interpolation is as follows:

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0} \tag{1}$$

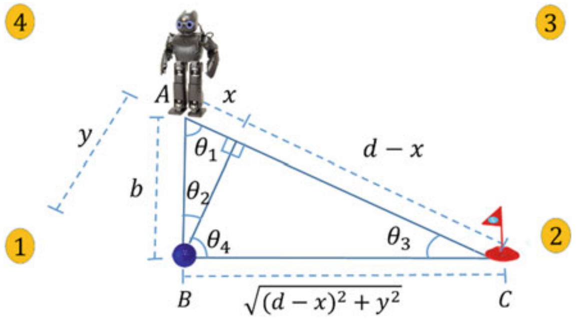
As mentioned above, the values in x-axis stay intact for the values obtained. Applying this principle to the case of distance of the ball and hole, it will be as shown in formula 2. Variable d corresponds to the distance calculated. The different iterations of y are the values of distance stored in the database. s_i symbolizes the different values of size that contain the pixel size registered by the vision system of DARwIn-OP when tests are running.

$$d = y_1 - (y_1 - y_0) * \left(\frac{s - s_n}{s_1 - s_0}\right) \tag{2}$$

Angle Calculation The second mathematical model pertains to the limitation in the movements of the robot's arm in order to hit the ball. As mentioned before, finding the ball and reaching it, the robot looks for the position of the hole by adjusting the rotational angle of its neck as much as the hole is positioned in its center of view. Formally, the robot forms a Cartesian coordinate system with the origin at the position of the ball as depicted in Fig. 4. As it can be seen, the best condition for the robot to hit the ball toward the hole is to form a 110 ± 15° angle between the line he is standing on and the line of the ball. This angle is shown in Fig. 4 as $\theta_2 + \theta_4$; i.e., when the rotational angle of the robot's neck locates the hole at 100°, it is between the goal targets. We assume that the robot merely tries to find the hole in region 2 shown as a circled number. Let us take θ_1 as the amount of degrees the robot rotates its neck counterclockwise to locate the hole. In order to provide him a proper orientation to direct the ball toward the hole, he should rotate its body a specific amount of degrees clockwise around the ball, where

$$\theta_2 = 90 - \theta_1 \tag{3}$$

Fig. 4 Locating the hole and adjusting the body to hit the ball



In case no hole is found in region 2, the robot rotates 45° counterclockwise and then repeats the process above until it locates the ball in region 2 from the robot standpoint.

From the linear interpolation, we get the distance to the ball. The same value is shown as b. The way to get the value of x is with the formula 4.

$$x = b^{*}\cos(\theta_1) \tag{4}$$

On the other hand, with the linear interpolation, the distance to the hole shown as d is also calculated, which leads us to the next formula that shows how we get the value of θ_4.

$$\theta_4 = tan^{-1}(\frac{(d-x)}{y}) \tag{5}$$

Next, it is necessary to verify that the angle should be between the goal ranges as shown in the next formula:

$$\theta_2 + \theta_4 = [95, 125] \tag{6}$$

The values of 95 and 125 were not arbitrary. They were determined after an exhaustive number of tests performed. Figure 5 illustrates a snapshot sequence of a successful test. Number 1 shows the initial state where the robot starts scanning to find the ball. Number 2 shows the tracking ball that locates the ball in a position to be able to hit it with a swing, no matter the position of the hole. Next, number 3 depicts when the robot is looking for the position of the hole, and after it is found, it

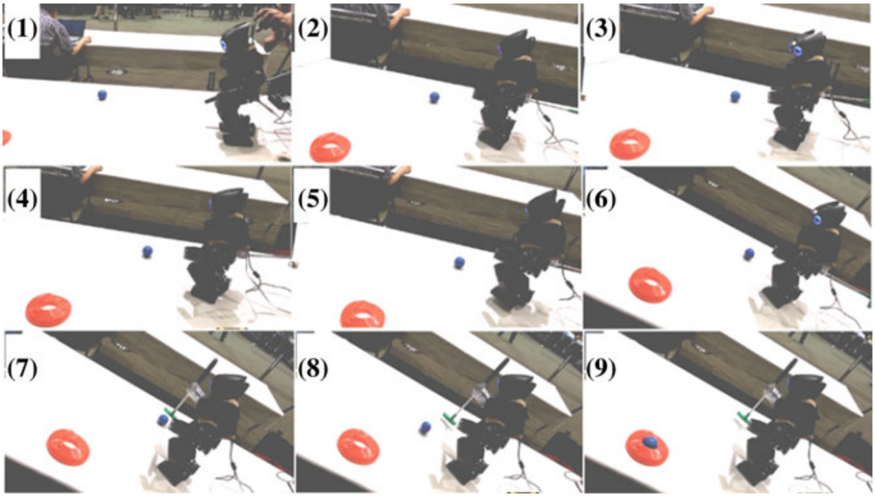


Fig. 5 Snapshot sequence of successful test

calculates the corresponding angles. Furthermore, in number 4, it makes 2 angle-shifting steps to locate it in the desired angle. Snapshot 6 then shows the visual verification of the hole and the ball, followed by the swing movement sequence in number 7. Number 8 clarifies the ball in motion. To finish with the ball inside the hole is number 9, which makes successful test.

4 Experiment Results

After finishing the program, we conducted a correlation analysis between the performances of a human versus DARwIn-OP. We directed a comparison of 5 trials for each of the combinations. Combinations were of [25, 45, 65] degrees, and initial location of the ball is [40, 60, 80] cm of distance from the hole. The success trials out of the 5 combinations and total effectiveness for each of the actors are shown in Table 1.

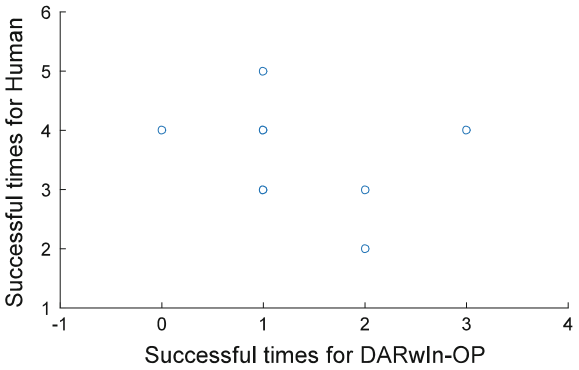
The human that realized the tests is by no means an expert to play with the golf set that was acquired for this research. However, an improvement of human's performance was noticeable while the tests were taking place, given that in the last tests, his performance was more successful than the first swings he performed. In the case of DARwIn-OP, this was not a part of the program, so he only had to follow the program with all calculations mentioned in previous chapter. This means that it does not contain any learning algorithms at the moment of the tests.

Figure 6 shows the scatter plot of the values of successful trials of both actors taken into account in the tests. A calculation of the correlation value is −0.27. This means that they have a weak negative correlation. The performance of DARwIn-OP is variable for different reasons such as the surface where he is walking and a minimal difference of brightness of light. However, in spite of all those factors that affect the performance of DARwIn-OP, the system showed a 27 % of success in overall tests.

Table 1 Effectiveness of DARwIn-OP and Human

Combination		Actor	
Angle	Distance	Human	DARwIn-OP
25	80	3	1
25	60	2	2
25	40	4	1
45	80	4	3
45	60	3	2
45	40	5	1
65	80	4	1
65	60	3	1
65	40	4	0
Success rate in 5 trials (%)		71	27

Fig. 6 Performance of DARwIn-OP versus human



5 Conclusion

In this work, we have developed a theoretical approach that makes the humanoid named DARwIn-OP able to play golf. This approach is still not in human or natural sizes. However, the study made us realize a lot of complexities that are involved and a device like a humanoid has to overcome. Some of the adversities are not common in any other kind of robot. Nevertheless, in this work, we showed that it is possible to make a humanoid be able to perform human activities, even though this type of studies is not addressing completely the problems of a human daily life. They are developed for at least two different purposes. One is to make the learning experience in an entertaining way. The other is to improve the human–robot interaction. We have shown that the performance of the robot is desirable.

One of the future improvements that can be done is to make the robot learn and improve his accuracy on the swing. This approach could be done using iterative or reinforcement learning. And further extension of work could be to make the program to be able to work as a team in a hockey field environment. Of course, this will make a completely different and even more complex setting, where not only the accuracy will be an important factor, but also the interaction with other mobile agents. Those agents can be part of the team or opponents.

References

1. Ha, I., Tamura, Y., Asama, H., Han, J., Hong, D.W.: Development of open humanoid platform DARwIn-OP. In: SICE Annual Conference, Tokyo, Japan (2011)
2. Eaton, M.: Evolutionary Humanoid Robotics. Springer, New York (2015)
3. Gomez, M., Kim, Y., Matson, E.T.: Iterative learning system to intercept a ball for humanoid soccer. In: International Conference on Automation, Robotics and Applications, Queenstown, New Zealand (2015)
4. Takahashi, T., Shimizu, M.: Is that Robot Allowed to Play in Human Versus Robot Soccer Games. In: Lecture Notes in Artificial Intelligence. Springer, Switzerland (2015)

5. Jardón, A., Zafra, P., Martínez de la Casa, S., Giménez, A.: CEABOT: Nationalwide Little Humanoid Robots Coimpetition; Rules, Experiences and New Challenges. Venice, Italy (2008)
6. Benavidez, P., Gleinser, C., Jaimes, A., Lbrado, J., Riojas, C., Jamshidi, M.: Design of Semi-Autonomous Robots for Competitive Robotics. Puerto Vallarta, Mexico (2012)
7. Tazhibayeva, S.B., Kuanyshbaiuly, M., Aldabergenov, A., Hong, J.H., Matson, E.T.: Bowling with the DARwIn-Op Humanoid Robot. In: Robot Intelligence Technology and Applications 2, pp. 733–741. Springer, Switzerland (2014)
8. Kober, P.J., Müllling, K., Krämer, O., Neumann, G.: Towards robot skill learning: from simple skills to table tennis. In: European Conference, ECML PKDD, Berlin Heidelberg (2013)
9. Mülling, K., Kober, J., Kroemer, O., Peters, J.: Learning to select and generalize striking movements in robot table tennis. Int. J. Robot. Res. 263–279 (2013)
10. Sun, Y., Xiong, R., Zhu, Q., Wu, J., Chu, J.: Balance Motion Generation for a Humanoid Robot Playing Table Tennis. Bled, Slovenia (2011)
11. Lund, H.H., Nielsen, J.: An edutainment robotics survey. In: International Symposium on Human and Artificial Intelligence Systems: The Dynamic Systems Approach for Embodiment and Sociality, Fukui, Japan (2002)
12. Pereira, N., Ribeiro, F., Lopes, G., Whitney, D., Lino, J.: A golf ball picking robot design and development. In: International Conference on Experimental Mechanics, Porto, Portugal (2015)
13. Wu, S.-L., Cheng, M.-Y., Hsu, W.-C.: Design and implementation of a prototype vision-guided golf-ball collecting mobile robot. In: International Conference on Mechatronics, Taipei, Taiwan (2005)
14. Camarano, T., Beck, J., Li, B., Wu, W., Chow, L., Wu, T., Drumm, D., Lebouitz, M., Napolitano, D.: Kinematic and dynamic modeling of robot manipulator for golf swing training system. In: IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, 2015. Kinematic and Dynamic Modeling of Robot Manipulator for Golf Swing Training System. Camarano, T., et al. (ed.) Seville, IEEE International Conference on Industrial Technology (ICIT). IEEE, Spain (2015)
15. Jabson, N.G., Leong, K.G.B., Licarte, W., Oblepias, G.M.S., Palomado, E.M.J., Dadios, E.P.: The autonomous golf playing micro robot with global vision and fuzzy logic controller. Int. J. Smart Sens. Intell. Syst. **1**(4) (2008)
16. Jouaneh, M., Carnevale, P.: Watch out, tiger woods! [golf playing robot]. Robot. Autom. Mag. **10**(2), 56–60 (2003)
17. Williams II, R.L.: DARwIn-OP humanoid robot kinematic. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, IL, USA (2012)

An Advanced Spider-Like Rocker-Bogie Suspension System for Mars Exploration Rovers

Aswath Suresh, Nitin Ajithkumar, Sreekuttan T. Kalathil, Abin Simon, V.J. Unnikrishnan, Deepu P. Mathew, Praveen Basil, Kailash Dutt, Ganesha Udupa, C.M. Hariprasad, Maya Menon, Arjun Balakrishnan, Ragesh Ramachandran, Arun Murali and Balakrishnan Shankar

Abstract This paper describes the working of the system design for the Mars rover. The rover, developed to compete in the Mars Society's University Rover Challenge 2015, was designed to perform various tasks such as site survey, sample return, equipment servicing, and astronaut assistance in a Mars-like landscape of dry, non-vegetated, rocky terrain. The complete design features a bioinspired eight-wheeled drive mechanism, an integrated robotic arm along with a stereo vision technique for advanced image processing. This paper focuses on the drive mechanism of the rover design. The 8-wheeled rover combines the rocker-bogie mechanism with four rocker wheels and four spider-leg wheels. The spider legs ensure that it can traverse over heights greater than the chassis height, which could be three times as much as the diameter of the wheels. NASA's current rover can only traverse a height twice the diameter of the wheel. Additionally, the wheels are actuator-powered, and hence, the slope of the rover can be adjusted in such a way that it does not topple for a wide range of inclination allowing the rover to traverse over highly rugged terrain. The rover design can be modified for many applications notably the exploration of alien planets, deep sea trench, and other environments

A. Suresh (✉) · N. Ajithkumar · V.J. Unnikrishnan · D.P. Mathew · A. Balakrishnan · R. Ramachandran
Department of Electronics and Communication Engineering,
Amrita Vishwa Vidyapeetham, Kollam 690525, Kerala, India
e-mail: aswathashh10@gmail.com

N. Ajithkumar
e-mail: nitinakumar@gmail.com

S.T. Kalathil · A. Simon · P. Basil · K. Dutt · G. Udupa · C.M. Hariprasad · M. Menon · A. Murali · B. Shankar
Department of Mechanical Engineering,
Amrita Vishwa Vidyapeetham, Kollam 690525, Kerala, India
e-mail: sreekuttan44440@gmail.com

G. Udupa
e-mail: ammasganesh@gmail.com

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_34

where human exploration is almost impossible. This effort to make the rover mechanism more efficient may one day be instrumental in detecting life and many such possibilities, in Mars and other planets.

Keywords Drive mechanism · Mars rover · Spider legs

1 Introduction

The Mars rover is a vehicle that has been designed to traverse the rugged terrains of Mars and collect samples of various items on Mar's surface. Scientists over the years have tried to explore the possibility of life on Mars. Such explorations have been mostly done using rovers. Hence, rovers need to be specially designed to traverse all kinds of terrains and must be equipped with state-of-the-art technology. A common design element is most rovers over the years, which is the rocker-bogie mechanism. The rocker-bogie mechanism has quite a lot of advantages and is hence a well-established mechanism. The main advantage is that it ensures that all the wheels of the rover are in contact with the ground at all times. This advantage is key to creating a stable all-terrain system. Consequently, the traction of the rocker-bogie provides is equal and reliable, allowing a smooth running even on the uneven terrains.

The earliest notable Mars explorer, the *Pathfinder*, landed on Mars with a fully function rover on Mars on July 4, 1997. The rover carried a wide array of scientific instruments to analyze the Martian atmosphere, climate, geology, and its rock and soil composition. Sojourner, the Pathfinder's rover, made observations that answered numerous questions about the origin of the rocks and other deposits on Mars. Following its stead, the *Opportunity* successfully investigated soil and rock samples and managed to take panoramic images of its landing site giving us valuable information about the Martian terrain and other site conditions. It is the data that were collected in these missions, using sampling technology, which allowed scientists at NASA to theorize about the presence of hematite and consider exploring the possibility of finding water on the surface of Mars. *Curiosity*, another historic milestone in the history of alien planet exploration, was assigned the role of investigating Martian climate and geology. It assessed whether the selected field site, Gale Crater, had ever offered environmental conditions favorable for microbial life and future investigated the role of water in planetary habitability as preparation for future human exploration [1–4]. In these types of rovers, only 6-wheeled rocker-bogie [5–12] suspension is used.

This paper proposes an 8-wheeled rover which combines four rocker wheels which form the rocker-bogie mechanism along with four spider-leg wheels. Figure 1 shows the SolidWorks model of the rover. The spider legs ensure that it can traverse terrains with heights much greater than the chassis height. Additionally, since the wheels are actuator-powered, the slope of the rover can be adjusted in such a way that it does not topple for a wide range of inclination and allows the

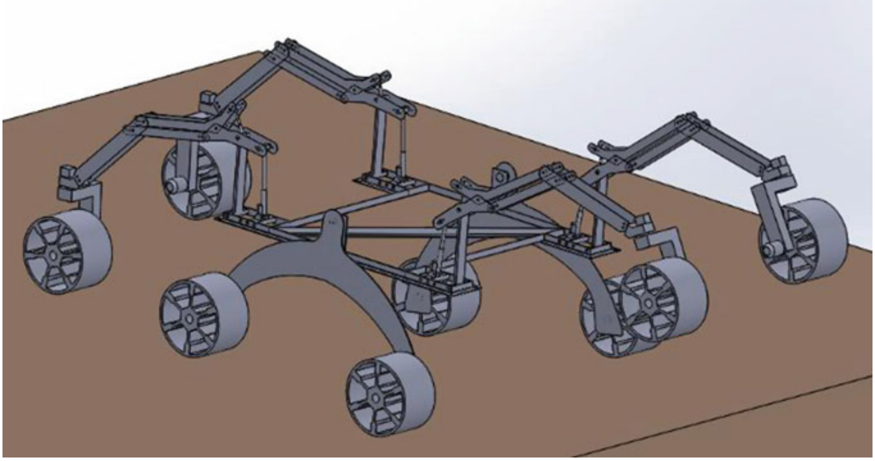


Fig. 1 SOLIDWORKS model of the rover with front spider legs elevated

rover to traverse over highly rugged and uneven terrain. It provides a significant amount of supplemented traction with the ground even in terrains where there is a negative slope or vertical drop of around 1 m using a spring–damper suspension mechanism. This is a significant improvement to the existing rocker-bogie mechanism which provides traction using its body weight alone. Moreover, this is done without compromising the strength of the chassis. The chassis has a factor of safety of 3.2 as per finite element analysis. The leg, which is a combination of the four-bar linkage, spring–damper, and the linear actuator, provides assistance for the rover to traverse over small hindrances. A member is hinged to the chassis. One end of that member is pinned to the actuator and the outer end to a four-bar linkage, making the system stable and flexible at the same time. Thus, the rover mechanism that has been described in this paper allows the rover to traverse all kinds of surfaces while maintaining stability and protecting the instruments that the rover carries.

2 Rover Mechanical Design

The design of the rover is to allow it to traverse rough and rocky terrain which at present is not possible using conventional vehicle design. Therefore, the major aim is to design a rover which has an effective mechanism removing the disadvantage of modern day rovers. The chassis designed is in accordance with the set of rules and requirements that have to be met in order to contest in the event. There are certain restrictions set on the length and width of the chassis, its overall weight, ground clearance, etc., by the governing body of the “University Rover Challenge 2015,” and it was made mandatory that the rules be followed by the competing teams.

Table 1 Material properties of Aluminum 6061 alloy

Property	Value
Ultimate strength	310 MPa
Density	0.0975 lb/in^3
Strength-to-weight ratio	Moderate
Machinability	Highly machinable/weldable using TIG
Cost	Moderate
Availability	Available in Indian market

2.1 Material Selection

Performance and strength are the major parameters which reflect the reliability of a vehicle. While designing and fabrication of the rover chassis, suitable low cost material had to be selected which maintained high strength-to-weight ratio, thereby improving the performance of the rover. Considering various parameters such as yield strength, material availability, density, strength-to-weight ratio, and mainly the availability and cost of the material, various options were explored. The search narrowed down to either graded aluminum or composite materials. Composite materials such as carbon fiber are mainly used in the conventional NASA designs of MARS rovers. Based on the factors in Table 1, we choose our material of construction as Aluminum 6061 just to produce the prototype required for Mars Desert Research Station mars-like terrain.

Aluminum 6061, a precipitation hardening alloy, majorly comprises of magnesium and silicon. It was chosen for its good physical properties and great weldability. Being one of the most common alloys of aluminum, it is easily available and cheap. Pretempered grades such as 6061-O (annealed) and tempered grades such as 6061-T6 (solutionized and artificially aged) and 6061-T651 (solutionized, stress-relieved stretched, and artificially aged) are the commonly available types [13].

2.2 Chassis Model

In order to design and manufacture the original chassis, continuous optimization had to be done. The FOS has to be higher, and the deflection should be minimal. Hence, the careful selection of the channel type is required for further modeling. For minimal deflection and increased FOS, a rectangular channel of 3 mm thickness was selected as the suitable channel. The rectangular channel of 3 mm thickness is capable of withstanding both vertical and longitudinal load. Figure 2 shows the channel dimension, and Table 2 shows the channel specification. As most of the loads are exerted upon the mounting regions of linear actuators, the channel can now take fluctuating loads. The final design incorporates enough space which

Fig. 2 Dimensions of the channel used

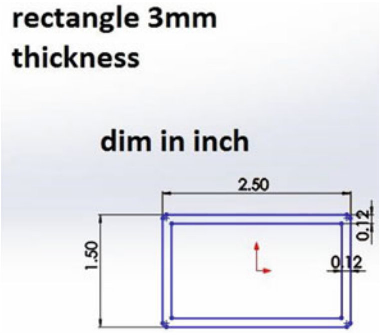
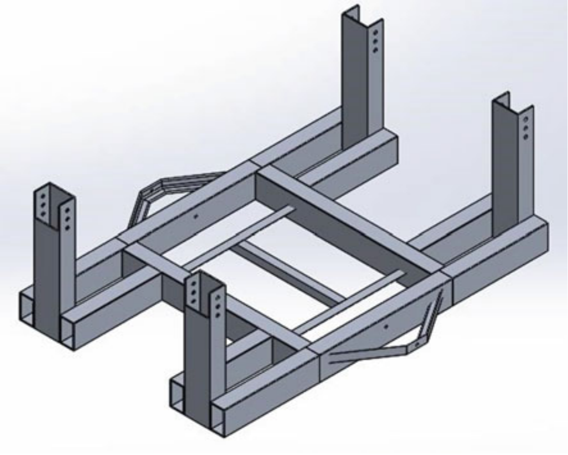


Table 2 Channel Specifications

Material Specification	Al-6061 T6
Type of channel used	Rectangular channel
Channel size	2.5 in * 0.5 in
Thickness	3 mm
Chassis dimensions	105 cm * 80 cm
Weight	4.1 kg

satisfies the component restraints and provides necessary strength for the rover. The chassis analysis was done after designing the final chassis. Figure 3 shows the final chassis design.

Fig. 3 Final design of the chassis



3 Static Analysis of Model

For the static analysis, the fixed supports were given as the support members of C arms. The overall weight requirement comes around to be 50 kgf. So a force of 500 N is given to each of the leg members. Figure 4 shows the distribution of forces and the fixed support, whereas Fig. 5 shows the meshing. Figures 6 and 7 show the safety of factor and total deformation.

The values obtained from the FEA analysis are as follows: factor of safety = 5.8571(withstand high load), deformation = 0.001886 cm (little deformation), and max equivalent stress = 4.263 * 10^7 Pa (low stress). Analyzing the data, it can be concluded that the chassis is strong and can easily withstand high strength. Therefore, the chassis need not be further optimized. The chassis is then built with real-time boundary conditions. Figure 8 shows the real chassis model.

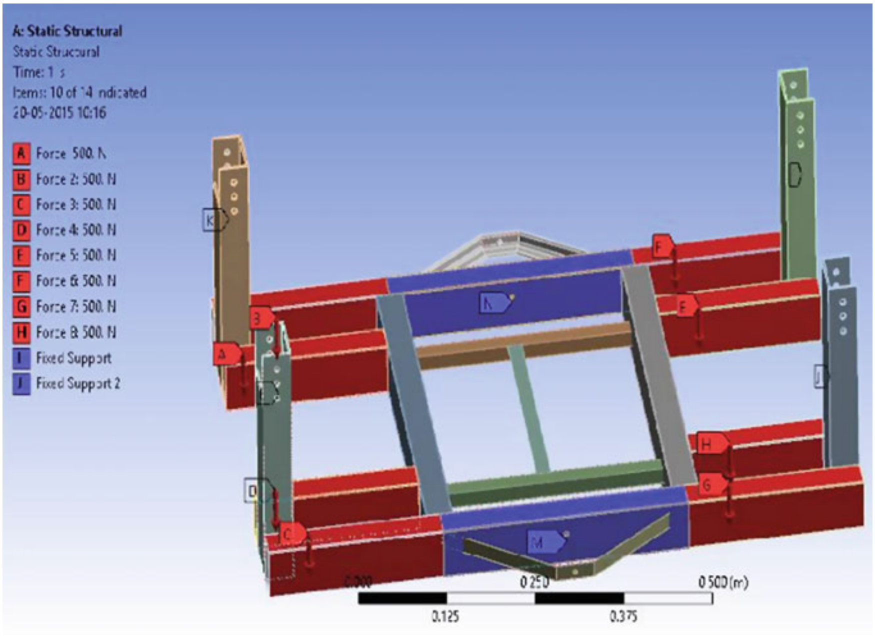


Fig. 4 Distribution of forces and the fixed supports

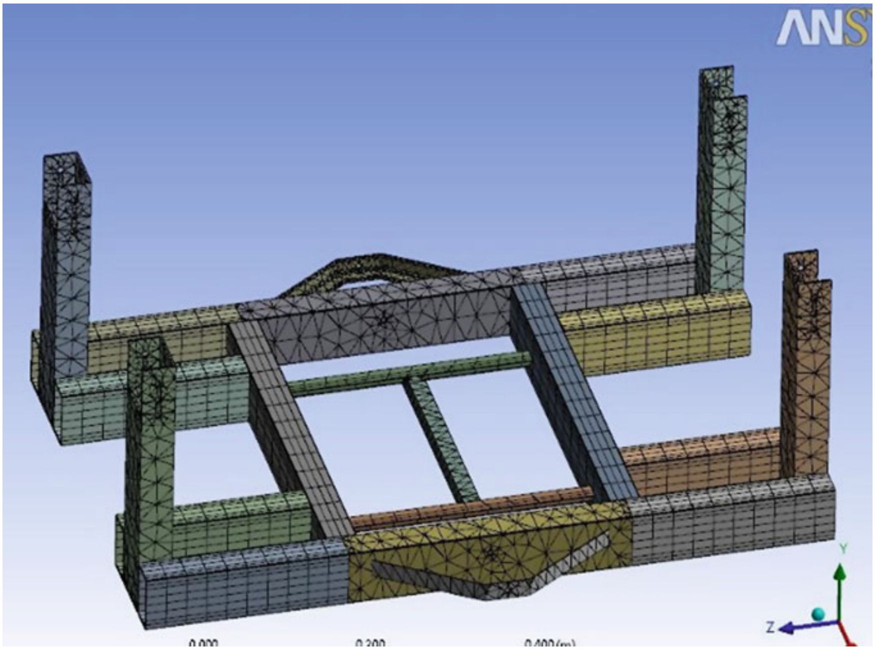


Fig. 5 Meshing

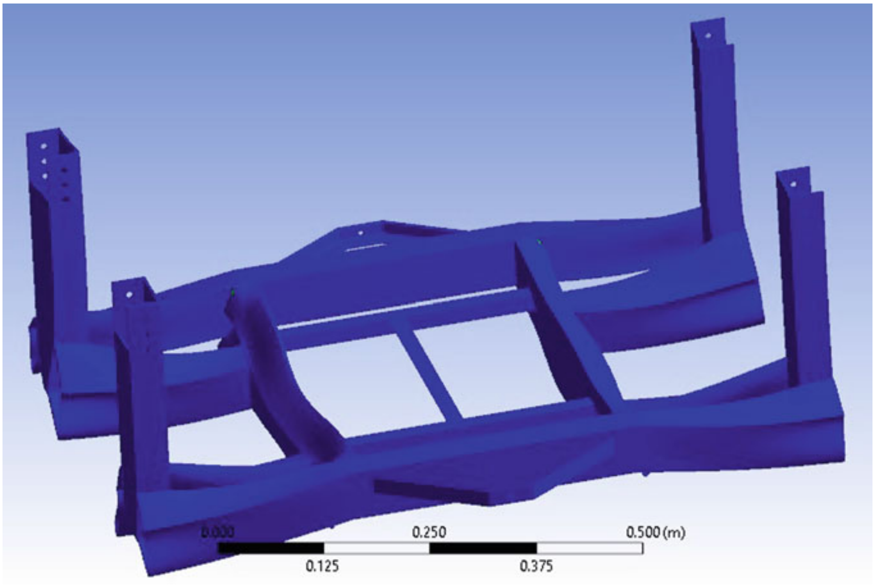


Fig. 6 Factor of safety

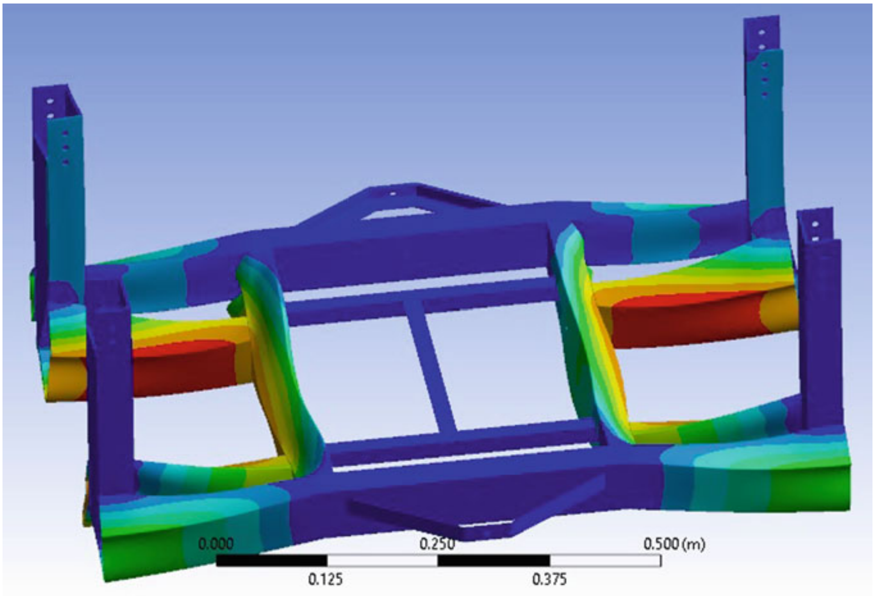


Fig. 7 Total deformation

4 Mechanical Development and Kinematic Analysis

For the chassis that has been built, a proper mechanism has to be developed which enables the legs of a vehicle to traverse up steps/rocks and still be able to take the impacts of varying ground terrain. The primary motive of the mechanism is to allow the rover to a climb high stepped terrain and still have a suspension system in place to provide restoring force for the legs to come down and provide traction for the wheels over the ground terrain. Such a mechanism will allow rovers to travel rough and rocky terrain while at present is not possible for conventional rovers.

In currently used designs, there is a limit to the height that the rover can traverse, and it is extremely difficult for these designs to climb up very high steps. Using suspension designs with very soft shocks can cover rough and rocky terrain to an extent, but unfortunately, it is impossible for it to climb up huge steps and rocks. It is well known that an unexplored terrain can have unexpected challenges. Thus, it is essential for a mechanism to exist which can overcome such unexpected challenges. It is possible to test the conventional designs in unexplored areas, but these designs tend to suffer from disadvantages which would prevent them from prolonged use; hence, such designs are not feasible. Thus, there is a need for a novel and innovative design which will enable rovers to face unaccounted landscape challenges and allow it to carry on with the unmanned exploration, without human intervention or repair issues.

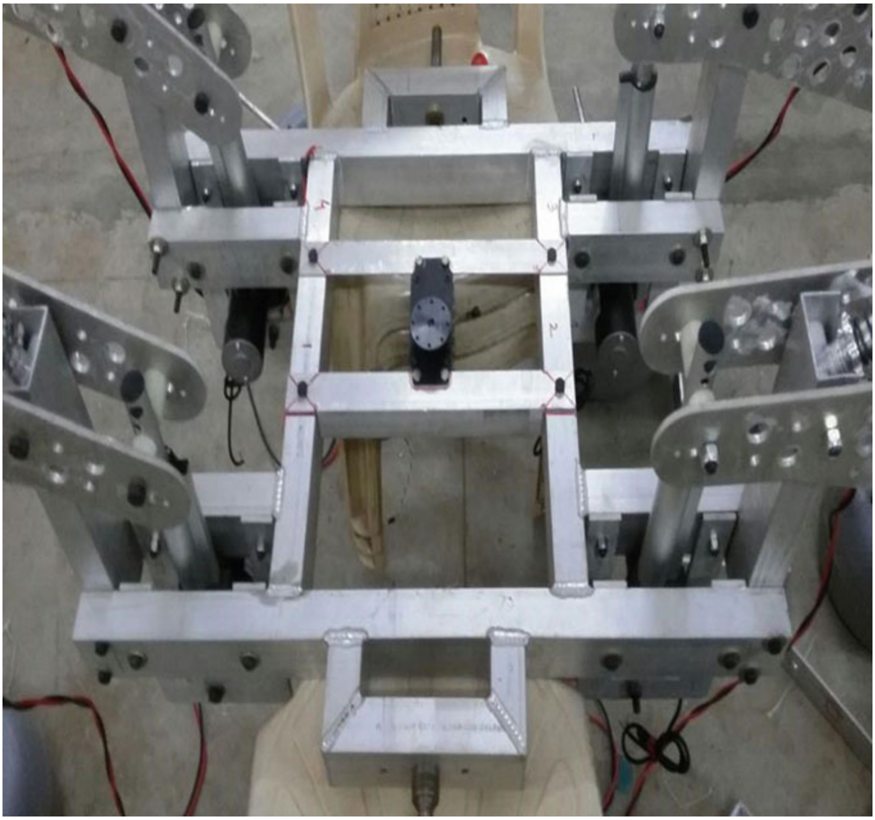


Fig. 8 Real-time chassis model

4.1 The Spider Mechanism

The spider-leg mechanism ensures that the vehicle can traverse over a height far greater than the chassis height. This could extend as far as thrice the diameter of the wheels. Additionally, since the mechanism is actuator-powered, the slope of the rover can be adjusted in such a way that it does not topple for a wide range of inclination and allows the rover to traverse over highly rugged or uneven terrain. It provides a large amount of traction with the ground even in terrains where there is a negative slope or vertical drop of around 1 m using a spring–damper suspension mechanism, and it achieves this without compromising on the strength of the chassis.

4.2 Line Diagram Depiction Showing Initial Position

Figures 9 and 10 depict how the spider legs help to support the load when all the wheels are in ground. Notice that initially, all the linkages are pivoted at the center point of Sl no 3 as shown in Fig. 9. This pivot point moves only when the rover encounters an unexpected terrain. The springs are mainly involved in traction control and in supporting the weight.

Figure 10 clearly depicts that the spider-leg, which is a combination of the four-bar linkage, spring–damper, and the linear actuator, provides assistance for the rover to traverse over small hindrances. As mentioned earlier, one link of the mechanism is hinged to the chassis. One end is pinned to the actuator and the other end to a four-bar linkage.

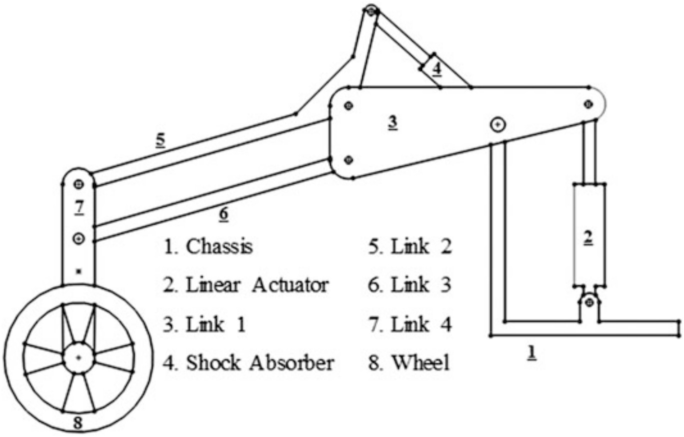


Fig. 9 Wheel in ground level

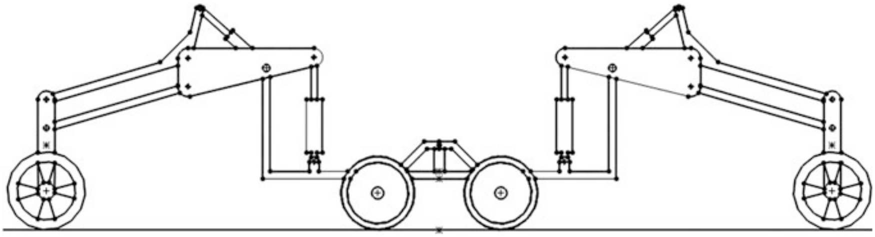


Fig. 10 The complete side view of the spider legs

4.3 Line Diagram Depictions Showing the Movements of the Rover

In Fig. 11, an example of the working of the spider-leg mechanism is shown at its mean position. At this position, the links are able to transmit the shocks of varying terrain to the shock absorber. When the wheels of the spider-leg mechanism encounter a small obstacle relative to its wheel size, it is able to easily overcome the obstacle as the shock absorber mounted on the top of the wheel's link gets compressed and the lift of the wheel is compensated this way.

In Fig. 12, as the spider-leg mechanism encounters a high obstacle, the linear actuator is retracted wirelessly, this causes the link mechanism to extend out and reach a greater height, and the shock on top of the mechanism is also compressed. However to get maximum reach and traction, there needs to be 2 spider legs in front of the vehicle. Therefore, 2 legs are raised. After this initial step, the vehicle rests on

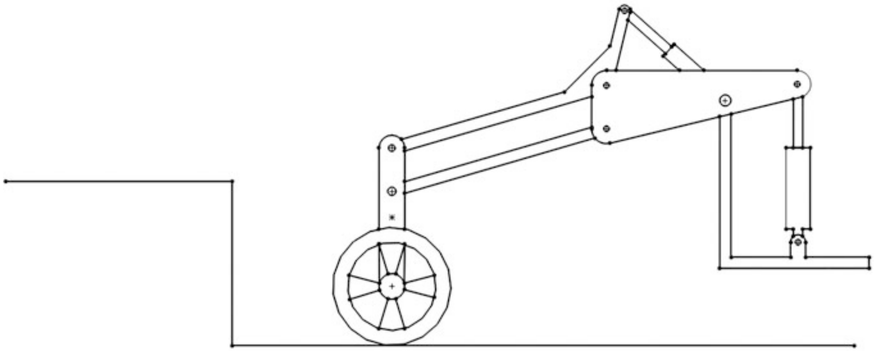


Fig. 11 An example of spider-leg mechanism approaching a stepped terrain

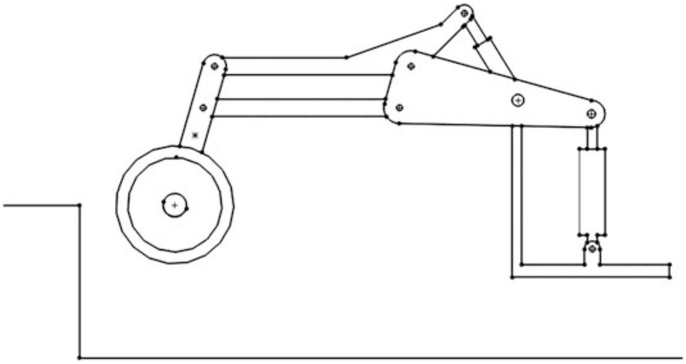


Fig. 12 An example of spider-leg mechanism raising itself

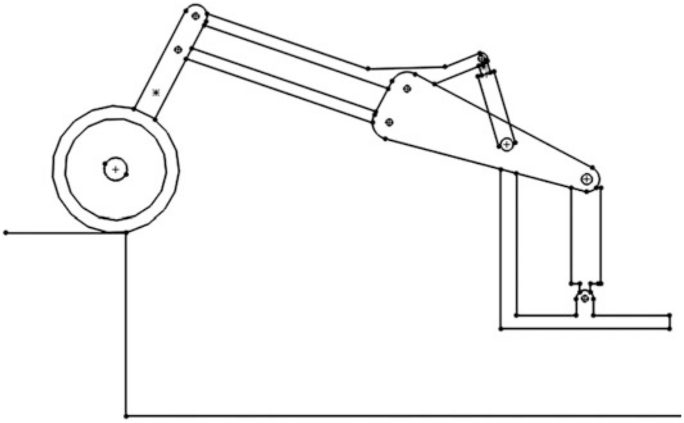


Fig. 13 An example of spider-leg mechanism overcoming a stepped terrain

the middle 4 wheels and the 2 rear spider legs. The lifting of the front spider legs allows the vehicle to reach terrain previously inaccessible and gain traction on top of the high terrain.

In Fig. 13, to effectively climb over a high obstacle, a total of 4 spider legs are required on a vehicle with individual drives, 2 spider legs at the front, which can be raised onto the high terrain and gain traction on the terrain, and 2 spider legs at the rear of the vehicle. The legs at the rear lower onto the ground effectively lifting the vehicle and giving a push to the entire vehicle, while the spider legs in front of the vehicle act to gain traction at the top of the obstacle and effectively pulls the entire vehicle up. This combination of the two forces allows the vehicle to overcome high stepped terrain.

4.4 Software Design Depiction

A SolidWorks model was designed, and the rover assembly was completed. This was essential for the manufacturing of the model. Figure 14 shows the solidWorks' rendered image of the mechanism, and Fig. 15 shows the isometric view of the entire rover making it easier to study the mechanism.

5 Kinematic Analysis of the Mechanism

The kinematic analysis is an integral part of the mechanism implementation. It helps us to study the trajectory of points, bodies (objects), and systems of bodies without considering the motion.

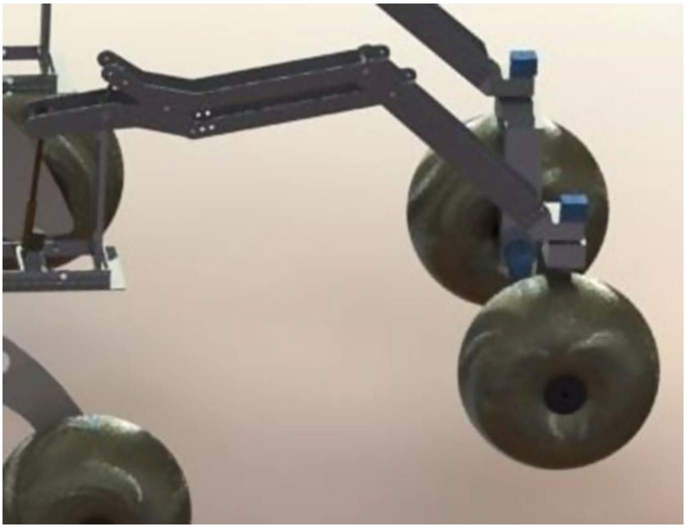


Fig. 14 SolidWorks' rendered image of the mechanism

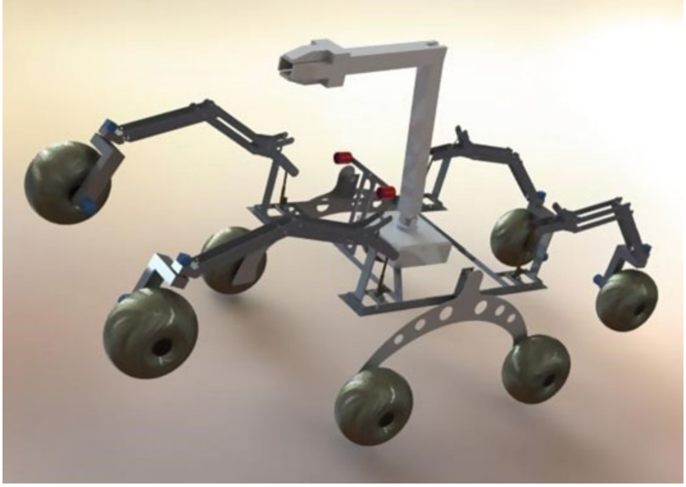


Fig. 15 An isometric view of the overall rover

5.1 Formulation of Velocity Polygon

If a number of bodies are assembled in such a way that the motion of one causes a constrained and predictable motion to others, it is known as a mechanism. The function of a mechanism is to transmit and modify the associated motion.

The leg of the rover moves according to motion which constitutes a planar mechanism. For planer mechanism, the degree of freedom of the mechanism can be calculated by Gruebler's criterion.

$$F = 3(N - 1) - 2j$$

where j = (1/2) (2N2 + 3N3)
where

N = the total number of joints, N2 = the number of binary joints, and N3 = the number of ternary joints.

According to the Grubeler's criterion if for a planar linkage the degree of freedom is one, then it is called a mechanism.

A spring in a mechanism can be converted to a turning pair. This is because the action of a spring is to elongate or shorten as it gets subjected to in tension or compression. A similar variation is obtained by two binary link joined by a turning pair.

The validation of the mechanism is done as shown below.

The mechanism consists of four binary links and two ternary links as shown in Fig. 16.

Hence, the total number of links is six.

$$\begin{aligned} F &= 3(N - 1) - 2J; \\ 2J &= 2 * n_2 + 3 * n_3 \\ &= 2 * 4 + 3 * 2 = 14 \end{aligned}$$

$$\begin{aligned} F &= 3(6 - 1) - 14 \\ &= 1 \end{aligned}$$

Hence, it is a mechanism.

The links move dependent of each other. Hence, the velocities of each link maintain a relation. So once we experimentally find a velocity of a link, we can find

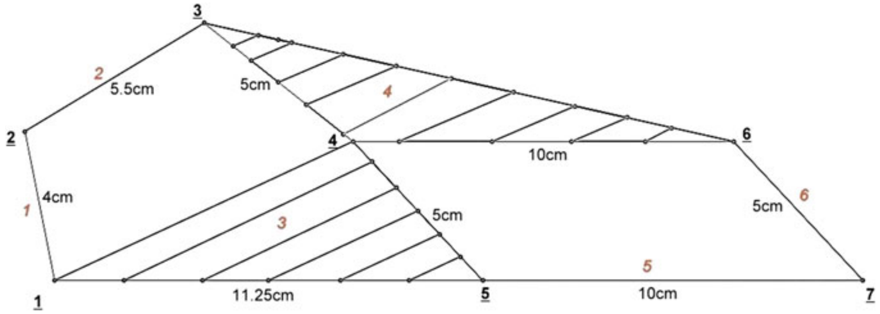


Fig. 16 Mechanism link diagram

out the absolute velocity as well as the relative velocity of other links with respect to reference links. To find out velocities this way, we must follow certain laws and constraints. They include laws like, the velocities of a joint are perpendicular to the direction of motion of a link and that the meeting points of line of action of velocity lines gives the velocity of the links with respect to ground.

Applying this method on the leg mechanism of the rover, the initial velocity given was found to be 7 cm/s. This velocity was given at the wheel of the rover. An assumption was made that the velocity of the wheel will be the same as that of the legs of the rover while traversing the terrain. From the velocity polygon, it was understood that the velocity with which the spring displaces is about one-tenth of the velocity with which the wheel moves up. Figures 17 and 18 show velocity diagram of links 5 and 7, and 6 and 7.

Fig. 17 Velocity diagram of links 5 and 7

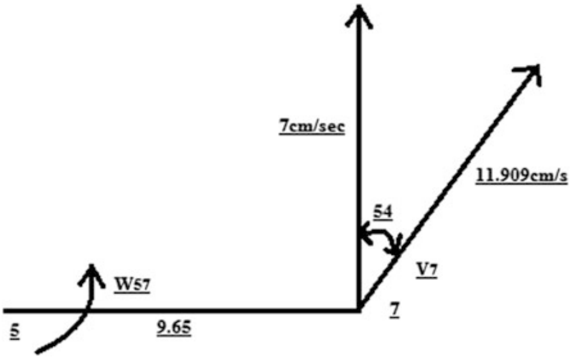
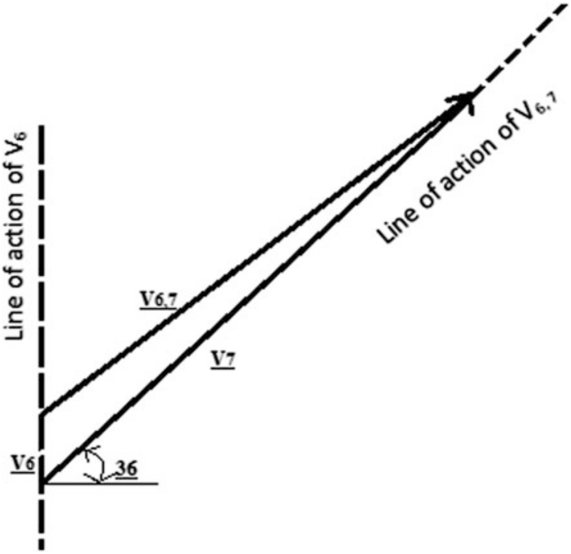


Fig. 18 Velocity diagram links 6 and 7



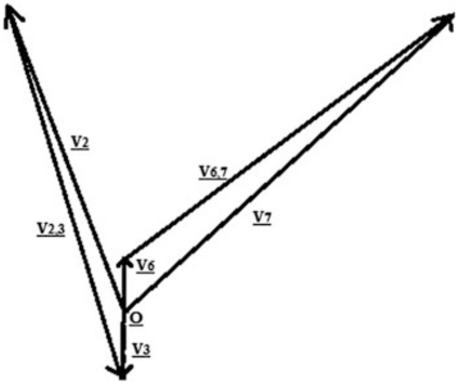


Fig. 19 Velocity polygon diagram

Initially, we assume the velocity of joint 7 to be 7 cm/s. Later, we have to find out the angular velocity of link 57 by the relation $V = r\ \omega$.

$W_{57} = V_7/L_{57}$

$W_{57} = 11.909/19.3 = 0.617$ rad/s

The known velocities are V_7 and the line of action of V_6 and line of action of $V_{6,\ 7}$

$V_6 = 0.8$ cm/s and $V_{6,7} = 11.5$ cm/s

$W_6 = W_{6,7} = (V_{6,7}\ /L_{6,7}) = (11.5/7) = 1.15$ rad/s

$W_3 = V_6/L_{4,6} = (0.8/20) = 0.04$ rad/s

$V_3 = W_3 * L_{3,4} = 0.04 * 10 = 0.4$ cm/s

Now from the known $V_{3,}$ we find out V_2.

$V_2 = 5.7$ cm/s and $V_{3,2} = 6.3$ cm/s

Finally, velocity polygon is drawn showing all the absolute velocities and relative velocities of each link with respect to other links as shown in Fig. 19.

6 Dynamic Analysis Using Adams

The various parameters for the newly developed mechanism are found out with the help of ADAMS software. The parameters that were to be found out include velocity of every joints and links, acceleration of the links and joints, angular velocity, angular acceleration, force and torque acting on various parts, and finally the deformations of the spring, deformation velocity, and the forces that act on the spring. These are important for validating the new design. The basic governing equations that govern the kinematic analysis in ADAMS are based on the Eulerian and Lagrangian dynamics.

The mechanism of the leg of our rover was analyzed in ADAMS, and for the acceleration due to gravity 9.81, with the weight of the members, the graphs were plotted for each link. The results for each part that was analyzed are given below.

6.1 Simulation of the Spider Leg Using ADAMS

The leg mechanism was kinematically analyzed in ADAMS software as shown in Fig. 20 by giving the center of gravity for each specific links and specifically giving the connectors to each link. Afterward, motions were assigned to each of the joints. Later, graphs were plotted when simulated. Some of these are as shown below from Figs. 21, 22, 23, 24, and 25.

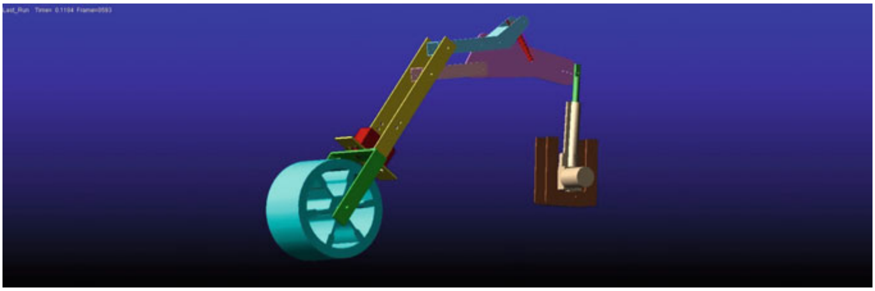


Fig. 20 Simulation in ADAMS

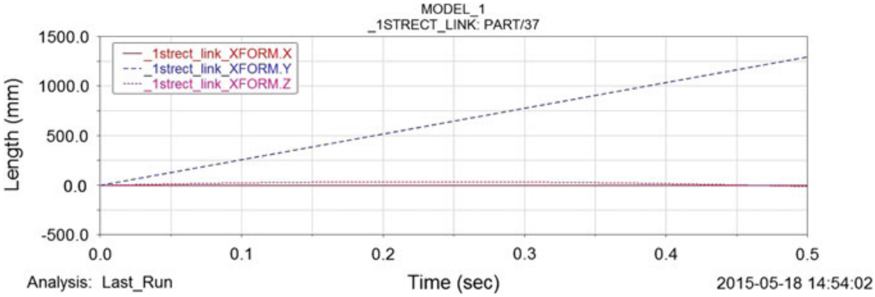


Fig. 21 Length versus time graph of rectangular link 1

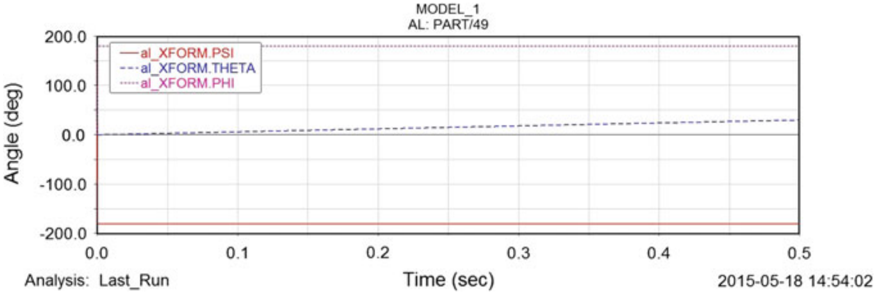


Fig. 22 Angle versus time graph for wheel

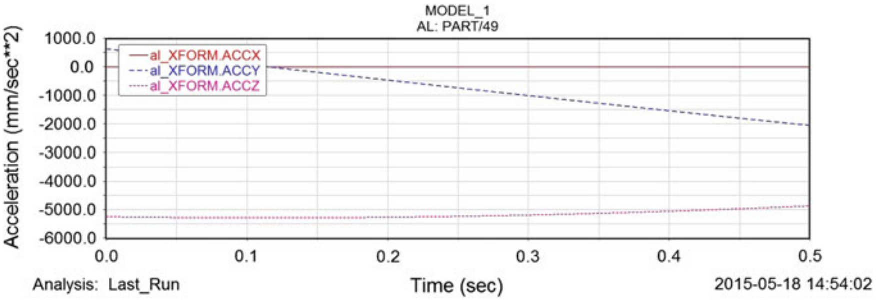


Fig. 23 Acceleration versus time graph for wheel

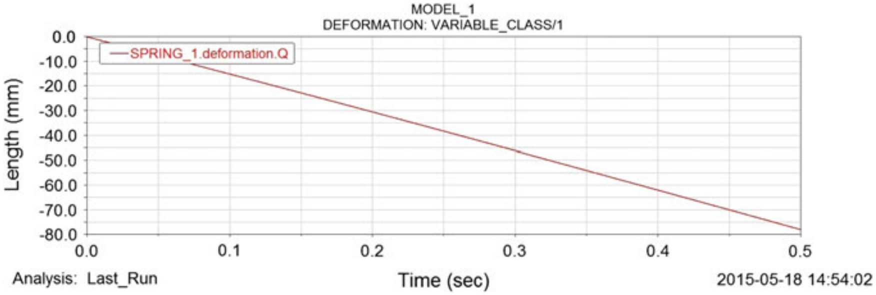


Fig. 24 Deformation versus time graph for spring

7 Linkage Spring Design

The linkage spring has to be carefully designed to support the weight while traversing a terrain in order in turn providing traction. For this, various parameters such as spring rate (spring constant), spring material, and free length have to be considered, and number of coils had to be fixed.

The cycle shock was the best available alternative for the shock absorption and maintaining traction. But the standard available cycle shocks have a very high stiffness rate. A spring has to be custom made for the specific purpose. So in order to do that, a spring had to be selected whose free length and the compressed length are known. Figure 26 shows the dimension of the spring used. But it is to be evaluated whether the spring can support prescribed weight. The load calculation for the spring is as follows.

To calculate the spring rate, we know that
Wire diameter = 4 mm;
Free length = 118 mm;
Compressed length = 80;
No of active coils = 13;

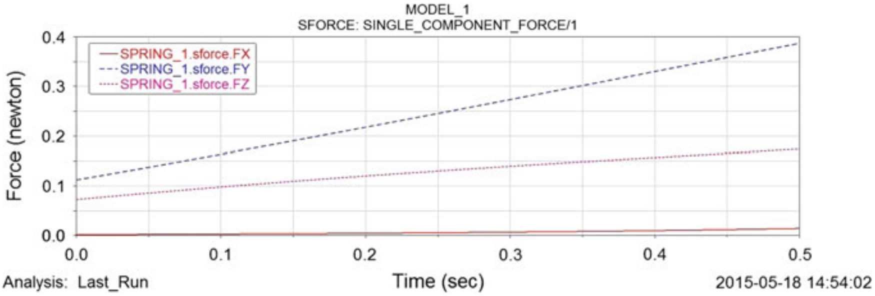


Fig. 25 Force versus time graph for spring

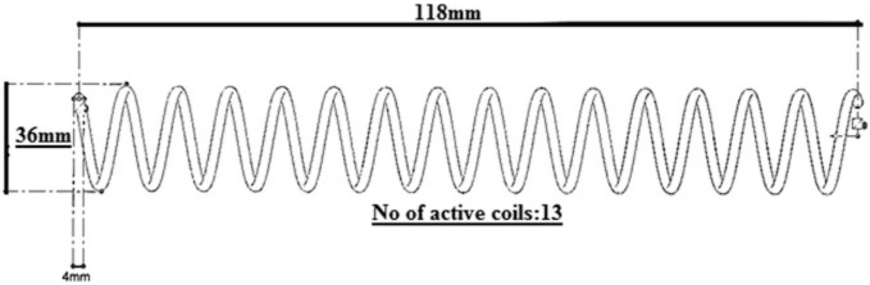


Fig. 26 Spring dimensions

Outer diameter = 36 mm;

$$\begin{aligned}\text{Deflection} &= \text{Free length} - \text{Compressed length}\\ &= 118 - 80 = 38\ \text{mm}\end{aligned}$$

Spring rate = Spring rate/stiffness = (Modulus of spring steel * (wire diameter) ^{4})/ (8 * no of active coils*(mean coil diameter)^3

$$\begin{aligned}\text{Spring Rate} &= \left(\left(209 * 10^9\right) * \left(4 * 10^{-3}\right)^4\right) / (8 * 13 * \left(36 * 10^{-3}\right)^3\\ &= 5957\ \text{N/m}\end{aligned}$$

$$\begin{aligned}\text{Force which can compress } 1'' &= \text{spring rate} * \text{deflection}\\ &= 5957\ \text{Nm} * \left(38 * 10^{-3}\right)\\ &= 226\ \text{N}\end{aligned}$$

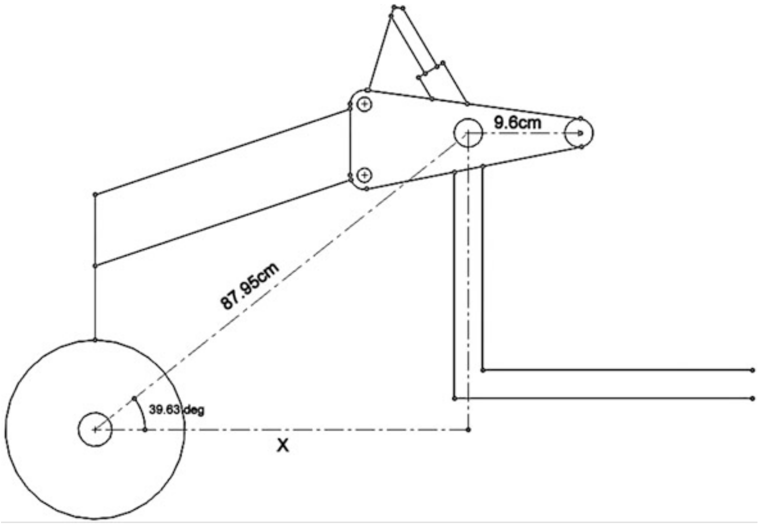
Hence, the load will be approx. 23 kgf required for the spring to compress “1” which is the desired value.

8 Calculation of Height Traversed by the Wheels Using the Mechanism

It is necessary that the calculation of height traversed by the wheel should be done for analyzing the mechanism. Here, the height traversed by the wheel is calculated when the actuator moves down 1 cm. The following figure shows the structural dimensions of the mechanism. Using this, the maximum height traversed when the actuator is at its maximum limit can also be found out.

Under a load of 200 lbf (90.7 kgf), the actuator speed is 0.90 in/s.

Under a load of 600 lbf (272 kgf), the actuator speed is 0.39 in/s.



$$\theta = 39.63^\circ$$

$$X = 87.95 \cos 39.63 = 68.34 \text{ cm}$$

$$\begin{aligned}\text{Ratio maintained} &= 68.34/9.61\\ &= 7.11134\end{aligned}$$

Hence, when the linear actuator moves 1 cm, the wheel moves 7.11 cm from the ground.

For a load of 200 lbf, the velocity of the wheel at x cm/s = ratio maintained * velocity of the linear actuator in cm/s. For a load of 600 lbf,

$$\begin{aligned}\text{The velocity of the wheel at } 0.39\ \text{in}/\text{s}(0.99\ \text{cm}/\text{s}) &= 7.1134 * 0.99\ \text{cm}/\text{s}\\ &= 7.04\ \text{cm}/\text{s}\\ \text{The velocity of the wheel at } 0.9\ \text{in}/\text{s}(2.286\ \text{cm}/\text{s}) &= 7.1134 * 2.286\ \text{cm}/\text{s}\end{aligned}$$

9 Weight Distribution

The weight distribution is the apportioning of weight within a vehicle. Typically, it is written in the form x: y where x is the percentage in front and y is the percentage in the back. In a vehicle which relies on gravity in some way, weight distribution directly affects a variety of the vehicle characteristics including handling, acceleration, and traction. For this reason, weight distribution varies with vehicle's intended usage. The following section shows how the weight is distributed when the Mars rover is in different positions.

9.1 The Rover Is in Normal Position

Figure 27 shows the rover in normal position with X = 78 cm, Y = 396 cm, and Z = 714 cm.

The calculation of weight distribution is as follows:

$$\text{Overall length} = 230\ \text{cm}$$

When four wheels are in ground, the weight distribution is

50: 50; that is, the center of gravity lies in the center as shown in Fig. 28. The vehicle is stable as the CG lies in the center of the rover.

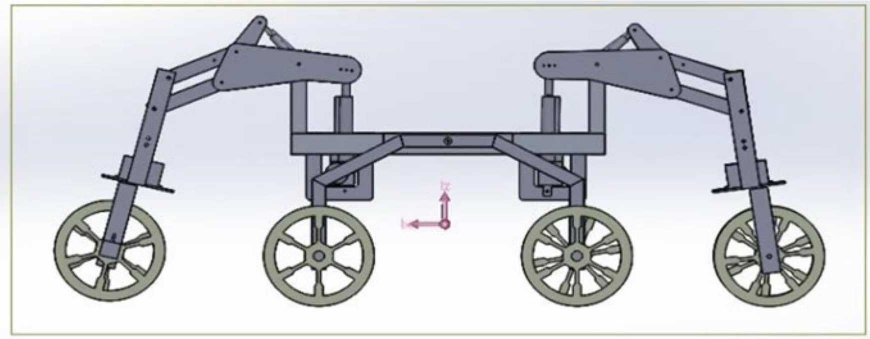


Fig. 27 Normal position

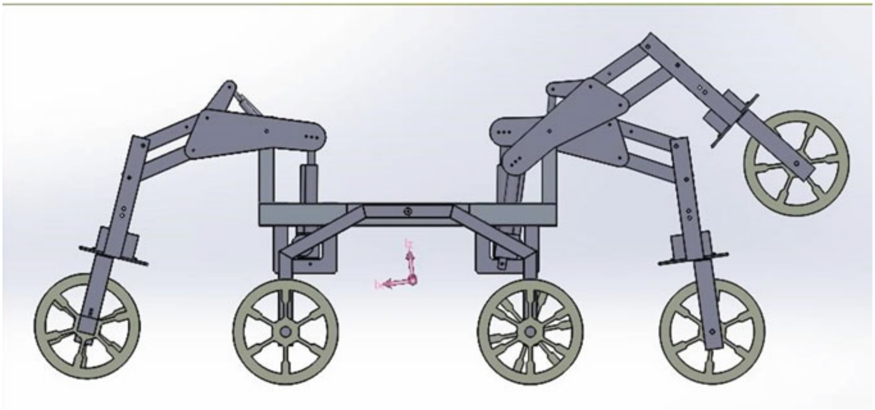


Fig. 28 Rover's one leg lifted position

9.2 *The Rover's One Leg Is in the Lifted Position*

Figure 28 shows the rover in one leg lifted position with $X = 78$ cm, $Y = 400$ cm, and $Z = 711$ cm.

The calculation of weight distribution is as follows:

Shift in length = 4 cm.

Ratio = 230/(115 + 4) = 1.932.

Overall weight distribution = 25 * 1.932 = 48.3.

Hence, the weight distribution is 51.7: 48.3.

9.3 *The Rover's Both Legs Are Lifted*

Figure 29 shows the rover in both legs' lifted position with $X = 78$ cm, $Y = 401$ cm, and $Z = 711$ cm.

The calculation of weight distribution is as follows:

Shift in length = 5 cm.

Ratio = 230/(115 + 5) = 1.9166.

Overall weight distribution = 25 * 1.9166 = 47.915.

Hence, the weight distribution is 52.1: 47.9.

The weight distribution was well optimized when compared to the previous versions. In the final model, there is a significant difference in the weight distribution and in the CG value. After optimization, the CG shift is very low which makes the rover stable with the proper working of the mechanism. Figure 30 below shows the final prototype of the Mars rover.

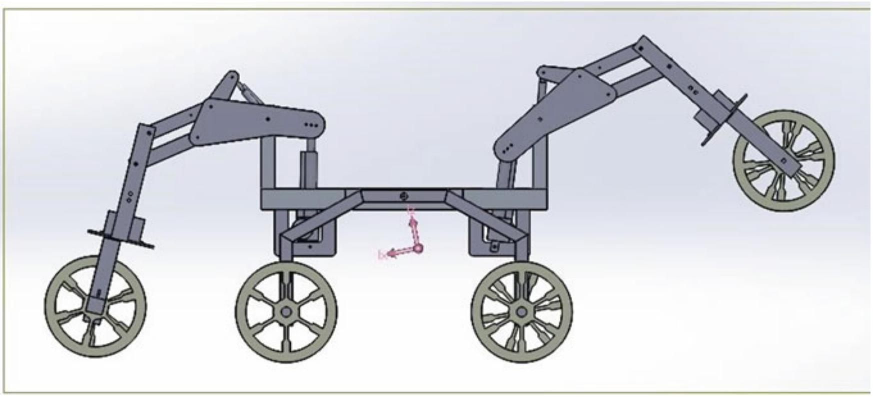


Fig. 29 Rover's both the legs are lifted

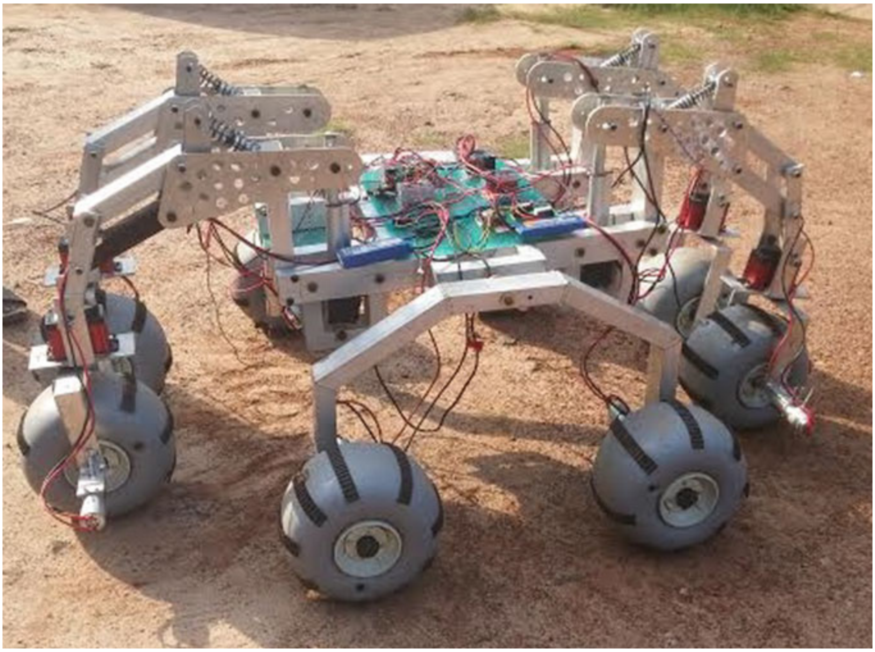


Fig. 30 Final prototype

10 Conclusion and Future Works

The Mars rover has been designed with an improved mechanism when compared to existing versions, and the final prototype has been built. The rover was built with aluminum 6061 and using various mechanical techniques such as water jet cutting, TIG welding, and drilling. The team's design had cleared the Critical Design Review of the University Rover Challenge 2015 conducted by Mars Society, USA. An innovative spider leg mechanism has been implemented in which suitable kinematic linkage analysis and dynamic analysis were conducted to analyze and verify whether it is capable of traversing all terrains. The prototype manufacturing helped us to experimentally test the mechanism, so that we could bring necessary changes to the design. The present mechanism enables the vehicle to traverse up steps/rocks while still being able to take the impacts of varying terrain. In currently used designs, the height which the wheels of a vehicle can traverse is limited, and it is extremely difficult for these vehicles to climb up high steps. However, a suspension mechanism with very soft shocks can cover rough and rocky terrain to an extent. Unfortunately, it is impossible for them to climb up huge steps and rocks. Study and analysis have been done on the design, and finally, an optimized design is obtained. The final design is again tested experimentally using a fully functional prototype.

Acknowledgments The authors would like to thank Mechatronics and Intelligence Systems Research Laboratory, Department of Mechanical Engineering, Amrita University, Amritapuri campus, for providing support to carry out the research and experiments.

References

1. www.wikipedia.org/wiki/Curiosity_(rover)
2. www.wikipedia.org/wiki/Mars_Pathfinder
3. www.wikipedia.org/wiki/Opportunity_(rover)
4. Aswath, S., Ajithkumar, N., Tilak, C.K., Saboo, N., Suresh, A., Kamalapuram, R., Mattathil, A., Anirudh, H., Krishnan, A.B., Udupa, G.: An Intelligent Rover Design Integrated with Humanoid Robot for Alien Planet Exploration, in series. Advances in Intelligent Systems and Computing of springer, vol. 345, pp. 441–457
5. Bickler, D.B.: US 4840394, Articulated suspension system. published 1988-04-21, issued 1989-06-20, assigned to NASA
6. Miller, D.P., Lee, T.L.: High-speed traversal of rough terrain using a rocker-bogie mobility system
7. Makovsky, A., Ilott, P., Taylor, J.: Mars Science Laboratory Telecommunications System Design (PDF). Pasadena, California: Jet Propulsion Laboratory (2009). Retrieved 7 on Aug 2012
8. NASA Patent Abstracts Bibliography, Section 1. Abstracts (pdf). June 1990. p. 19. ARTICULATED SUSPENSION SYSTEM
9. Lindemann, R., Voorhees, C.J.: Mars Exploration Rover mobility assembly design, test and performance. In: 2005 IEEE International Conference on Systems, Man and Cybernetics. vol. 1. IEEE (2005)

10. Harrington, B.D., Voorhees, C.J.: The Challenges of designing the rocker-bogie suspension for the mars exploration rover. In: Aerospace Mechanisms Symposium, pp. 19–21
11. Bickler, D.: Roving over Mars. Mechanical Engineering Magazine. American Society of Mechanical Engineers (1998)
12. Patel, N., Slade, R., Clemmet, J.: The ExoMars rover locomotion subsystem. J. Terrramech. **47** (4), 227–242 (2010)
13. www.aluminiumsupplier.com.cn/News

System of 3-D Printed Components for the Rapid Prototyping of Legged Robots

David Claveau

Abstract A system of components designed for the rapid prototyping of different robot body plans for legged robots is described in this chapter. The components are designed and fabricated using 3-D printing technology. As such, they constitute a low-cost way to experiment with legged robots for researchers, practitioners, and students. As legged robots become more common, there will be a need for new designs and new intelligent behaviors. A system of components that facilitates experimentation can play an important role in this development. And because these components are 3-D printed, they can easily be shared and fabricated by others.

1 Introduction

Legged robots are becoming more common both inside and outside of the laboratory. Anyone who has designed and built a legged robot knows how difficult it can be to experiment with its physical form or body plan. Usually, a fixed body plan is used for a particular application and the control hardware and software are designed to work with that body. It may be difficult to know whether a particular body plan is well-suited to a task/environment without first experimenting with some options. This is particularly true for legged robots because they have a very large number of body plan parameters. These include number of limbs, number of joints per limb, length of limbs, width, and height of body. The exploration and development of new behaviors for legged robots would be greatly aided by the ability to rapidly prototype different designs. Simulation can be used to explore these options, but accurate and useful simulations of legged robots are not easy to achieve. They can be misleading and do not give the designer the physical intuition obtainable from a working prototype.

Legged robots often consist of custom designed brackets and interfaces that make it difficult to rapidly prototype new ideas. Instead, it would be convenient to

D. Claveau (✉)
CSU Channel Islands, Camarillo, CA, USA
e-mail: david.claveau@csuci.edu

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_35

have a set of building blocks or standard components that could be used to connect actuators and sensors together in different ways. Recent advances in additive manufacturing or 3-D printing have made it possible to take steps toward creating such a modular approach to legged robots. This paper describes a system of components that have been designed and printed at a very low cost and which could be shared and replicated in any laboratory. The goal has been to rapidly prototype different body plans or morphologies without consideration for the special demands of any particular application. As such, concerns about power budgets, payload, and torque ranges are not addressed. The focus is on enabling quick experimentation with different robot bodies to explore and develop new intelligent behaviors with lightweight low-cost robots, benefiting designers and especially students. The rest of this paper describes this effort.

2 Background and Related Work

Some form of modularity has always been important to robotics. Early manipulators had end effector tools that could be quickly swapped out. In the 1980s, Fukuda and Nakagawa [1] created some of the first self-reconfigurable robots composed of components with standard interfaces. This idea was further developed by Yim et al. [2] and others in the 1990s, creating robots composed of chains of uniform blocks. These systems are designed for online self-reconfigurability. The approach described in this paper is only for off-line, manual reconfiguration. An example of this is presented in Haynes et al. [3], wherein a hexapod and a tetrapod are built from the same system-level components. Their system takes more design parameters into account than the one presented here; however, it appears to not be designed for rapid or easy experimentation.

The approach in this paper is based on the design and fabrication of components using additive manufacturing or 3-D printing. Additive manufacturing technologies began appearing in the 1980s [4]. An object is first drawn using a computer-aided-design (CAD) tool. A standard description or model of the object can then be used by a 3-D printer to fabricate the object by dispensing successive layers of molten plastic from a robotic extruder nozzle. The plastic hardens immediately after extrusion from the nozzle. Variants of additive manufacturing were used for robotics applications as early as 1992 [5]. Today, it is possible to find complete models of small humanoid robots for hobbyists, available on the Internet and ready to be printed [6].

Simulation has been used by others to explore different body plans. Marbach and Ijspeert [7] used a simulator and a genetic algorithm to co-evolve the body plan and the control of a legged robot. Lund [8] also used a simulator but used Lego Mindstorms to physically assemble the robots and attempted to achieve modular building blocks with processing power.

3 Design and Implementation of the Components

The goal of this exploratory project was to support the rapid prototyping of lightweight legged robots that would not be carrying any significant payload. The system of components was designed to satisfy a basic set of requirements. The components should be as follows:

- few in number of types,
- composable,
- lightweight, and
- low cost.

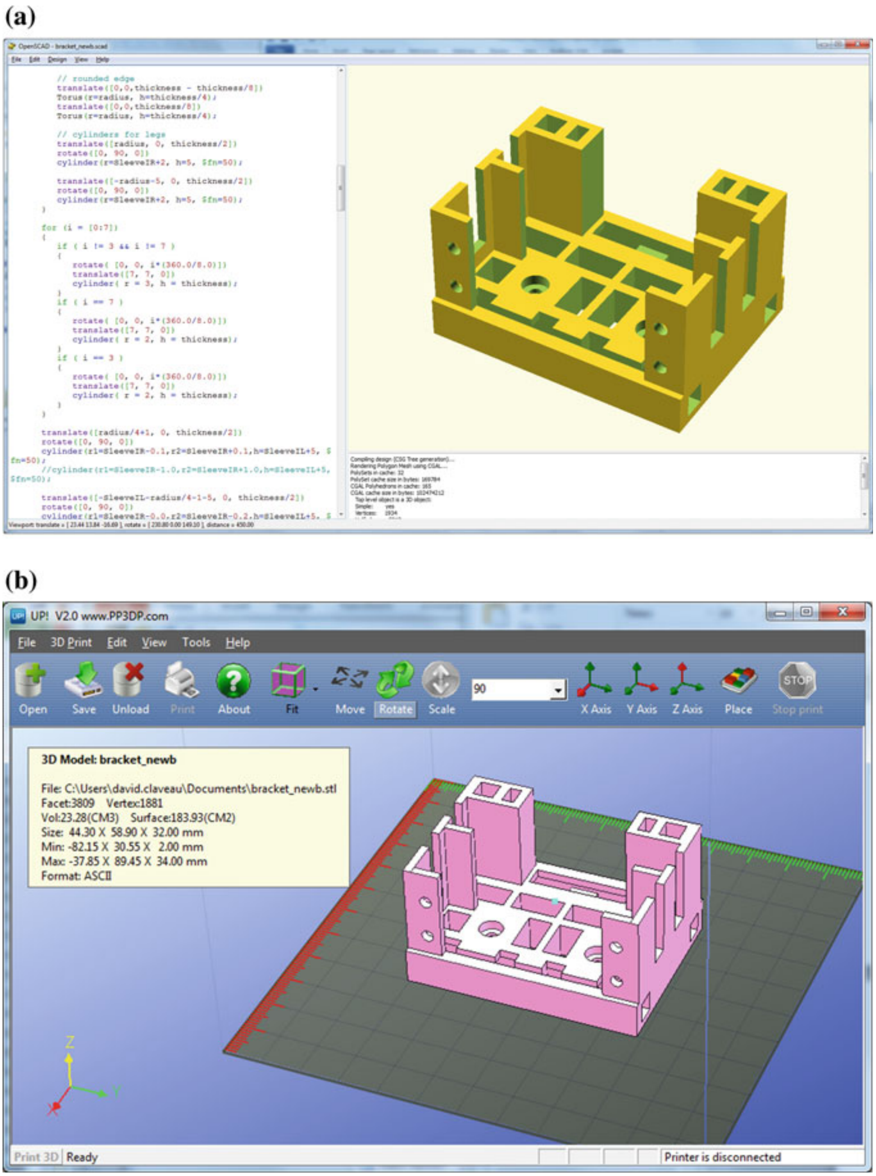
The first two requirements are related in that a single modular component that supports easy interfacing and composability can then be used to build most of the robot's body. The last two are also related in the case of additive manufacturing since a lightweight component is one that is made from less material and so takes less time to make. The initial design consists of three types of components: a main bracket for actuators, a servo arm for limbs, and a simple foot. The design and fabrication of each of these is described in the following subsections.

3.1 Main Bracket

The legs of a robot typically consist of rigid segments connected at joints. Each joint has an actuator such as a servo motor for each degree of freedom (DOF). A single motor can be used for an elbow or knee joint while two or three are used for a hip or ankle joint. What is needed is a bracket that can hold a motor and can also be attached to other motors and to structural members that form limbs and body frames.

To design such a component, the first step was to obtain the measurements of a common off-the-shelf servo motor that would serve as a good test actuator. Then, a model of a simple bracket could be created in a CAD program. The freely available OpenSCAD [9] was used for the modeling. It supports the use of constructive solid geometry through an interpreted scripting language that is easy to learn and use. In fact, it is possible to find models in the OpenSCAD format on the Internet that can then be modified. The model can be viewed in the program and Fig. 1a shows an example of the OpenSCAD interface with the script for the component on the left and a graphical rendering of the component on the right.

The model is opened in a utility program that converts it into the commands that will drive the 3-D printer. Figure 1b shows the interface of this program. The part will take 120 min to print and will use 31.6 g of ABS plastic. After several iterations of printing, the finished component appears as shown in Fig. 2. The component on the left is empty and on the right a motor is held by the bracket. There are small holes to mount the motor, but the fit was designed to be snug so that the


(a)
OpenSCAD - bracket_newb.scad
File Edit Design View Help
(b)
UP! V2.0 www.PP3DP.com
File 3D Print Edit View Tools Help
Open Save Unload Print About Fit Move Rotate Scale 90 X Axis Y Axis Z Axis Place Stop print
Z
Y
X
Print 3D Ready
Printer is disconnected

```
        // rounded edge
        translate([0,0,thickness - thickness/8])
        Torus(r=radius, h=thickness/4);
        translate([0,0,thickness/8])
        Torus(r=radius, h=thickness/4);

        // cylinders for legs
        translate([radius, 0, thickness/2])
        rotate([0, 90, 0])
        cylinder(r=SleeveIR+2, h=5, $fn=50);

        translate([-radius-5, 0, thickness/2])
        rotate([0, 90, 0])
        cylinder(r=SleeveIR+2, h=5, $fn=50);
    }

    for (i = [0:7])
    {
        if ( i != 3 && i != 7 )
        {
            rotate( [0, 0, i*(360.0/8.0)])
            translate([7, 7, 0])
            cylinder( r = 3, h = thickness);
        }
        if ( i == 7 )
        {
            rotate( [0, 0, i*(360.0/8.0)])
            translate([7, 7, 0])
            cylinder( r = 2, h = thickness);
        }
        if ( i == 3 )
        {
            rotate( [0, 0, i*(360.0/8.0)])
            translate([7, 7, 0])
            cylinder( r = 2, h = thickness);
        }
    }

    translate([radius/4+1, 0, thickness/2])
    rotate([0, 90, 0])
    cylinder(r1=SleeveIR-0.1,r2=SleeveIR+0.1,h=SleeveIL+5, $
fn=50);
    //cylinder(r1=SleeveIR-1.0,r2=SleeveIR+1.0,h=SleeveIL+5,
$fn=50);

    translate([-SleeveIL-radius/4-1-5, 0, thickness/2])
    rotate([0, 90, 0])
    cylinder(r1=SleeveIR-0.0,r2=SleeveIR-0.2,h=SleeveIL+5, $
```

```
Compiling design (CSG Tree generation)...
Rendering Polygon Mesh using CGAL...
PolySets in cache: 32
PolySet cache size in bytes: 169784
CGAL Polyhedrons in cache: 165
CGAL cache size in bytes: 10247412
   Top level object is a 3D object:
   Simple:     yes
   Vertices:   1934
```

Viewport: translate = [23.44 13.84 -16.69], rotate = [230.80 0.00 149.10], distance = 450.00

3D Model: bracket_newb

File: C:\Users\david.claveau\Documents\bracket_newb.stl
Facet:3809 Vertex:1881
Vol:23.28(CM3) Surface:183.93(CM2)
Size: 44.30 X 58.90 X 32.00 mm
Min: -82.15 X 30.55 X 2.00 mm
Max: -37.85 X 89.45 X 34.00 mm
Format: ASCII

Fig. 1 The OpenSCAD program in (**a**) and the 3-D printer application in (**b**), both showing the model of the main bracket

motor would remain in the bracket unless forcibly removed by a person. There are slots at the corners to receive structural members for legs and the body frame. An example of how these might be used is shown in Fig. 3. There a 2-DOF ‘hip’ joint is shown. There is a female spline on the broad side of the bracket that is used to

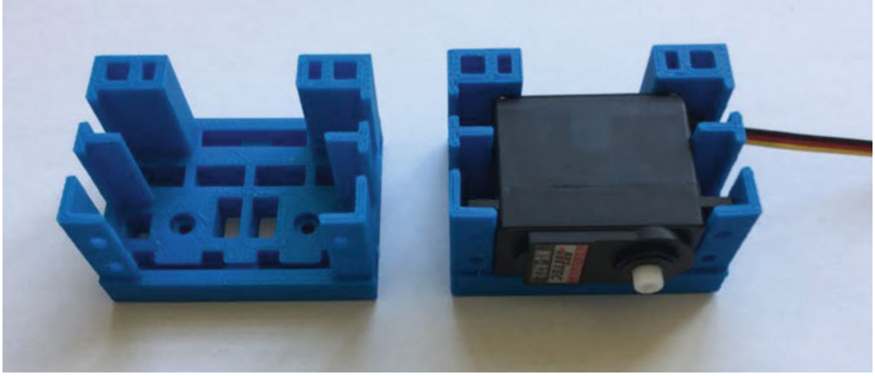


Fig. 2 Two main brackets printed in ABS plastic. The one on the *left* is empty and the one on the *right* is holding a servo motor

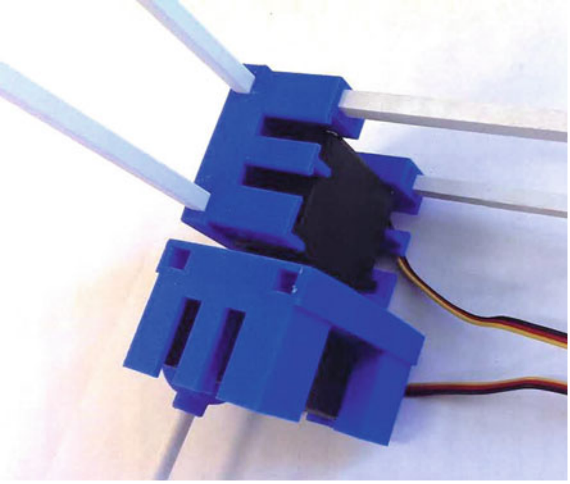


Fig. 3 An example of how the main bracket is used as a structural element to build a robot body. The bracket has slots that can receive rigid or compliant members made from a variety of materials

attach to the other servo motor in the lower bracket. All components were printed on a consumer-grade desktop 3-D printer from PP3DP [10] using ABS plastic.

3.2 *Servo Arm*

Each joint needs a way to attach a limb to the drive gear or spline of the motor. For this purpose, a servo arm was designed. The CAD model is shown in Fig. 4a and the printed component is shown in Fig. 4b as it would be attached to a motor.

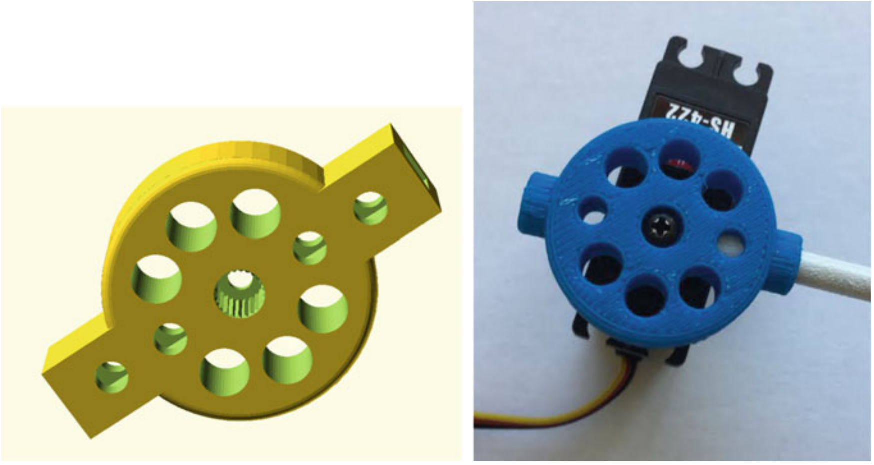


Fig. 4 On the *left* is the model of the servo arm and on the *right* is a printed component attached to a servo motor

The CAD model shows the details of the female spline and an inner rim to allow a screw to be used to attach it to the servo spline. There are two sleeves to hold rigid structural members. The model in the figure shows rectangular sleeves that are used to keep the member from rotating. Each sleeve has two holes that can be used to secure the member with a screw; however, the sleeves have been sized to hold snugly onto some off-the-shelf structural members. Different materials can be used of course, with aluminum giving more rigidity and PVC plastic giving more compliance. The flat round portion of the component gives it strength and the added holes allow for easy mounting of sensors while making the component lighter and less expensive to print. It required 7.6 g of ABS plastic and 30 min to print.

3.3 Foot

A specialized foot was designed for lightweight prototype robots that need to keep their feet flat on the ground to receive a contact signal from a pressure sensor. The foot was based on a ball-and-socket joint that was originally intended for an office chair. The complete assembly is shown in Fig. 5. It had to be printed in two parts and then assembled by pressing the shaft portion onto the ball. It required 15.7 g of ABS plastic and 61 min to print. The base of the foot is designed to hold a flap (shown in white ABS in (b)) that keeps a pressure sensor in place. The removed flap and sensor are shown in Fig. 5c.

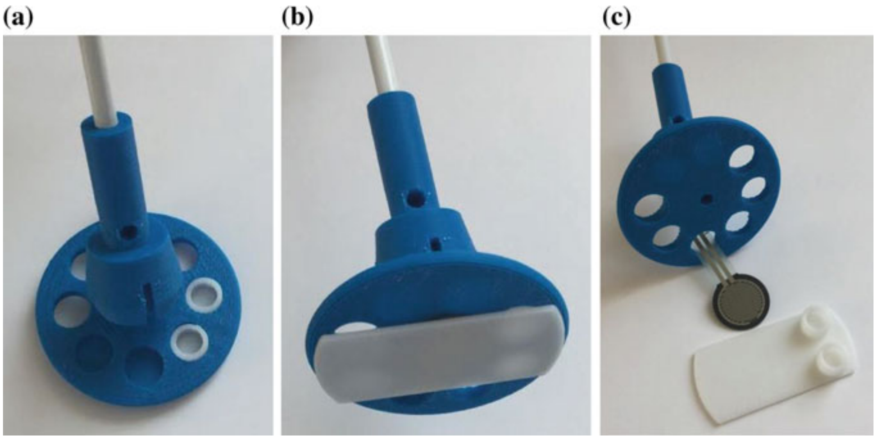


Fig. 5 The complete foot shown in (**a**) and its underside in (**b**). The white flap was designed to hold a pressure sensor as shown in (**c**)

4 Example Legged Robots

The components allow a designer to quickly build a physical prototype by hand without the need for special tools or skills. Two examples are shown in Fig. 6. Both were assembled in only a few minutes. The robot in (a) has a tall and narrow body and has long legs that have both hip and knee joints. The robot in (b) is short and wide and has simple 1-DOF legs. The robot in Fig. 7 has an Arduino controller

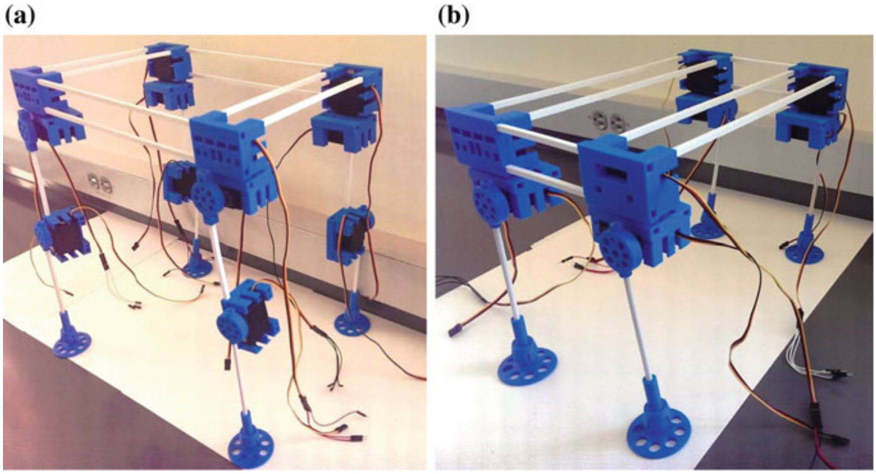


Fig. 6 Two example robots made with the components. The robot in (**a**) is tall and narrow and has legs with knee joints. The one in (**b**) is short and wide and has 1-DOF legs

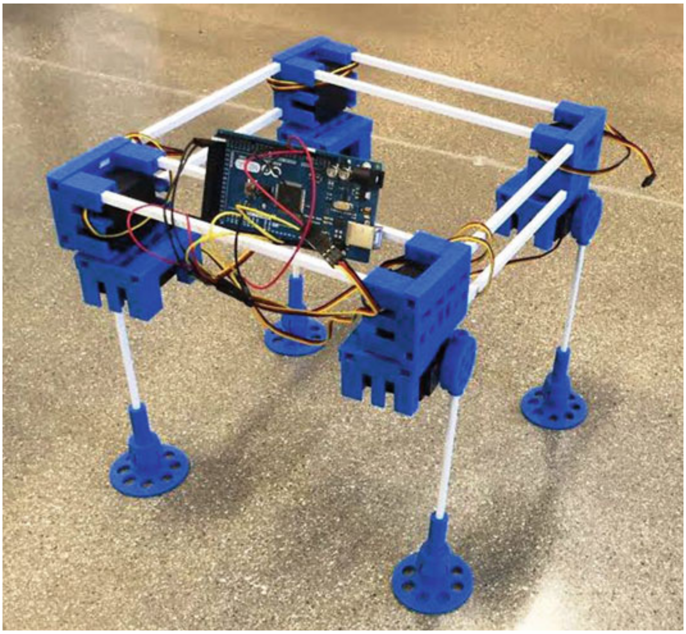


Fig. 7 A similar robot to the one in Fig. 6b with an Arduino controller

connected to its motors. The robot was powered with rechargeable 9-V batteries. A video of our experiments with the robot is available on our Web site [11].

5 Conclusion

Legged robots are increasingly important because they can perform a wide range of useful tasks in challenging environments. Experimenting with the body plans of these robots can be difficult and time consuming because of the many parameters involved. This paper has described a system of components that allow a designer to easily combine sensors and actuators to form different body plans. The design and fabrication of the components themselves depended on the recent availability of low-cost 3-D printing and associated CAD software. The ability to rapidly prototype different robot designs is important to the development of new behaviors and the study of their performance.

The components are fabricated on a low-cost 3-D printer and are made from ABS plastic. As such they are easy to work with and are suitable for educational environments in which students can learn by physically constructing a variety of legged robots and experimenting with them. This first attempt suggests that this is a promising approach that will benefit from improvements in 3-D printing. Digital design sharing will become more common in robot design at all levels.

References

1. Fukuda, T., Nakagawa, S.: Dynamic reconfigurable robotic system. In: Proceedings of IEEE Conference on Robotics and Automation, vol. 3, pp. 1581–1586 (1988)
2. Yim, M., Shen, W., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., Chirikjian, G.: Modular self-reconfigurable robot systems [Grand Challenges of Robotics]. IEEE Robot. Autom. Mag. **14**(1), 865–872 (2007)
3. Haynes, G.C., Pusey, J., Knopf, R., Johnson, A.M., Koditschek, D.E.: Laboratory on legs: an architecture for adjustable morphology with legged robots. In: Unmanned Systems Technology XIV, vol. 8387, no. 1. SPIE, p. 83870W (2012)
4. Apparatus for Production of Three-Dimensional Objects by Stereolithography (8 August 1984)
5. Beck, J.E., Fritz, B., Siewiorek, D., Weiss, L.: Manufacturing Mechatronics Using Thermal Spray Shape Deposition. In: Proceedings of the 1992 Solid Freeform Fabrication Symposium (1992). Accessed 20 Dec 2014
6. http://www.intorobotics.com/lesson-building-custom-3d-printed-humanoid-robot/
7. Marbach, D., Ijspeert, A.J.: Co-evolution of configuration and control for homogenous modular robots. In: Groen, F. et al. (eds) Proceedings of the Eighth Conference on Intelligent Autonomous Systems (IAS8). IOS Press (2004)
8. Lund, H.H.: Co-evolving control and morphology with LEGO robots. In: Hara, Pfeifer (eds) Morphofunctional Machines, Springer, Heidelberg (2002)
9. http://www.openscad.org/
10. http://up3d.com/
11. http://faculty.csuci.edu/David.Claveau/lab/legs/walking.html

Part III
Applications of Robot Intelligence Technology

A Rigid and Soft Combined Robot that Is Designed to Be Used in Confined Spaces

Xingsong Wang, Mengqian Tian, Jianfei Chen, Xiang Li and Yu Xiao

Abstract This article describes a rigid and soft combined robot that is designed to be used in confined spaces. The robot is based on pneumatically actuated peristaltic motion. This motion is accomplished by the use of a pair of pneumatic annular grippers in concert with a novel pneumatic soft actuator. The actuator is capable of actuation and it is made of soft material. So it can passively adapt to the outside passage, with this characteristic the robot can navigate through some constrained environments that would otherwise be inaccessible with conventional rigid actuators. The annular grippers consist of six scissor structures wrapped around the cylinder which can dramatically increase the maximum diameter of the grippers. So the robot can adapt its shape to external constraints and obstacles. The robot also contains a 2-DOF pneumatic soft joint that promises the robot capable of changing its direction actively.

Keywords Soft robot · Soft actuator · Annular gripper · Confined spaces

1 Introduction

Conventional robots and machines are made of rigid materials. With decades of development, they are now incredibly powerful and precise, and these robots are typically used in well-defined environments in which they repetitively perform a prescribed motion with great precision. However, their actuation systems are usually complex and relatively huge, and what is more, rigid materials limit their ability to elastically deform and adapt their shape to external constraints and obstacles. This makes them powerless in field exploration and disaster relief that will often encounter easily deformable surfaces such as sand, mud, and soft soil [1].

Recently, soft robotics is emerging as a new field of engineering. Soft robots are usually made of easily deformable matters such as silicon rubber and fiber and

X. Wang (✉) · M. Tian · J. Chen · X. Li · Y. Xiao
School of Mechanical Engineering, Southeast University, Nanjing 211189, Jiangsu, China
e-mail: xswang@seu.edu.cn

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_36

contain little or no rigid components like the multigait soft robot developed by Shepherd et al. [2] and fiber-reinforced actuators developed by Maeder-York et al. [3] and Galloway et al. [4]. The appearance of soft robotics creates new opportunities in many fields and challenges traditional engineering thinking [5]. With the application of easily deformable matters, soft robots can deform itself in large scale and provide theoretically an infinite number of degrees of freedom. By using soft materials, the soft robots usually contains high dexterity which makes them ideal for applications such as healthcare robots that interact with humans without causing injury, defense and rescue robots that need to reach some confined spaces and operate in cluttered environments [6]. But the soft robots whose structure is composed entirely of soft materials also have some disadvantages. They usually move slowly and unable to provide too much force.

In order to develop a robot capable of searching for natural disaster relief, both soft and rigid components are integrated into the design and combined some of their advantages. The robot consists of a pair of annular grippers made of rigid materials, a nonlinear actuator, and bending joint that both are made of silicon rubber and Kevlar fiber. The whole robot is driven by compressed air and based on peristaltic motion.

2 Design of the Hybrid Robot

Traditional rescue robots are usually based on wheels, tracks, or legs and require precise sensing and fine motor control. Complex drive system makes the robots relatively large in scale, which limits their ability to operate in unstructured or tightly constrained environments. Organism like earthworm possess high dexterity and can enter into almost any place. Inspired from the earthworm, the robot proposed in this paper is based on peristaltic motion and driven by compressed air. This motion is accomplished by the use of a pair of annular grippers in concert with a soft actuator.

Figure 1 shows the peristaltic motion of the robot, and it can move forward by conducting the following eight steps:

1. The robot expands the posterior gripper until it grips on the outside environment firmly.
2. The soft linear actuator elongates.
3. The anterior gripper expands until it grips on the outside environment firmly.
4. Deflate the actuator that drives the posterior gripper, and it contracts to the original phase.
5. Deflate the soft linear actuator, and it contracts to the original state.
6. Expand the posterior gripper again.
7. Release the anterior gripper.
8. Return to Step 2.

Figure 2 shows the components of the robot.

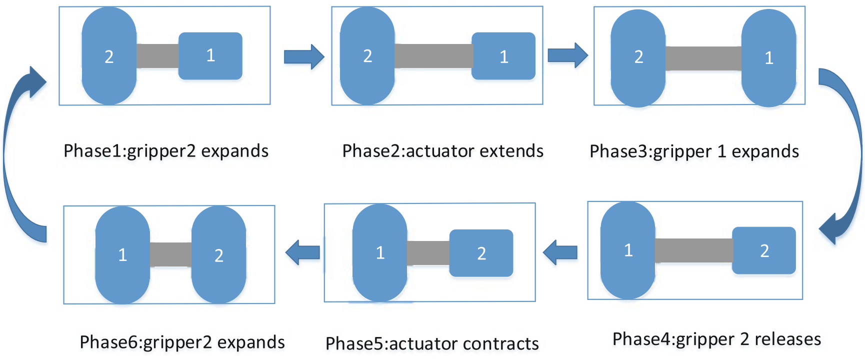


Fig. 1 Peristaltic motion

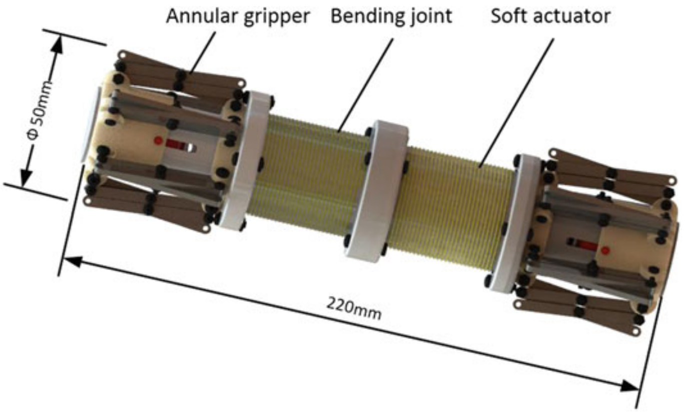


Fig. 2 Schematic design of the robot

2.1 Material Selection and Test

In this study, silicon rubber (Ecoflex 0030, Dragon Skin 30, Smooth-on, USA) was selected to fabricate the body of the actuator. These two materials are platinum-catalyzed silicones that need to be mixed in the ratio of 1A:1B by weight or volume and can cure at room temperature with negligible shrinkage. Ecoflex 0030 is a supersoft silicone, while the Dragon Skin 30 is relatively strong and stretchy. We mixed these two materials in the ratio of 1:1 to meet our requirement for utilization.

Tensile testing was conducted to obtain the strain–stress relationship of the mixed silicone rubber. The dimension of the specimens is 2 × 50 × 10. Figure 3 shows the test results for two specimens. The material exhibits a nearly linear strain–stress relationship up to 250 % strain, which is in the range of the

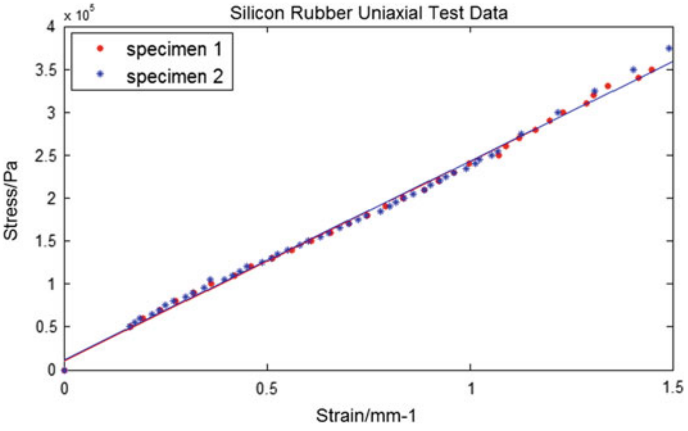


Fig. 3 Strain–stress testing results of two specimens

deformation required for the actuator, with approximately linear Young's modulus 0.23 MPa [7].

2.2 Design of the Soft Actuator

The soft actuator in the middle is the point that the robot can elongate and contract, thus makes the robot advance. As is depicted in Fig. 4, the actuator consists of a hollow cylinder, Kevlar fiber, and caps. The hollow cylinder is made of silicon rubber, and there is a high-performance Kevlar fiber densely and evenly wrapped around the cylinder. The rigid caps on both ends are designed to prevent the silicon

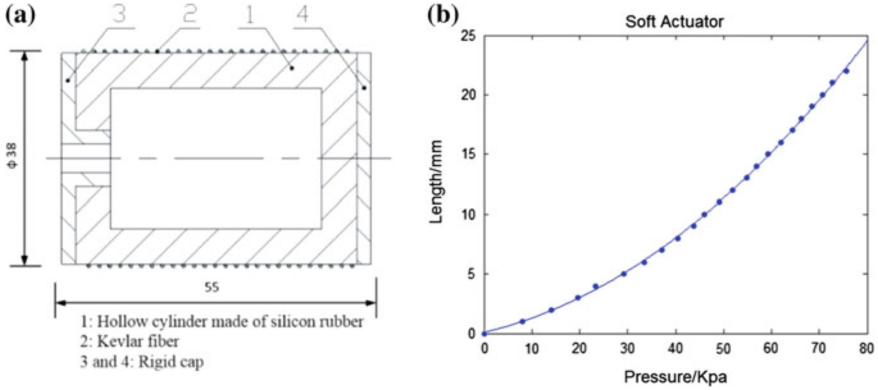


Fig. 4 **a** Cross-sectional view of the soft actuator and **b** the relation between the pressure and the extension

rubber protrude when the hollow cylinder is supplied with compressed air. The silicon rubber is a material that can elastically deform in large scale, and on supplying air, the rubber expands in both radial and axial directions, but the Kevlar fiber limits its expansion in the radial direction. As a result, the soft actuator can only expand in the axial direction. Once the soft actuator deflates, it will contract to its original state.

Figure 3b shows the relation between the pressure and the extension.

2.3 Design of the 2-DOF Bending Joint

The passages that formed after disasters are not always straight and usually contains severals turns. Except for the ability to passively adapt to the cluttered environment, the robot should also capable of changing its direction actively. A 2-DOF soft bending joint is designed to make sure that the robot can navigate through passages with turns. As is depicted in Fig. 5, the bending joint is a cylinder with four cavities in it. It is made of silicon rubber, and both ends are sealed. There is a high-performance Kevlar fiber densely and evenly wrapped around the cylinder to prevent the rubber expanding in the radical direction. When one of the cavities is supplied with compressed air, the joint bends. Using the silicone rubber, the bending joint possesses many degrees of freedom theoretically, which means that the deformation of the joint is difficult to predict. Before fabricating the physical model of the bending joint, finite-element analysis was employed to simulate the deformation of the joint and to aid the structure design. As the above material test shows, the silicone rubber exhibits a nearly linear strain–stress relationship up to 250 % strain, so the soft material is considered as linear and isotropic in finite-element analysis. The result is shown in Fig. 5a. When one of the chambers is inflated with compressed air the joint will bend. Figure 6 shows the relation between the pressure and the bending angle.

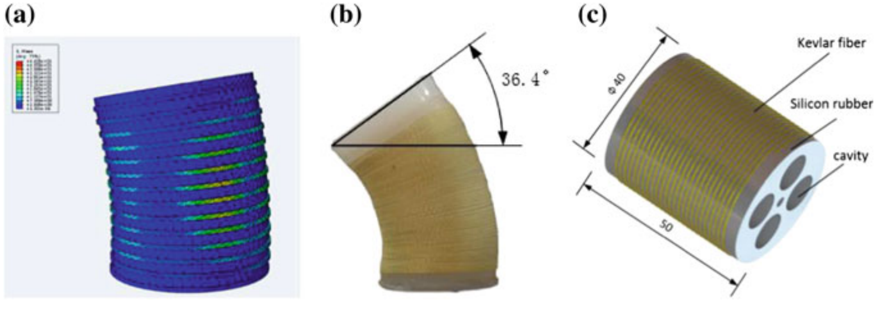


Fig. 5 **a** Deformed model in FEA environment, **b** a prototype, and **c** schematic design of the bending joint

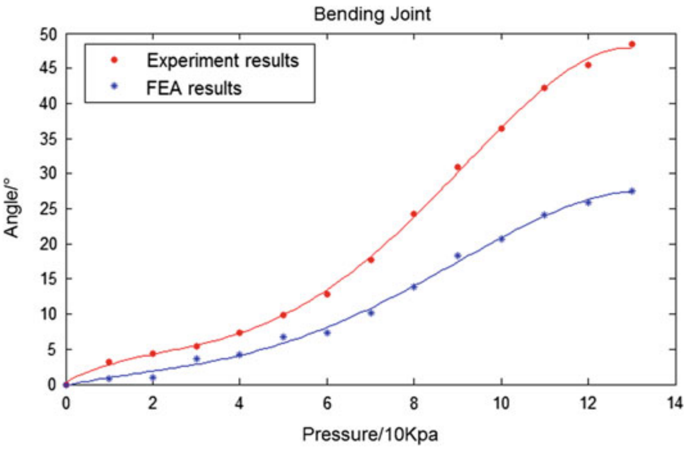
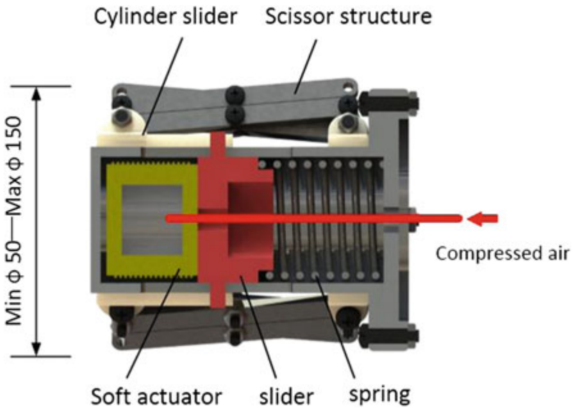


Fig. 6 Relation between the pressure and the bending angle

2.4 Design of the Annular Gripper

Unlike the cylindrical water pipes,the dimension of the passage that leads to the survivors varies, the gripper should be capable of adapting to unstructured environment. To achieve the peristaltic motion, at least two grippers are needed. As is shown in Fig. 7, the annular gripper is composed of six scissor structures uniformly distributed around a cylinder, a cylindrical slider, a soft actuator, and a spring. When the soft actuator is supplied with compressed air, it elongates and pushes the cylindrical slider. Under the drive of the slider, the gripper expands in the radical direction. Once the soft actuator deflates, the gripper contracts under the drive of the spring. The original diameter of the gripper is 50 mm, and it can reach 150 mm at most.

Fig. 7 Schematic design of the annular gripper



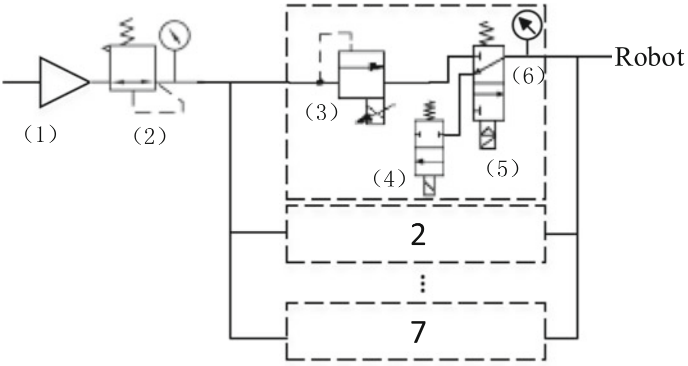


Fig. 8 Pneumatic drive system: *1* air source; *2* air pressure regulator; *3* proportional solenoid valve; *4*, *5* directional valve; and *6* pressure sensor

3 Preliminary Experiments

Preliminary experiments were conducted to investigate the performance of the robot. An environment that consists of different sizes of stones is built as the experiment place, aiming to mimic the disastrous situation. Figure 8 shows the pneumatic drive system used during the experiments. Figure 9a shows the prototype, and Fig. 9b shows the artificial disastrous environment. Under such circumstance, the robot is capable of navigating through the passage at a speed of 100 mm/min approximately.

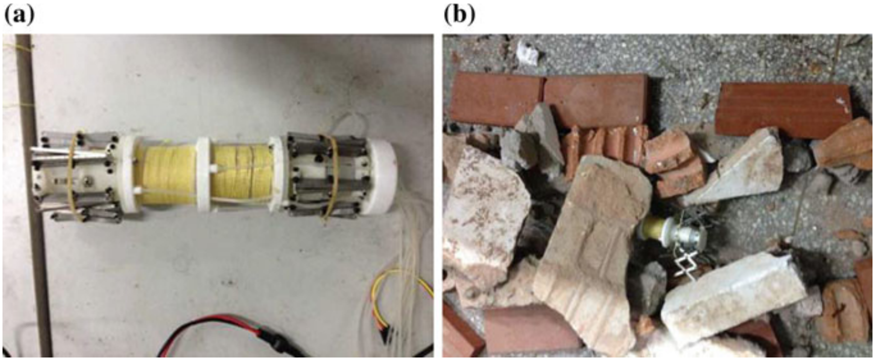


Fig. 9 **a** A prototype of the robot and **b** artificial disastrous environment

4 Conclusion

In this paper, a novel rigid and soft combined robot that can be used in confined spaces was developed, and the structure design, mechanical behavior of the material, soft structure analysis, and preliminary experiments were described.

Tensile test shows the material exhibits a nearly linear strain–stress relationship up to 250 % strain, so the soft material is considered as linear and isotropic in finite-element analysis. Experiments were carried out to investigate the characteristic of the two soft components. Both of the components show nonlinear features. At last, experiments were conducted to test the ability of the robot. Results show that the robot is capable of navigating through the passage at a speed of about 100 mm/min under the artificial disastrous environment.

References

1. Majidi, C.: Soft robotics: a perspective—current trends and prospects for the future. Soft Robot. **1**(1), 5–11 (2014)
2. Shepherd, R.F. et al.: Multigait soft robot. In: Proceedings of the National Academy of Sciences, vol. 108, no. 51, pp. 20400–20403 (2011)
3. Maeder-York, P. et al.: Biologically inspired soft robot for thumb rehabilitation. J. Med. Devices **8**(2), 020933 (2014)
4. Galloway, K.C. et al.: Mechanically programmable bend radius for fiber-reinforced soft actuators. In: 2013 16th International Conference on Advanced Robotics (ICAR). IEEE (2013)
5. Lipson, H.: Challenges and opportunities for design, simulation, and fabrication of soft robots. Soft Robot. **1**(1), 21–27 (2014)
6. Trivedi, D. et al.: Soft robotics: biological inspiration, state of the art, and future research. Appl. Bionics Biomech. **5**(3), 99–117 (2008)
7. Chen, F.J., et al.: Soft actuator mimicking human esophageal peristalsis for a swallowing robot. IEEE/ASME Trans. Mechatr. **19**(4), 1300–1308 (2014)

Design and Experimental Research of Pneumatic Soft Humanoid Robot Hand

Mengqian Tian, Yu Xiao, Xingsong Wang, Jianfei Chen and Wei Zhao

Abstract A pneumatic soft robot hand which imitates the shapes and gestures of human hand is introduced in this article. Soft robotics, 3D printing technology, and casting technology are used to fabricate the soft robot hand. The robot hand consists of palm and fingers. The thumb has two joints, and other four fingers have three joints. The fingers are driven by soft actuators which are made of highly extensible, elastomeric material (Ecoflex 00-30; Smooth-On, Inc.). Kinematics model is established based on the homogeneous coordinate transformation matrix. The soft robot hand has good flexibility and compliance; thus, it can perform complex gestures and grasp objects with different shapes. The robot hand is mainly made of silicon rubber which makes it light in weight and compatible with different objects; additionally, it is even capable of lifting heavy objects. The heaviest object it can grasp weighs 0.55 kg. The pneumatic soft humanoid robot hand can be applied in industrial manufacturing and medical field.

Keywords Pneumatic • Soft actuator • Humanoid robot hand • Soft robotic

M. Tian · Y. Xiao · X. Wang (✉) · J. Chen · W. Zhao
School of Mechanical Engineering, Southeast University,
Nanjing 211189, Jiangsu, China
e-mail: xswang@seu.edu.cn

M. Tian
e-mail: 101004755@seu.edu.cn

Y. Xiao
e-mail: xxmuyu1989@163.com

J. Chen
e-mail: cjfseu@163.com.cn

W. Zhao
e-mail: zhao_baowei@126.com

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_37

1 Introduction

With the rapid development in the past decades, robots can do almost everything. However, robot hands are not so flexible and reliable as human hands. Therefore, robot hand which imitates the shape, structure, and gestures of human hand has attracted continuous attention from researchers. In 2004, British Shadow Company successfully developed a dexterous hand Shadow [1–3] which was driven by pneumatic artificial muscles (PMA) and ropes. Afterward, researchers turn to new functional materials such as shape memory alloys [4], polymers [5–7], and functional liquid [8].

Recently, researchers began to focus on soft robots designed with biological characteristics. The team from Harvard University led by American chemist George M. Whitesides has developed a bionic starfish robot [9] driven by compressed air. It can squirm like a worm with its soft body in a very narrow space. In 2013, another team from Harvard University developed a soft pneumatic glove [10] for hand rehabilitation based on soft robotic. The same year, a soft prosthetic hand [11] for amputees was designed by a group of students from the Global Immersion Summer Program in India. Its thumb was designed as a spherical hinge made of aluminum while its other four fingers consist of fiber reinforced actuators. Zhejiang University of Technology of China developed a pneumatic flexible five-fingered robot hand [12] with unidirectional and multidirectional bending joints. In this article, a pneumatic soft humanoid robot hand is developed to perform complex gestures and grasp objects with different shapes. The kinematics model for the soft hand is developed based on homogeneous coordinate transformation matrix.

2 Structural Design of the Pneumatic Soft Robot Hand

Human hand is mainly composed of metacarpals, phalanges, carpals, and the surrounding soft tissue. All the five fingers connect to metacarpals through metatarsophalangeal joint. The thumb consists of base joint, near joint while the other four fingers consist of base joint, near joint, far joint. The carpometacarpal joint of thumb can perform flexion, extension, adduction, exhibition, and rotation. Other four fingers can only achieve flexion and extension with uniaxial joint.

In this paper, a pneumatic soft robot hand was developed imitating the human hand. As depicted in Fig. 1a, the soft robot hand consists of five fingers and a palm. The connecting rod plunges into the supporting block, and they are surrounded by elastomeric material (Ecoflex 00-30; Smooth-On, Inc.) to serve as the palm. The connecting rod and the supporting block are created by 3D printer. The fingers are driven by the actuators fabricated through casting technology. The soft robot hand weights 0.38 kg.

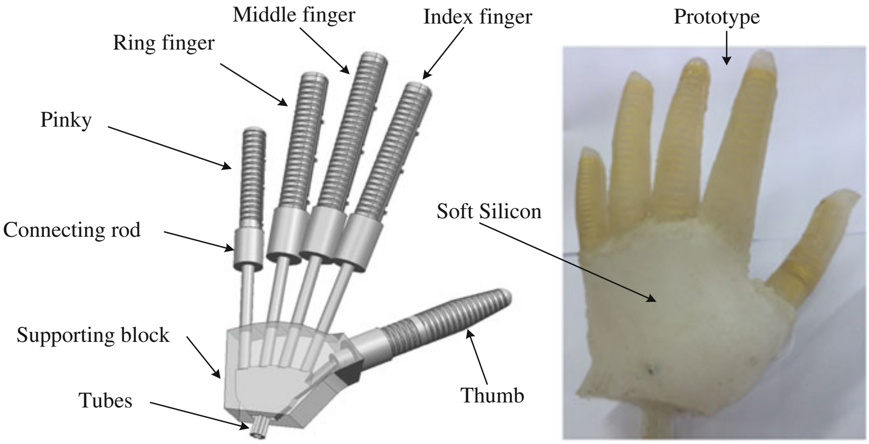


Fig. 1 General structure and prototype of the pneumatic soft humanoid robot hand

2.1 Structural Design of Four Fingers

Each of the four fingers has three uniaxial rotary joints with same structure. Therefore, only index finger is introduced in this article in detail. As is shown in Fig. 2, the index finger consists of a bending actuator made of elastomeric material (Ecoflex 00-30) and four rigid pieces of plastic connected by three links. These two components are glued together. Rigid parts prevent the bending actuator from expanding in the region of rigid boards. The inextensible Kevlar fibers wrapped around the actuator constrain its expenditure in radical direction and provide a more reliable connection to ensure no relative movement on the interface between them.

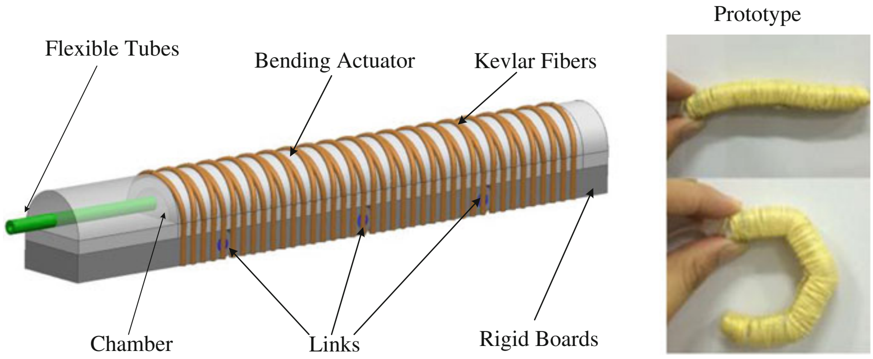


Fig. 2 General structure and prototype of the finger

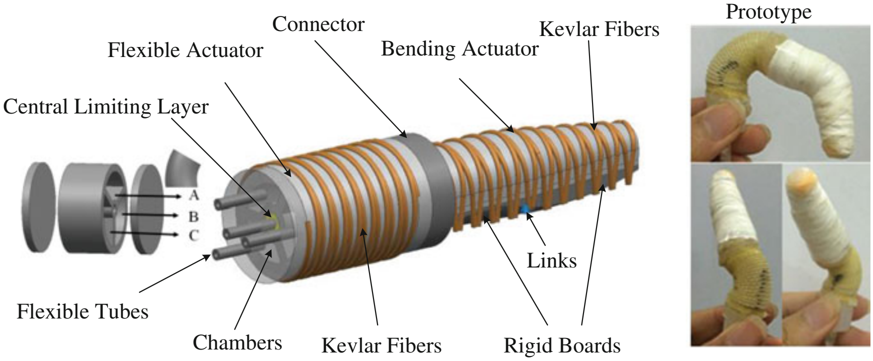


Fig. 3 Structure and motion function of the thumb

Since one part expands axially and another part does not, the bending actuator will drive the rigid boards to rotate around the links when the chamber is inflated with compressed air.

As is depicted in Fig. 3, the thumb consists of a flexible actuator (Ecoflex 00-30) and a bending actuator which are linked by a connector. The bending actuator has the same structure as the index finger and it is wrapped with Teflon tapes to fasten the Kevlar fibers. There are three chambers and a center hole in the flexible actuator which is wrapped with Kevlar fibers to constrain its expenditure in radical direction. The central hole is filled with relatively inextensible material (Dragon skin 30, yellow part in Fig. 3) to serve as the central limiting layer, preventing the actuator from extending in axial direction. The actuator will bend when one of the chambers is inflated with compressed air. Changing the air pressure in the three hollow chambers numbered with A, B, C independently, the flexible actuator will bend in multiple different orientations. Four tubes are used in the thumb: one tube goes through the center part of the flexible actuator to plunge into the bending actuator, and the other three tubes plunge into the flexible actuator.

2.2 Structural Design of Thumb

Human hand can achieve accurate and powerful grasp of different objects and perform complex gestures with high dexterity due to the high dexterity and adaptability of its thumb. So the design of thumb is crucial.

3 Kinematics of the Pneumatic Soft Humanoid Hand

Building kinematics model for the soft robot hand is essential to the control of the motion of fingers. Homogeneous coordinate transformation matrix of joint is used to calculate position and orientation of the key points in the palm coordinate system according to the joint angle.

To facilitate the analysis of the position and posture of the robot hand, we define axis intersection coincides with the origin of the palm coordinate system $o_0x_0y_0z_0$ (see Fig. 4). There are certain angles among fingers, from index finger to pinky, the angles between each finger and the axis-y_0 are φ_1, φ_2, φ_3, φ_4, and φ_5 and the values are −45°, −14°, 0°, 14°, and 24°, in turn. As shown in Figs. 5 and 6, using coordinate system $o_{ij}x_{ij}y_{ij}z_{ij}$ to denote the local coordinate system of joints and fingers, where i (i = 1, 2, 3, 4, 5) is the symbol of finger, and it denotes the thumb, index finger, middle finger, ring finger, and pinky, respectively; j (j = 0, 1, 2, 3, 4) denotes the labels of the local coordinate system. L_{12}, L_{13}, and L_{14} denote the length of the proximal phalange, middle phalange, and distal phalange,

Fig. 4 Coordinate system of the palm of the robot hand

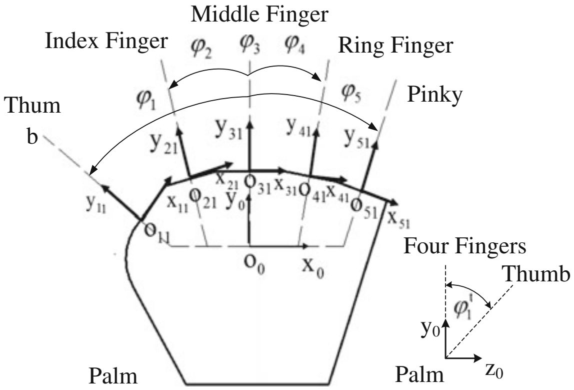
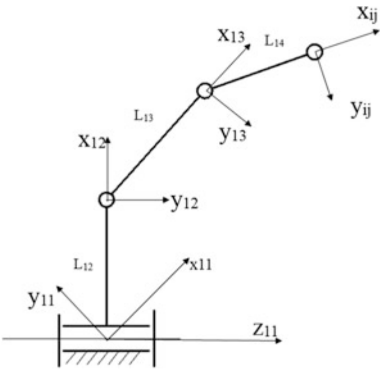


Fig. 5 Coordinate system of the thumb of the robot hand



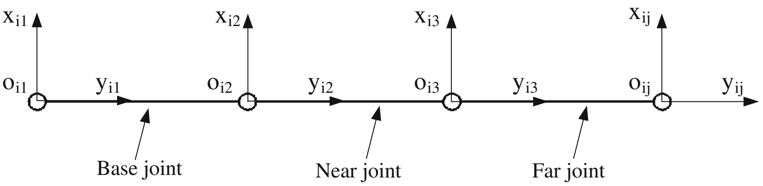


Fig. 6 Coordinate system of the four fingers of the robot hand

respectively. The thumb bends in multiple different orientations, the four fingers bend to revolve around axis-x_{ij}.

In order to form closing force, the thumb is not placed in the same plane as the palm action of the thumb is affected synthetically by the flexible actuator and the unidirectional bending joints. Movements of the flexible actuator can be described as a sphere joint without the axial rotation. So the sphere joint with two rotational degrees of freedom can be replaced by a cardan joint and the value of L_{12} is zero. Figure 5 denotes the kinematic sketch of the thumb. Base coordinate system of the thumb is $o_{11}x_{11}y_{11}z_{11}$. As the result of the origin of the coordinate system coinciding with the key points, ${}^{j}P_{1j}$ denotes the coordinates of the key points in the local coordinate system, that is = [0, 0, 0, 1]. The coordinates of any key points within the thumb in base coordinate system can be expressed as follows:

$$ {}^{11}_{1j}P = {}^{1}_{2}T_1 \ldots {}^{j-2}_{j-1}T_1 \cdot {}^{j-1}_{j}T_1 \cdot {}^{j}P_{1j} \tag{1} $$

where ${}^{j-1}_{j}T_1$ denotes the transformation matrix from coordinate system j to coordinate system j − 1 within the thumb, that is,

$$ {}^{j-1}_{j}T_1 = \begin{cases} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta_{1j} & s\theta_{1j} & a_{1j} \\ 0 & s\theta_{1j} & c\theta_{1j} & b_{1j} \\ 0 & 0 & 0 & 1 \end{bmatrix} & \mathrm{j} = 3, 4 \\ \begin{bmatrix} c\theta_{1j} & s\theta_{1j} & 0 & a_{1j} \\ s\theta_{1j} & c\theta_{1j} & 0 & b_{1j} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \mathrm{j} = 2 \end{cases} \tag{2} $$

where θ_{1j} is the bending angle of the joints. Coordinates of the key points in base coordinate system can be expressed as (0, a_{ij}, b_{ij}). *h* denotes the tip thickness of the thumb. *s* and *c* are short for sin and cos.

The thumb is not placed in the same plane as the palm in order to form closing force. The projection from the thumb onto the palm can have an angle with the middle finger. Angle between the thumb and palm is β. The transformation matrix from coordinate system $o_{11}x_{11}y_{11}z_{11}$ to $o_0x_0y_0z_0$ can be expressed as follows:

$$
{}^{0}_{1}T_{1} = \begin{bmatrix} R_{0,\beta,\alpha} & P \\ 0 & 1 \end{bmatrix} \tag{3}
$$

where $[R_{0,\beta,\alpha}]$ is the rotational transformation matrix from $o_0x_0y_0z_0$ to $o_{10}x_{10}y_{10}z_{10}$, that is,

$$
\left[R_{0,\beta,\alpha}\right] = \left[R_{\alpha,z}\right] \cdot \left[R_{\beta,x}\right] \tag{4}
$$

Thus, the kinematics model of the thumb has been completed, and the kinematics model of the four fingers can be completed with similar method. Workspace of pneumatic soft humanoid robot hand can be achieved through the kinematics model of five fingers.

4 Experiments

Experiments have been done to test the ability of the soft robot hand. As is depicted in Fig. 7, the platform includes a pneumatic soft robot hand, a pneumatic solenoid valve, a throttle valve, a precise reducing valve, a DC power supply, and an air

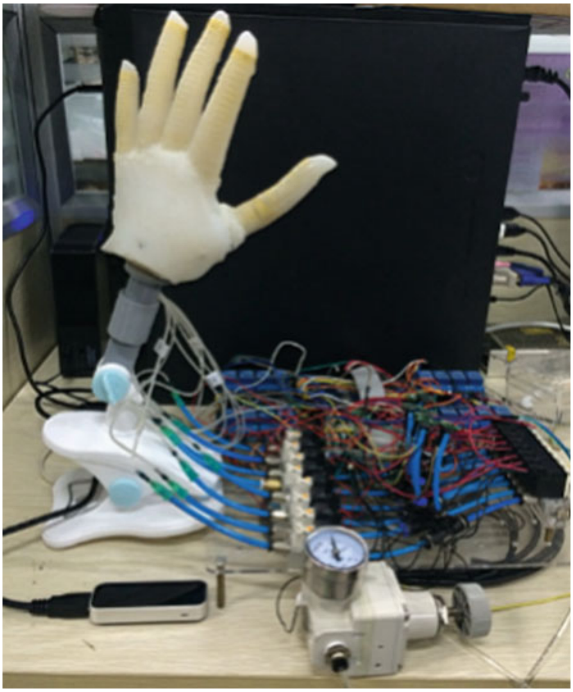


Fig. 7 Experimental platform

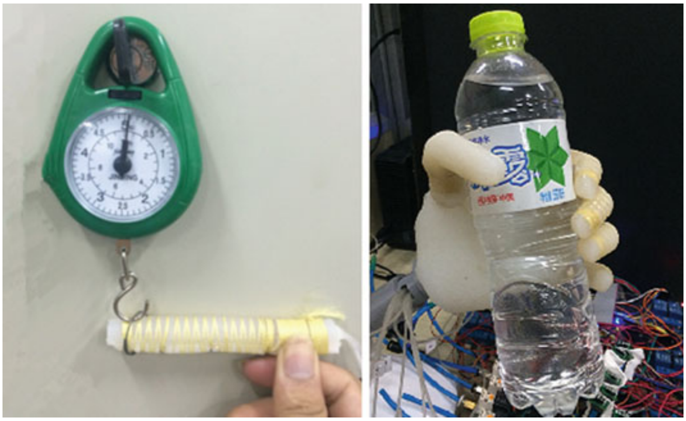


Fig. 8 Grasping forces of the fingers and the robot hand

pump. The throttle valve is used to adjust the air flow while the precise reducing valve is used to adjust the pressure.

4.1 Grasp Force

As is depicted Fig. 8, a spring dynamometer is used to test the clamping force of the fingers. One end of the finger is fixed on the wall, the other end will bend when inflated with compressed air. Results show that the maximal bending force of the fingers is 7 N. The hand can pick up a bottle of water that weighs 5.5 N. The heaviest object it can grasp can reach 6 N.

4.2 Grasp Postures

As is depicted in the Fig. 9, the soft robot hand can achieve typical gestures imitating the human hand such as “OK” (Pic A2), “Orchid” (Pic A3), “Thumb Up” (Pic A4), “V” (Pic A5), and “1” (Pic A6).

4.3 Experiments of Grasping Objects with Different Shapes

The size of the soft hand is identical with a human hand, and it can perform gestures such as grasp, pinch, catch, and drag by controlling the pressure of chambers

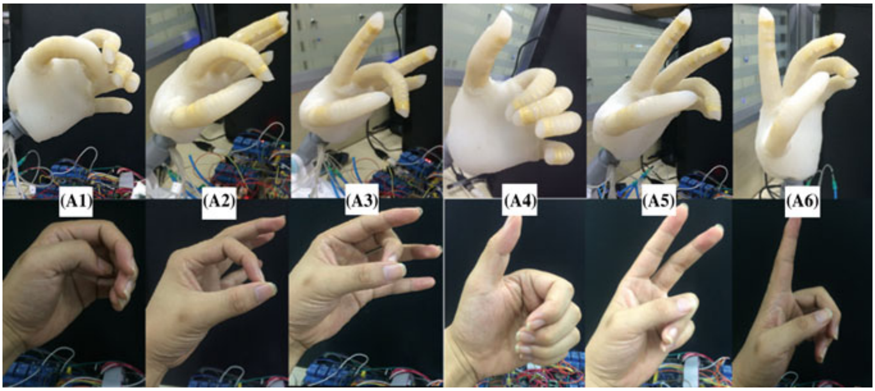


Fig. 9 Postures of the soft robot hand

formed in the joints. The soft robot hand can grasp different objects with different gestures.

As is depicted in Fig. 10, experiments have been performed to test the capability of the soft hand, the objects used in the experiments include eggs (Pic B1, B6), pencil (Picture B2), boxes (Pic B3, B4, B8, B9), cups (Pic B5, B7), and bottle (Pic B4). For some large-scale objects like cylinders, the soft hand grasps (Pic B3, B4, B7). For relatively small balls and cylinders, it uses its thumb and other fingers to catch them (Pic B1, B2, B6). For heavy objects such as boxes (Pic B8), it lifts them. The results (B1, B6) show the soft hand has advantage in grasping fragile objects like eggs because of its elasticity.

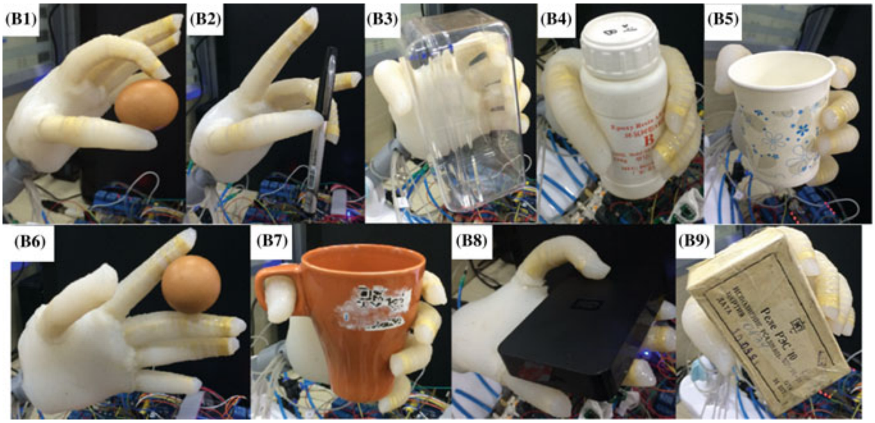


Fig. 10 Pneumatic soft robot hand grasping objects of different shapes

5 Conclusion

In this paper, an 8-DOF soft hand was proposed based on bionics. Soft robotics, 3D printing technology, and casting technology have been adopted to make the shape and size of the robot hand as similar as possible to the human hand. It is capable of realizing a lot of kinds of gestures such as grasp, pinch, catch, and drag. Homogeneous coordinate transformation matrix is employed to establish a kinematics model to achieve the workspace for the soft robot hand. Experiments have been done based on the pneumatic control platform, and the result indicates that the robot hand can perform with high dexterity. It can grasp balls, cylinders, blocks, and irregular objects. The heaviest object it can grasp weighs 0.55 kg.

References

1. Chou, C.P., Hannaford, B.: Measurement and modeling of mckibben pneumatic artificial Muscles. IEEE Trans. Robot. Autom. **12**(1), 90–102 (1996)
2. Tsagarakis, N., Caldwell, D.G.: Improved modeling and assessment of pneumatic muscle actuators. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 3641–3646. IEEE Press, San Francisco (2000). Shadow Dexterous Hand CS Technical Specification, vol. 5. Shadow Robot Company (2008)
3. Shadow Robot Company. Shadow Dexterous Hand CS Technical Specification, vol. 5 (2008)
4. Nakano, Y., Fujie, M., Hosad, A.Y.: Hitachi's robot hand. Robot. Age **6**(7), 18–20 (1984)
5. Akihiro, Y., Kenjiro, T., Shinichi, Y., Kazuya, E.: A robot hand using electro-conjugate fluid. Sens. Actuators A: Phys. **170**, 139–146 (2011)
6. Akihiro, Y., Kenjiro, T., Shinichi, Y., Kazuya, E.: A robot hand using electro-conjugate fluid: grasping experiment with balloon actuators inducing a palm motion of robot hand. Sens. Actuators A **174** 181–188 (2012)
7. Ran, P., Wang, B., Wang, W.: The design of communication convertor based on CAN bus. In: IEEE International Conference on. Industrial Technology, pp. 1–5. IEEE (2008)
8. Schulz, S., Pylatiuk, C., Bretthauer, G.: A new ultralight anthropomorphic hand. In: Proceedings of the 2001 IEEE International Conference on Robotics and Automation, Seou, Corea, (2001)
9. Shepherd, R.F., et al.: Multigait soft robot. Proc. Natl. Acad. Sci. **108**(51), 20400–20403 (2011)
10. Polygerinos, P., Lyne, S., Zheng, W., Nicolini, L.F.: Towards a soft pneumatic glove for hand rehabilitation. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 3–7 Nov 2013
11. Global, I.S.P.: Soft robotic prosthetic hand for amputees [EB/OL]. http://softroboticstoo-lkit.com/book/case-study-soft-robotic-prosthetic-hand-amputees-fr
12. Wang, Z.H., Bao, G.J., Zhang, L.B., et al.: Design and control of integrated pneumatic dexterous robot finger. J. Cent. S. Univ. Technol. **18**(4), 1105–1114 (2011)

Concept of Distributed Processing System of Image Flow

Aleksey Kondratyev and Igor Tishchenko

Abstract The paper describes a concept of software tools for data stream processing. The tools can be used to implement parallel processing systems. Description of the task is presented in the first part of paper. The system is based on pipeline parallelism and was distributed for using on a cluster computer. The paper describes a base scheme and a main work algorithm of the system. An actual application example is presented. The system has some weak sides which are described at the end of paper. Direction of future research is presented at the end of the article.

Keywords Distributed computing • Parallel computing • Cluster • Multi-processor systems • Software tools

1 Introduction

Active development of parallel computing is accompanied by the growth of computing power and the creation of high-performance systems. Creation of supercomputers has several objectives:

- The maximum capacity (Top 500).
- The solution to a specific problem (data storage and data processing).

Supercomputers are not used for playing games or watching movies, so the majority of computer users are not interested in them. No one would create a cluster computer without a goal because of the high production and maintenance costs. Clusters without load usually are switched off completely or partially. We need

A. Kondratyev · I. Tishchenko (✉)
Program System Institute of Russian Academy of Sciences, Pereslavl-Zalessky,
Russian Federation
e-mail: igor.p.tishchenko@gmail.com

A. Kondratyev
e-mail: ronkajitsu@gmail.com

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_38

high-quality custom software that is designed to perform a specific task which will make a full use of a cluster computer. In most cases, users create this software “from scratch” by using libraries for parallel programming (MPI, Ocaam, OpenMP, PVM, etc.). Highly skilled professionals who can produce the correct decomposition of tasks for parallel execution and its correct implementation are required in order to develop effective programs to fully utilize available hardware.

One of many problems which require capabilities of supercomputers is the problem of data flow processing. The data may be from sensor nodes with wide spectrum. The data processing could consist of several stages such as searching, tracking, and monitoring object state. The main condition is a possibility to schematize task as a block scheme of processing stages.

2 General System Features

To split the computations into independent parts, it is essential to analyze the task solution scheme. Equal computation volume and minimization of dependencies are the main requirements for splitting. In most cases, analysis and task splitting are complex problems. To use the low-level libraries in a new supercomputer application, it is important to have highly qualified specialists and a lot of time. The best way is to use special software tools in most cases [1].

Our program system has been successfully used in various scientific fields and allows quickly creating applications oriented toward running on cluster nodes. Some of them could be named as an example:

- processing of remote sensing data [2];
- processing of medical information [3];
- designing and using of neural networks [4, 5];

The system allows dividing roles between users and developers. Users can solve a task without writing program code in contradistinction to typical ways. User may not know how to split algorithm into parts and make parallel program. They could just build a linear scheme of data processing which consists of program modules with data channels and initial conditions (see Fig. 1).

Users can describe the whole data process algorithm by modules. The scheme will have modules with typical connection slots. It may be done by the graphical user interface (GUI) (see Fig. 2).

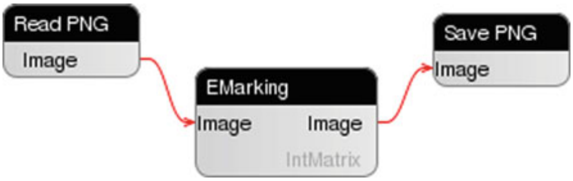


Fig. 1 Scheme of image region isolation

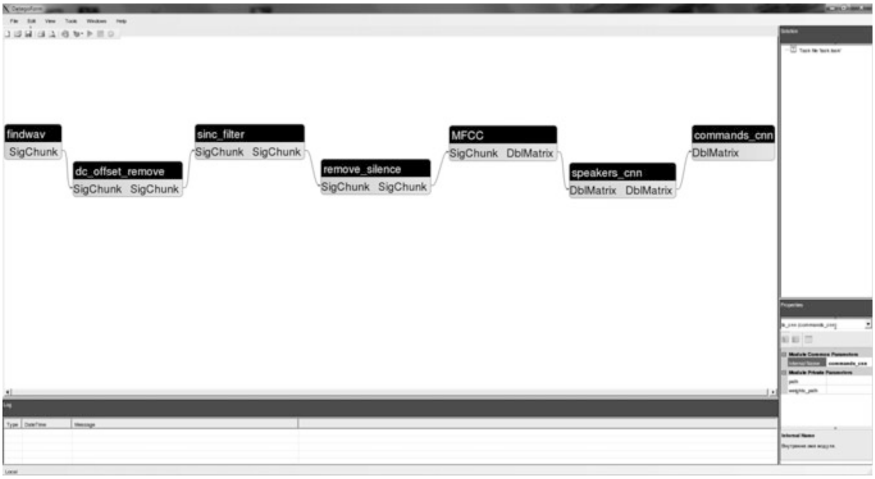


Fig. 2 Graphical user interface

Table 1 The parallel program effectiveness

CPU kernel count	Time	Acceleration
1	185.7	1
2	93.9	1.98
3	64.8	2.86
4	51	3.64
5	47.7	3.89
6	33	5.48

GUI has all the information about modules like a list of available modules and channels. It allows users to make some part of work using their intuition and detect some errors on the design stage.

The system algorithms could help reach calculation acceleration on supercomputers. As an example, let us consider a problem of repainting a map based on texture classification. Results of the tests are shown in Table 1 and are illustrated in Figs. 3 and 4.

The textural classifier provides results inaccessible for point classifiers. But it requires more hardware resources. The system parallelism could provide close to linear acceleration as shown in Fig. 3. As shown in Fig. 4, processing time can be easily decreased in some cases.

The field of parallel processing is very well studied. A lot of software tools were created for a lot of applications. Some of them are completed software. Some of the software tools can be used for implementation of parallel data processing systems: CODE, HeNCE, GRADE, TRAPPER, Kepler, etc. But all of them require knowledge about parallel programming. Most of them are only GUI for parallel programming library. And only Kepler allows implementing applications with

Fig. 3 Speedup of the program

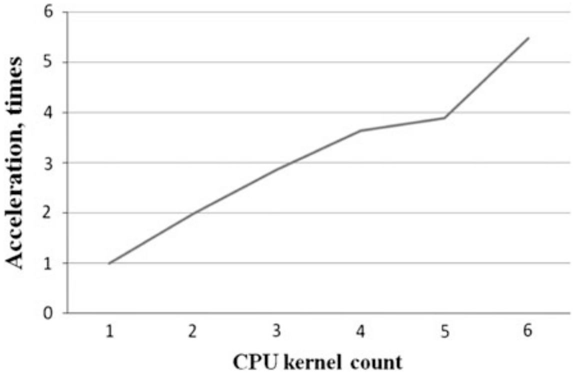
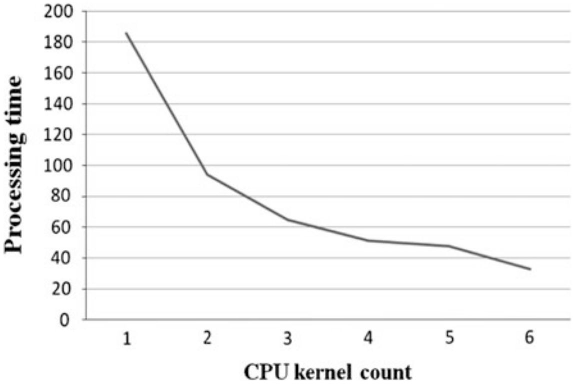


Fig. 4 The parallel program results



conveyor. Other software does not allow this. The described tools simplify the development of new data processing systems rather than trying to achieve maximum performance.

3 Key Features

The system is based on a modular structure. Each component has its own role in the system, but all of them are intended for one goal. The system is the environment for running program modules in the pipeline parallelism mode. Principal working scheme is presented in Fig. 5. Figure 6 illustrates system architecture.

Each module is a minimal program with some processing function and may have several input and output channels. The system considers each module as one function with input parameters. It provides an ability to organize calculation process with pipeline parallelism.

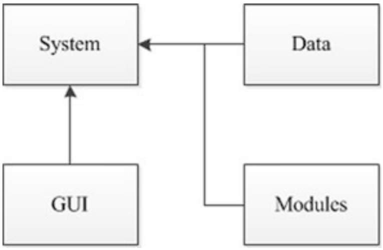


Fig. 5 Principal working scheme

In the traditional work-sharing approach, threads run different iterations concurrently [6–9]. Each thread runs iteration in its entirety and then proceeds to the next iteration. An alternate approach is to use pipeline parallelism where we split loop iterations into stages and threads operate on different stages from different iterations concurrently (see Fig. 7).

Pipeline parallelism is powerful because it can expose parallelism in ordered loops where iterations are non-independent and cannot run concurrently. By splitting each loop iteration into segments, we can expose intraiteration parallelism. When multiple cores are assigned to a pipeline stage, its throughput increases (ideally) linearly. Thus, a common mechanism to speed up a pipeline workload is to assign more cores to the slowest stages to balance the throughput. However, programmers are often unable to balance pipelines completely which leads to thread waiting and thus lost performance opportunity [10].

This work is performed by the system. The system transforms a task into a list of system commands. The system can be described by a state machine. Each command is a conditional transition between its states. Each command describes some action like data transfer or module execution. System scheduler strives to use all available resources at any time [11]. With the release of hardware resources, scheduler must choose a command which can be executed in that moment from the command queue. This method proved to be good for modules with considerable time of processing for little amount of data. So the time for command selection is smaller than processing time.

When we talk about image stream processing, we have a lot of fast processing steps. If we try to test it as a part of the system, we get a terrible result with multiple nodes. The time of processing and transferring data is smaller than command selection time. Similar type of problems must be solved by another method of scheduling and data management. The main idea of that method lies in phrase "transfer data when needed." A system module is described as a "pure" function. It means that a module has two input parameters. One of them is data for processing, and another one is an internal module state with module initial parameters. The system tries to process the data locally for cluster node.

Schematically, the basic algorithm can be described as follows:

1. System initialization;
2. Waiting for new task;
3. Generating commands from input task;

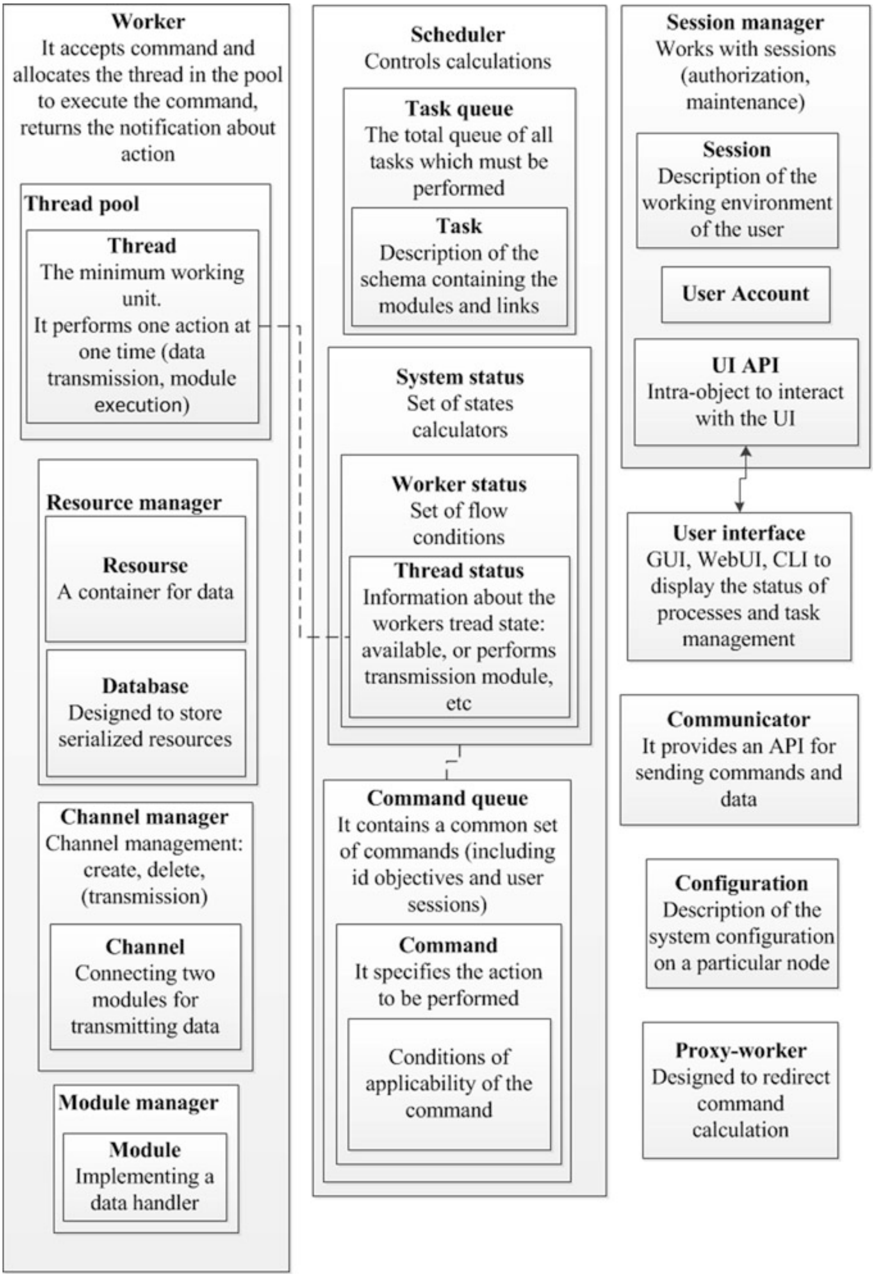

Worker
It accepts command and allocates the thread in the pool to execute the command, returns the notification about action
Thread pool
Thread
The minimum working unit.
It performs one action at one time (data transmission, module execution)
Resource manager
Resourse
A container for data
Database
Designed to store serialized resources
Channel manager
Channel management: create, delete, (transmission)
Channel
Connecting two modules for transmitting data
Module manager
Module
Implementing a data handler
Scheduler
Controls calculations
Task queue
The total queue of all tasks which must be performed
Task
Description of the schema containing the modules and links
System status
Set of states calculators
Worker status
Set of flow conditions
Thread status
Information about the workers tread state: available, or performs transmission module, etc
Command queue
It contains a common set of commands (including id objectives and user sessions)
Command
It specifies the action to be performed
Conditions of applicability of the command
Session manager
Works with sessions (authorization, maintenance)
Session
Description of the working environment of the user
User Account
UI API
Intra-object to interact with the UI
User interface
GUI, WebUI, CLI to display the status of processes and task management
Communicator
It provides an API for sending commands and data
Configuration
Description of the system configuration on a particular node
Proxy-worker
Designed to redirect command calculation

Fig. 6 System architecture

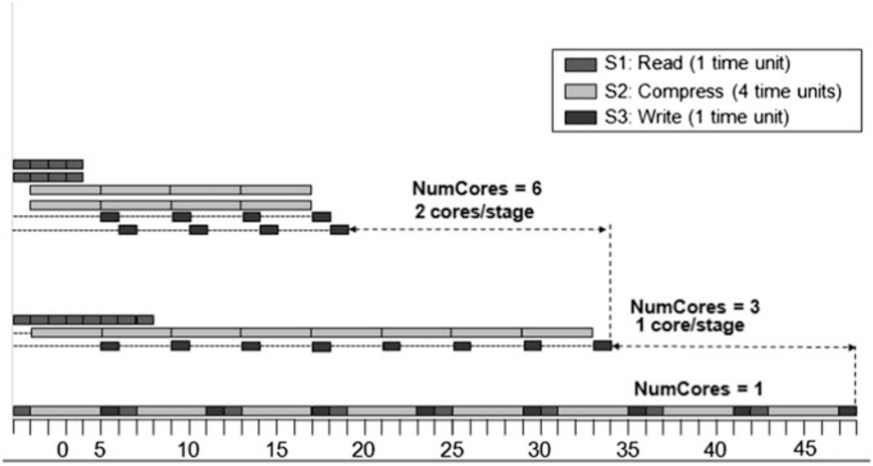


Fig. 7 Pipeline parallelism

4. Module initialization on cluster nodes;
5. Command executing with rules:
   - 5.1. Process data on one node with several modules;
   - 5.2. Spread data from stream on several nodes;
   - 5.3. Combine data transmissions;
   - 5.4. Minimize data transmissions;
6. After the completion of the task, go to Step 2.

The importance of the data distribution for processing among cluster nodes is proved by various papers. Preliminary data distribution can significantly reduce the processing time [12–15].

4 Conclusions

The described system is useful software tools for users who do not have knowledge about parallel programming. The processing division into stages allows to process data independently with several threads. It may be perfect choice for quickly making data processing application for distributed system. Prepared scheduling algorithms allow obtaining acceleration in various problems.

The tools have several drawbacks. The best results can be achieved with coarsely granular concurrency. Scheduler has numerous leaks of time when choosing the next action. The other disadvantages are as follows:

1. Not all algorithms are suitable;
2. Maximum load of computing resources is not always the best choice.

The concept has worked well for the computer network. The next stage is to use the concept in a heterogeneous computing environment. Various computing power of nodes requires a more sophisticated approach to load balancing.

Acknowledgments This work was supported by the Ministry of Education and Science of the Russian Federation: № 14.607.21.0088 agreement for a grant on "Development of methods and means of processing and intelligent analysis of images and dataflow obtained from a variety of stationary and mobile sensors, using high-performance distributed computing for the tasks of monitoring the premises and the surrounding area." Unique identifier is RFMEFI60714X0088.

References

1. Rutledge, E., Kepner, J.: PVL: an object oriented software library for parallel signal processing. In: Proceedings of the 2001 IEEE International Conference on Cluster Computing (CLUSTERí01). http://www.computer.org/csdl/proceedings/cluster/2001/1116/00/11160074.pdf
2. Stepanov, D.N., Kiryushina, A.E., Ivanov, E.S., Kondratiev, A.A.: Software for pipeline parallel processing of remote sensing data in the cluster computing installations and graphics processing unit. In: Proceedings of Junior research and development conference of Ailamazyan Pereslavl university, Pereslavl, SIT-2014, pp. 5–20 (2014). https://edu.botik.ru/proceedings/sit2014.pdf
3. Zadneprovsky, V.F., Talalaev, A,A,, Tishchenko, I.P., Fralenko, V.P., Khachumov, V.M.: Software tool complex high-performance image processing for medical and industrial use. Inf. Technol. Comput. Syst. **1**, 61–72 (2014). https://docs.google.com/uc?export=download&id=0B-Qay3kEFxqfSk8weXVwYmgtZ0E
4. Talalaev, A.A., Tishenko, I.P., Fralenko, V.P., Khachumov, V.M.: Analysis of the efficiency of applying artificial neuron networks for solving recognition, compression, and prediction problems. Sci. Tech. Inf. Process. **38**, 313–321 (2011). https://docs.google.com/uc?export=download&id=0B-Qay3kEFxqfWlFKMzhONmVWVTQ
5. Kondratyev, A.A.: Parallel clustering of color images based on the self-organizing maps Kohonen cluster using calculators. In: Proceedings of Junior research and development conference of Ailamazyan Pereslavl university. Pereslavl, SIT-2012, pp. 57–70 (2012). http://conf.sci.pfu.edu.ru/index.php/ittmm/2012/paper/view/313/425
6. Czajkowski, K., Foster, I.: A resource management architecture for metacomputing system. In: Job Scheduling Strategies for Parallel Processing (JSSPP 1998): Proceedings of the 4th Workshop, Orlando, Florida, USA, 30 March 1998
7. Seredynski, F., Zomaya, A.Y.: Sequential and parallel cellular automata-based scheduling algorithms. In: IEEE Trans. Parallel Distrib. Syst. **13**(10) (2002)
8. Li, K., Tang, X., Li, K.: Energy-efficient stochastic task scheduling on heterogeneous computing systems. IEEE Trans. Parallel Distrib. Syst. **1**. doi:10.1109/TPDS.2013.270
9. Pricopi, M., Mitra, T.: Task scheduling on adaptive multi-core. In: IEEE Trans. Comput. **1**. doi:10.1109/TC.2013.115
10. Suleman, M.A.: Parallel programming: do you know pipeline parallelism? http://www.futurechips.org/parallel-programming-2/parallel-programming-clarifying-pipeline-parallelism.html

11. Marshall, P., Keahey, K., Freeman, T.: Improving utilization of infrastructure clouds. In: Cluster, Cloud and Grid Computing (CCGrid 2011): Proceedings of the IEEE/ACM International Symposium, Newport Beach, CA, USA, 23–26 May 2011
12. Tyutlyaeva, E., Kurin, E., Moskovsky, A., Konuhov, S.: Abstract: using active storage concept for seismic data processing. In: High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion, pp. 1389–1390 (2012)
13. Iverson, M., Ozguner, F.: Dynamic, competitive scheduling of multiple DAGs in a distributed heterogeneous environment. In: Proceedings of Seventh Heterogeneous Computing Workshop, Orlando, Florida, USA, pp. 70–78. IEEE Computer Society, 30 March 1998
14. Maheswaran, M., Ali, S.: Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems. J. Parallel Distrib. Comput. **59**(2), 107–131 (1999)
15. Baker, M., Buyya, R., Laforenza, D.: Grids and grid technologies for wide-area distributed computing. J. Softw. Pract. Experience **32**(15), 1437–1466 (2002)

A Likelihood-Based Data Fusion Model for the Integration of Multiple Sensor Data: A Case Study with Vision and Lidar Sensors

Jun Jo, Yukito Tsunoda, Bela Stantic and Alan Wee-Chung Liew

Abstract Sensors have been developed and applied in a wide range of fields such as robotics and autonomous vehicle navigation (AVN). Due to the inability of a single sensor to fully sense its surroundings, multiple sensors based on individual specialties are commonly used in order to complement the shortcomings and enrich perception. However, it is challenging to integrate the heterogeneous types of sensory information and produce useful results. This research aims to achieve a high degree of accuracy with a minimum false-positive and false-negative rate for the sake of reliability and safety. This paper introduces a likelihood-based data fusion model, which integrates information from various sensors, maps it into the integrated data space and generates the solution considering all the information from the sensors. Two distinct sensors: an optical camera and a LIght Detection And Range (Lidar) sensor were used for the experiment. The experimental results showed the usefulness of the proposed model in comparison with single sensor outcomes.

Keywords Likelihood · Data fusion · Sensors · Vision · Lidar

1 Introduction

Various sensors have been developed and applied in a wide range of fields. However, the environment is too complicated, diverse and continuously changing, making it difficult for a single sensor to collect and process all the information regarding its surroundings. Multiple sensors based on individual specialties may complement the shortcomings and enrich the perception function. For example, it is not easy to analyse road images captured by one camera alone, especially when taken during the night or as it passes the shadow of a tree. The Lidar, a laser range

J. Jo (✉) · Y. Tsunoda · B. Stantic · A.W.-C. Liew
School of Information and Communication Technology, Griffith University,
Nathan, Australia
e-mail: j.jo@griffith.edu.au

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_39

finder, works better in such situations regardless lighting conditions. However, data obtained from a Lidar often include many noises, especially when it scans visually complex objects, such as trees or grasses. To solve these problems, a combination of both the camera and the Lidar can constructively interact, hence enhancing the sensing performance. Nevertheless, the integration of heterogeneous types of data is challenging.

Data fusion is a technology that integrates information from distinct sources to form a unified picture in order to obtain more specific inference or combining multiple measurements from a single sensor [1]. Much research has produced a number of methods regarding the integration of the various types of sensor data [2–4]. Most of these methods, however, are problem specific and therefore are not suitable for the complex or diverse range of situations. Integrating such complicated data types needs a robust platform for various situations.

This research introduces a data fusion model that can be applied to various combinations of sensors. In the model, each sensor module will compute the likeliness of the sensory data and the system will integrate the values in order to produce an incorporated outcome. A case study was implemented for road boundary detection using a camera and a Lidar. The result from this implementation demonstrated its promising aspect.

2 A Data Fusion Model

The proposed model is composed of two levels of modules: individual sensor module (ISM) and data fusion module (DFM). The ISM contains various sensors and, through a number of processes such as segmentation and feature extraction, it produces likelihood values of the individual sensor data. The values from all the sensors are integrated into the DFM and describe the feature of an object. Two types of sensors were used for the implementation in this paper: a camera and a Lidar. Other types of sensors can be added to this model with simple variations. This section introduces the structure and operations of the model with an implementation conducted at the Griffith Robotics Laboratory.

2.1 Vision Sensors

Vision-based edge detection has been the most intensive area of research in AVN and robotics, due to their advantages over other sensors, such as in high performance to low-cost ratio. A vision-based system is usually performed through the three steps: pre-processing, segmentation and edge extraction, Fig. 1.

Image pre-processing helps reduce computational redundancy for the following processes and enhances the effectiveness of feature extraction. Two pre-processing tasks are used for this experiment: region of interest (ROI) and noise reduction.

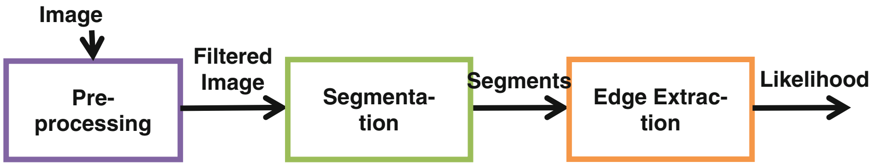


Fig. 1 Vision-based edge detection for proposing data fusion model

Defining ROI in an image is meaningful due to the following two reasons. First, the computational cost is a critical concern for perception systems, especially for real-time operations. Exclusion of unrelated parts in an image, such as the sky regions from road boundary detection, enhances the computational efficiency. Second, irrelevant parts of the image may affect the result and can lead to misdetections [5]. Noise reduction is also very important in image pre-processing. The presence of noise is inevitable in real-world situations and it increases the computational load. Image smoothing techniques, such as a median filter and Gaussian filter, are typically employed to remove image noises by smoothing the image [6].

The proposed system employs the mean shift segmentation (MSS) technique for segmentation. Mean shift is a nonparametric feature-space analysis technique that has been widely applied in the field of image processing and computer vision. It considers a given set of points as sampled data from the underlying density function and manifests the density function by detecting the local maxima of the density [7]. The MSS in this model classifies a captured image into different regions based on their textures, and the edges of the detected segments are considered as defining objects.

2.2 Lidar-Based Road Perception Systems

The most critical drawback of vision-based systems is that their efficiencies drop dramatically with insufficient illumination [8]. This fact becomes fatal when it comes to outdoor navigation systems because of the unpredictable variations of illumination. Laser-based navigation sensors have the advantage of precise range measurement capabilities under different lighting conditions [9]. The fast acquisition of large amount of perceptual data is also a key strength for the purpose of outdoor navigation [10]. For these reasons, there has been a great deal of interest in the field of robotics and UVN research using laser sensing over the past decade [5]. In order to analyse the 2D Lidar data, the procedure was divided into three key processes: data segmentation, line extraction and feature extraction, Fig. 2.

Before extracting lines, range data are segmented into different sets of points. The system employs two distinct techniques: adaptive break point detector (ABD) and iterative end point fit (IEPF). The ABD is applied to detect a break point of two consecutive points that are not close to each other. It is followed by the IEPF in order to detect corner points, Fig. 3b. Straight lines are extracted using the mean

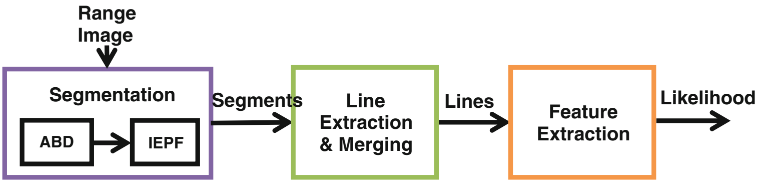


Fig. 2 Functional decomposition of the Lidar-based cue detection

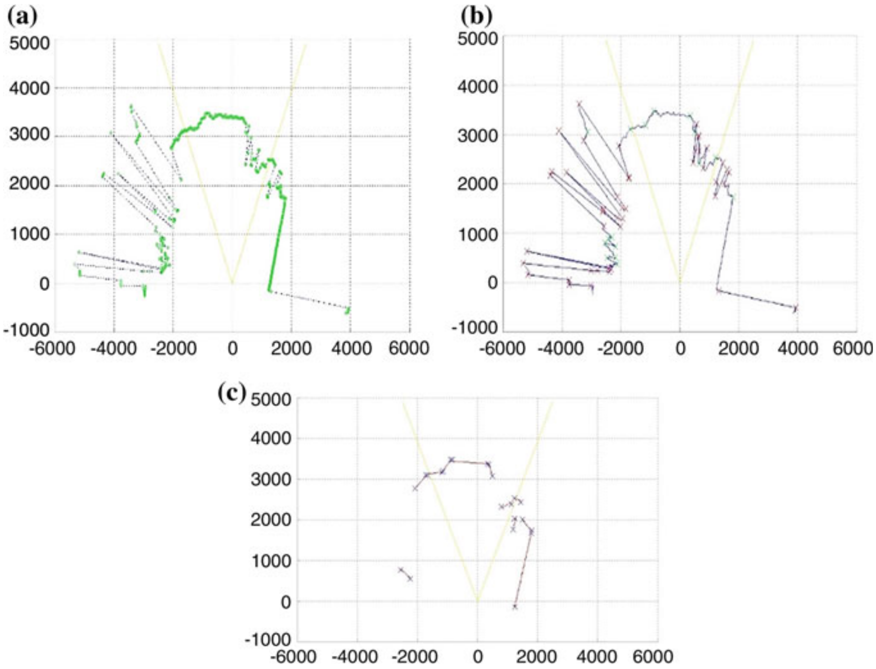


Fig. 3 Range data segmentation; **a** unprocessed range data, **b** the segmented data and **c** line extraction result. *Red* and *green* crosses indicate break points detected by ABD and IEPF, respectively. The *two yellow lines* define the camera's angle of view

squared error method [11], which is a simple line fitting method working on each segment whose number of points is greater than the threshold, Fig. 3c.

3 Likelihood Calculation

The likelihood values are generated from each ISM, using the kernel density estimation (KDE). KDE is a nonparametric technique to estimate underlying probability density function from random samples. The detected cue in each pixel, or node, is converted into a continuous density distribution as:

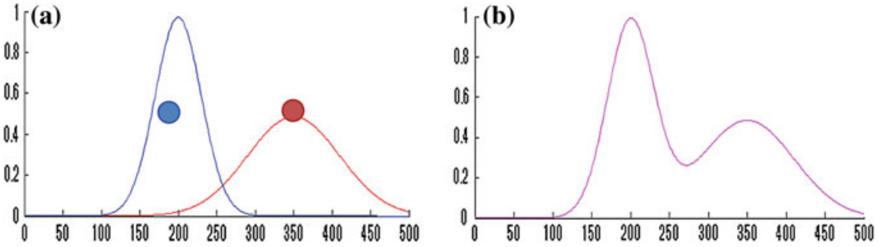


Fig. 4 Cue fusion using Kernel density estimation: **a** likelihood distributions from two different cues with different bandwidths. *Red/blue circles* indicate the positions of the cues in an ROI; **b** the calculated distribution by fusing the two cues

$$KDE(x) = \frac{1}{nh_d}\sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

For the kernel function $K(x)$, we employ the Gaussian kernel, which comprises:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

For each detected cue, distinct bandwidth $H(h_t, h_r, h_e, h_b)$ is selected according to the degree of ambiguities [12]. For instance, road curbs are salient cues for road boundaries and the locations of boundaries are relatively predictable; therefore, the density should have low variance in order to pinpoint the boundary location. In contrast, the bandwidth for ambiguous cues such as accumulative noise should be designed to be higher. Figure 4 illustrates an example of the fusion of two different cues. As shown in Fig. 4a, a lower bandwidth is given to the cue coloured blue due to the high saliency than the cue coloured red.

4 Case Study for the Road Boundary Detection

Experiments were conducted on a real road using a prototype vehicle with two distinct sensors: a camera and a Lidar. The proposed model was implemented and evaluated by its comparison with single sensor-based modes in order to investigate the effectiveness of the integrated mode. Figure 5a illustrates how the two sensors were set up in the vehicle. Both the camera and the Lidar were mounted at the tilt angle, α. They were stabilised at the heights h_C for the camera, and h_L for the Lidar, from the ground. Figure 5b shows the prototype vehicle with these sensors.

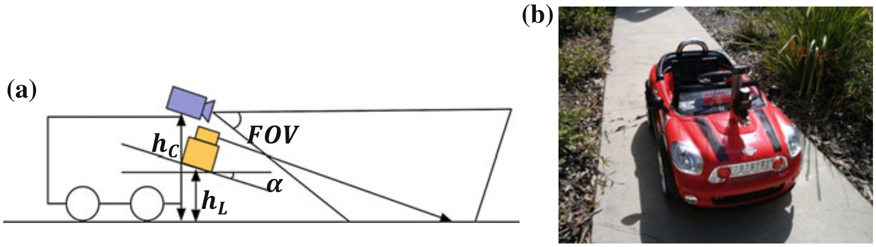


Fig. 5 A vehicle with a camera and a Lidar for the experiments. **a** Sensor set-up and **b** the prototype vehicle

4.1 Three Levels of Complexity

Experiments were carried out in a number of various road scenes at the Gold Coast campus of Griffith University. These levels were classified based on different lighting conditions, textures and road boundary types. Distinct road types such as straight, curved, flat and sloped were also considered, even though only three examples are introduced in this paper. These road scenes have been categorised into three levels according to the conditions described as follows.

- Level 1: relatively simple road situations. There are clear borders between a road surface and off-road regions, Fig. 6a.
- Level 2: boundaries are more difficult to detect due to the existence of slopes and noisy debris on the roads. There is often no distinct border between the road and the off-road, Fig. 6b.
- Level 3: these roads contain strong shadows, which make the boundaries ambiguous, or have complex road textures. The road colours are often very similar to the off-road region. Some images of this level are those captured at night-time, with insufficient illumination.

The comparisons of the experimental results were undertaken in three different modes: the vision-only mode, the Lidar-only mode and the integrated mode. The experimental results for each mode are explained as the rates of detection accuracy,

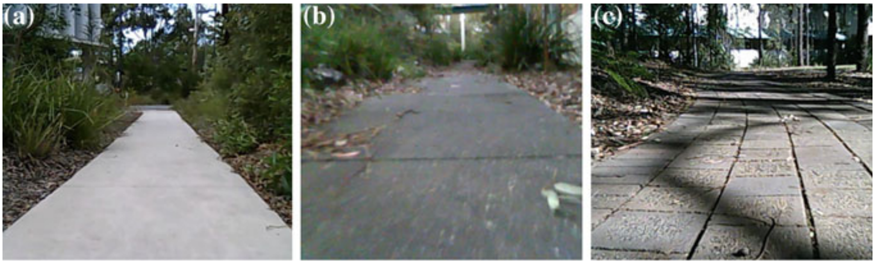


Fig. 6 Example road scenes with different complexities: **a** Level 1, **b** Level 2 and **c** Level 3

false-positive and false-negative. Distances between the detected and the actual road boundaries were measured to indicate the accuracy of the detections.

4.2 Vision-Only Mode

For the parameters used in the vision-only mode, the height of the ROI was set to 160 pixels, and a median filter with a kernel size 7 × 7 was applied. For the mean shift segmentation, the spatial bandwidth h_s and range bandwidth h_r were empirically determined as 7.0 and 5.0, respectively.

The overall results of vision-only mode in the three levels are illustrated in Table 1 and Fig. 7. Since the road scenes of Levels 1 and 2 contain high contrast between the textures of on-road and off-road regions, the regions are clearly divided. The accuracies of the detection in these levels are relatively high.

In contrast to these levels, however, both the detection rate and accuracy dropped dramatically in the Level 3. In Fig. 7c, e, the regions are not divided properly, and strong shadows caused false-positive detections. The scene in Fig. 7e does not have sufficient illumination; therefore, road boundaries became blurred.

Table 1 Experimental results of the vision-only mode

	Level 1	Level 2	Level 3
Detection rate (%)	84.16	82.25	58.03
Accuracy (cm)	6.14	6.95	11.06
False-positive (%)	166	0.80	12.5
False-negative (%)	14.16	16.93	29.48

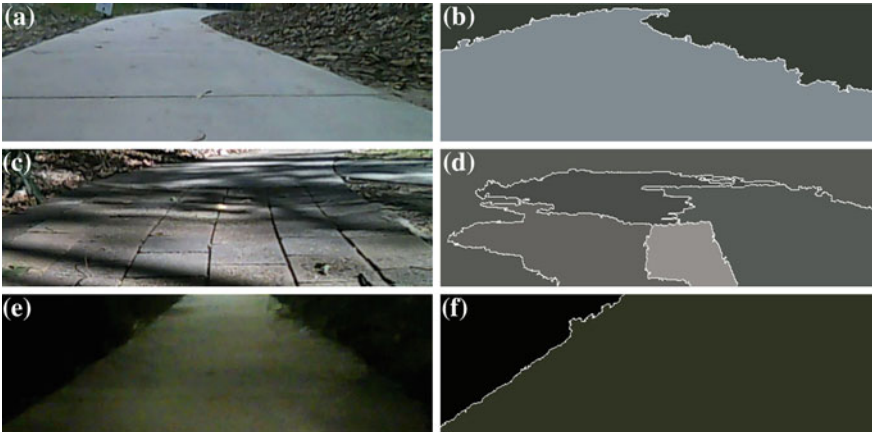


Fig. 7 Results of the vision-only mode: **a** shows high accuracy but **c** and **e** are poorly segmented

4.3 Lidar-Only Mode

The Lidar-based estimation was performed based on three cues: road surfaces, edges and cumulative break points. A different bandwidth was selected for each cue depending on the level of the ambiguity. As shown in Table 2, the overall detection rates were pragmatic in all levels. However, the average distance between the actual borders and the estimated borders was considerably larger than the vision-only modes. This is mainly due to the gaps between actual borders and the cues detected. An example of this issue is illustrated in Fig. 8.

Table 2 Experimental results of the Lidar-only mode

	Level 1	Level 2	Level 3
Detection rate (%)	89.16	82.25	85.71
Accuracy (cm)	14.33	12.88	15.18
False-positive (%)	1.66	1.61	10.71
False-negative (%)	9.16	16.12	3.57

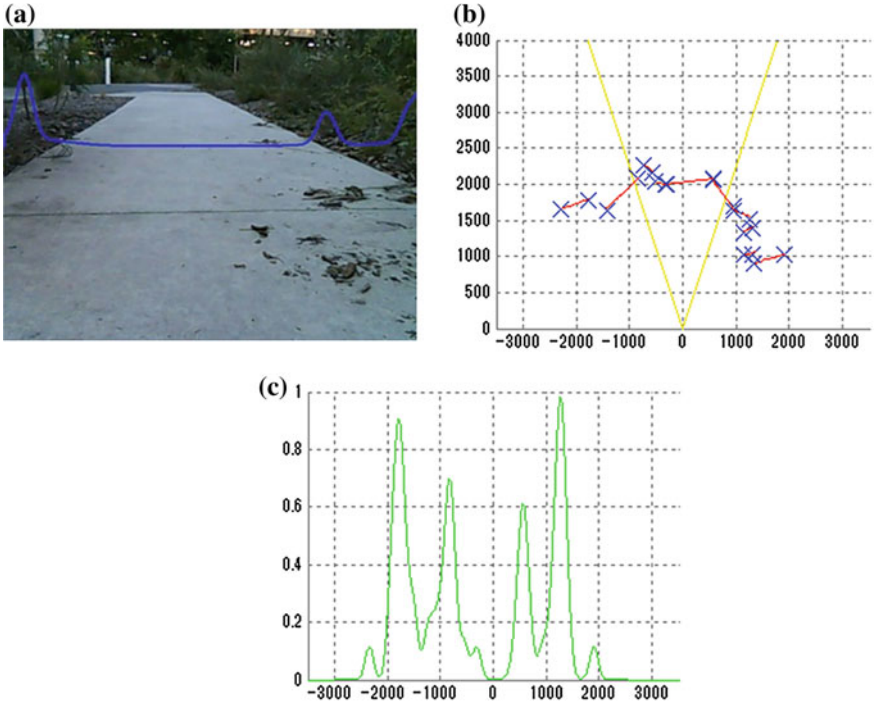


Fig. 8 Lidar-only detection results: **a** an input image with the road boundary likelihood (indicated by a *blue line*). **b** Line segments extracted from (**a**). **c** Likelihood calculated from (**b**)

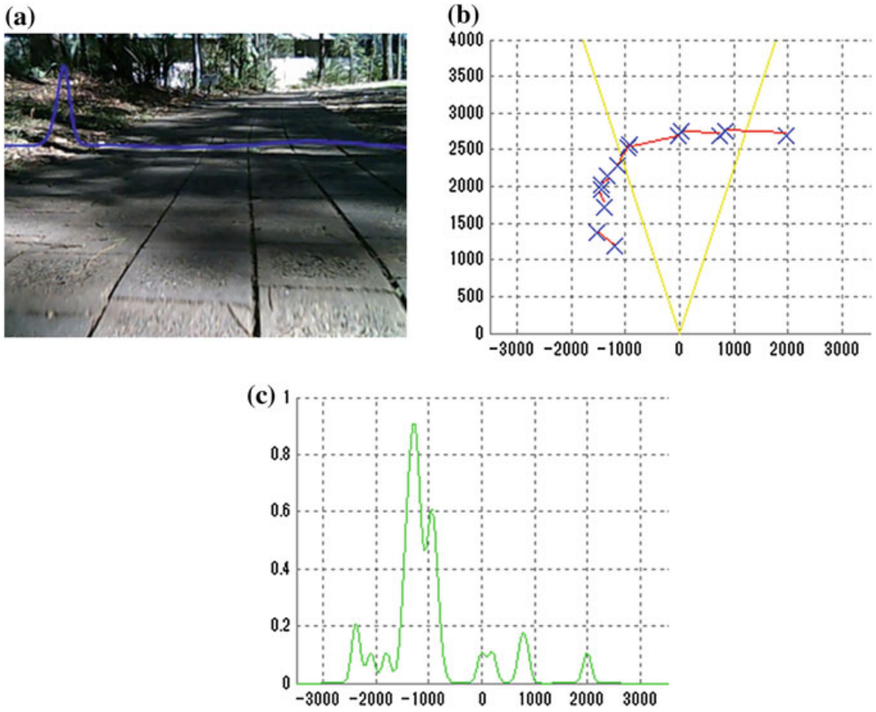


Fig. 9 Lidar-only detection results for a Level 3 case; **a** an input image with the road boundary likelihood (indicated by a *blue line*). **b** Line segments extracted from (**a**). **c** Likelihood calculated from (**b**)

Most of the false-positive detections were caused by the absences of salient cues. Cues are not likely to occur where boundaries are as flat as road regions. The right side of the road is flat in the road scene shown in Fig. 9. As a result, the likelihood for the right boundary is too small to be regarded as a boundary.

4.4 The Integrated Mode

In the integrated mode, the previous two single sensor-based detection modes are fused to enhance both the accuracy and the robustness of the estimation. Parameters are kept the same as the previous modes for each sensor's cue detection. Figure 10 shows the result produced by the integration of the two types of sensor data, camera

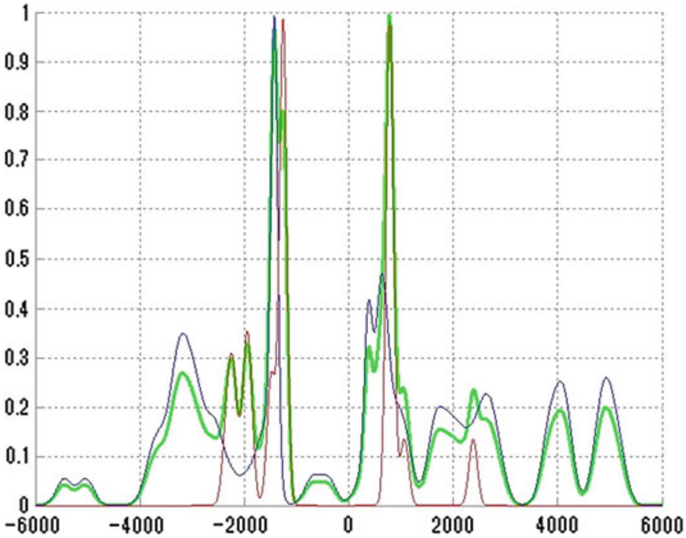


Fig. 10 The final result (the *green line*) of the road boundary detection task from the integration of a number of low-level tasks on two types of sensor data: line extraction and merging on the Lidar data (*red*), the mean shift segmentation applied to the camera image (*blue*)

Table 3 Experimental results from the integrated mode of the proposed data fusion model

	Level 1	Level 2	Level 3
Detection rate (%)	95.0	97.58	91.07
Accuracy (cm)	6.68	9.29	9.70
False-positive (%)	0.83	0.0	3.57
False-negative (%)	4.16	2.41	5.35

and Lidar. The graph clearly demonstrates the distinction of road boundaries (green) by combining the likelihood values of vision (blue) and Lidar (red).

As indicated in Table 3, the detection in all the levels resulted in higher success rates than the single sensor modes. Results for Level 3 were significantly improved relative to the vision-based mode. In Fig. 10, the red and blue lines show the road boundary likelihood obtained by the vision-based and Lidar-only modes, respectively. The positions of the road boundaries estimated by the proposed fusion mode are indicated by the yellow dots. According to the extracted boundaries, white lines are drawn to show the drivable road regions (Fig. 11).

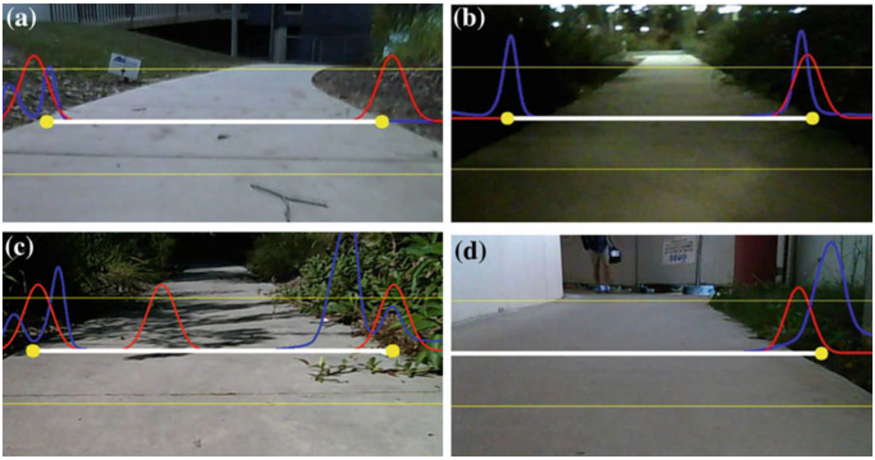


Fig. 11 Results of the data fusion mode: road boundary likelihood values obtained from the vision-only mode and the Lidar-only mode are indicated by *red* and *blue lines*, respectively. The *yellow dots* show detected road boundaries determined using the proposed fusion model. The *white lines* denote the estimated drivable regions

5 Conclusion

This paper proposed a likelihood-based integrated model for object detection systems, presenting a practical implementation with a camera and a Lidar. Cue detection models for a vision sensor and Lidar have been proposed to estimate the likelihood of road boundaries. For the vision-based detections, the mean shift segmentation was applied to extract the boundaries of road and off-road regions. For the Lidar-only approach, the ABD and IEPF were applied to detect break points and corner points. Straight lines were extracted using the mean squared error method. The cues obtained from each sensor were converted into likelihood values using the kernel density estimation.

The individual likelihood values were fused into an integrated data set and the effectiveness was observed. The experiments were conducted in a variety of road conditions. The vision-only mode performed well for the cases having enough illumination and high contrast between textures. However, it produced very poor results under Level 3 situations. The Lidar-only mode worked well under poor lighting conditions. However, it did not perform well when the road and off-road were too flat without enough difference in level. The proposed model performed well for the both conditions by utilising the strengths of the two sensors. The results from the experiments showed some promising aspects of the proposed model.

The data fusion model can accommodate any type or number of sensors with some simple modifications within the ISM. However, the way of operations in DFM is the same, and therefore, the proposed model is robust and versatile for perception.

References

1. Khaleghi, B., Khamis, A., Karray, F.O., Razavi, S.N.: Multisensor data fusion: a review of the state-of-the-art. Inf. Fusion (2011)
2. How, J.P., Bethke, B., Frank, A., Dale, D.,Vian, J.: Real-time indoor autonomous vehicle test environment. IEEE Control Syst. Mag. 51–64 (2008)
3. Subramaniana, V., Burksa, T.F., Arroyob, A.A.: Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation. Comput. Electron. Agric. **53**(2), 130–143 (2006)
4. Sun, J., Wu, Z., Pan, G.: Context-aware smart vehicle: from model to prototype. J. Zhejiang Univ. Sci. A **10**(7), 1049–1059 (2009)
5. Hillel, A.B., Lerner, R., Levi, D., Raz, G.: Recent progress in road and lane detection: a survey. Mach. Vis. Appl. 1–19 (2012)
6. Huang, A.S., Moore, D., Antone, M., Olson, E., Teller, S.: Finding multiple lanes in urban road networks with vision and lidar. Auton. Robots **26**(2–3), 103–122 (2009)
7. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell. **17**(8), 790–799 (1995)
8. Kodagoda, K., Ge, S.S., Wijesoma, W.S., Balasuriya, A.P.: IMMPDAF approach for road-boundary tracking. IEEE Trans. Veh. Technol. **56**(2), 478–486 (2007)
9. Garcia, F., Musleh, B., de la Escalera, A., Armingol, J.M.: Fusion procedure for pedestrian detection based on laser scanner and computer vision. In: 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1325–1330 (2011)
10. Morrison, R.: Fiducial marker detection and pose estimation from lidar range data. Technical Report. DTIC Document (2010)
11. Adhikari, S.P., Kim, H.: Dynamic programming and curve fitting based road boundary detection. In: 9th WSEAS International Conference on Computational Intelligence, pp. 236–240 (2010)
12. Jo, J.H., Tsunoda, Y., Sullivan, T., Lennon, M., Jo, T., Chun, Y.: BINS: blackboard-based intelligent navigation system for multiple sensory data integration. In: The 17th International Conference on Image Processing, Computer Vision, and Pattern Recognition, Las Vegas, USA (2013)

A Vision-Based 6-DOF Displacement Measurement Method for Assembling PC Bridge Structures Using a Planar Marker

Suyoung Choi, Wancheol Myeong, Yonghun Jeong and Hyun Myung

Abstract Precast concrete (PC) is used in many construction sites around the world, which is delivered after it is totally made on the plant. This is because it can reduce the rate of structural defection and building time. While in construction, many of PC materials are moved by a crane operator to specific positions just by looking. And the workers use hand gestures to fit a shear pin on the lower beam with an upper part. In this paper, a vision-guided method that is more efficient and faster than the conventional PC assembly methods is proposed. A vision-based 6-DOF displacement estimation method measures the relative displacement between the camera located in an upper slab and the marker located at the shear pin on the lower beam. Usually, PC slab has a quite long width, and it is not easy to estimate a precise 6-DOF information with just a pair of a camera and a marker. To mitigate this problem, multiple pairs of cameras and markers can be configured for large PC member and the beam. This paper deals with 6-DOF displacement estimation by using vision-based localization with a planar maker for PC construction sites. A camera detects the corner of the planar marker at first, and sub-pixel information is obtained for the corner and then the data are transferred to a main computer via Bluetooth communication. The main computer calculates 6-DOF displacement with corresponding points in the world coordinate frame. To show feasibility and robustness of the proposed method, some experiments are performed with varying distances.

S. Choi (✉) · H. Myung
Robotics Program, KAIST, Daejeon 34141, Korea
e-mail: suyoung.choi@kaist.ac.kr

H. Myung
e-mail: hmyung@kaist.ac.kr

W. Myeong · Y. Jeong · H. Myung
Department of Civil and Environmental Engineering, KAIST, Daejeon 34141, Korea
e-mail: wcmyeong@kaist.ac.kr

Y. Jeong
e-mail: yonghunjeong@kaist.ac.kr

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_40

1 Introduction

For the maintenance of civil structures, there are a lot of construction works in the urban space. From the economic point of view, it is very important to reduce construction time especially in a roadway because it also causes traffic congestion and economic losses. Precast concrete (PC) construction technology has great advantages that it can guarantee better quality and faster replacement work for damaged members with lower cost [1, 2].

PC members are manufactured from off-site factories, and they are delivered to the construction area. However, there are still remaining works. Some of workers should make an effort to control the precise position of the moving member by a crane. At that time, the crane operator only depends on other workers' gestures or voice. If a crane worker knows how closely the PC block and the beam approach each other during their rendezvous, the construction time can be reduced. Displacement measurement systems can provide a crane operator with useful information for those processes. There are lots of traditional methods for measuring the displacement such as a strain gauge, a linear variable differential transformer (LVDT), and a global positioning system (GPS). The strain gauge is quite subject to noises, and LVDT and GPS require high cost for purchasing and installing on the structural members [3]. For these reasons, a vision-based 6-DOF estimation method is proposed which consists of a vision sensor and landmarks such as planar markers. We propose the method for estimating 6-DOF with a low-price Webcam and a planar marker that can measure the relative displacement directly. This paper mainly deals with a pair of camera and marker system for 6-DOF displacement estimation between the PC block and the beam (Fig. 1). This system is implemented with a low-cost Webcam and a single-board computer (SBC). A camera installed on the upper PC captures a maker located on the lower beam. SBC connected with the camera calculates 12-pixel point coordinates of the marker and then transfers these pieces of information to a main computer for calculating relative 6-DOF estimation. The remainder of this paper is organized as follows. In Sect. 2, the relative pose estimation principle between a camera and a planar marker is demonstrated. Section 3 introduces the procedure of image processing and

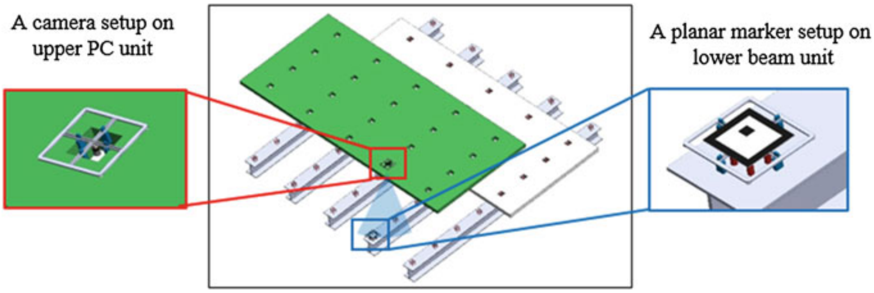

A camera setup on upper PC unit
A planar marker setup on lower beam unit

Fig. 1 The installation of a pair of a camera and a planar marker

transferring pixel data. In Sect. 4, experiments are conducted and the results are shown for validating our proposed system performance. Lastly, in Sect. 5, the conclusions are addressed.

2 Vision-Based 6-DOF Estimation with Planar Marker

For estimating relative 6-DOF related to a camera and planar marker, it can be induced from the camera model associated with extrinsic and intrinsic parameters. The extrinsic parameters show the relation between camera coordinate and world coordinate. It is expressed by a rotation matrix and a translation matrix. The intrinsic parameters include the information about the focal length, principal points, and skew coefficient of the camera. Geometric relationship of the camera and the marker is represented in Fig. 2. The matching point on the image plane from a camera coordinate and the world coordinate which is located in the marker can be presented with homography. The corresponding pixel points of the world coordinate and image coordinate are denoted by H_i and h_i ($i = 1, \ldots, N$), respectively, where N is the number of points. H_i and h_i can be defined using homogeneous vectors: $H_i = [X_i \quad Y_i \quad Z_i \quad 1]^T$, $h_i = [x_i \quad y_i \quad 1]$. For estimating 6-DOF displacement, the relation between two homogeneous vectors can be represented as follows [4]:

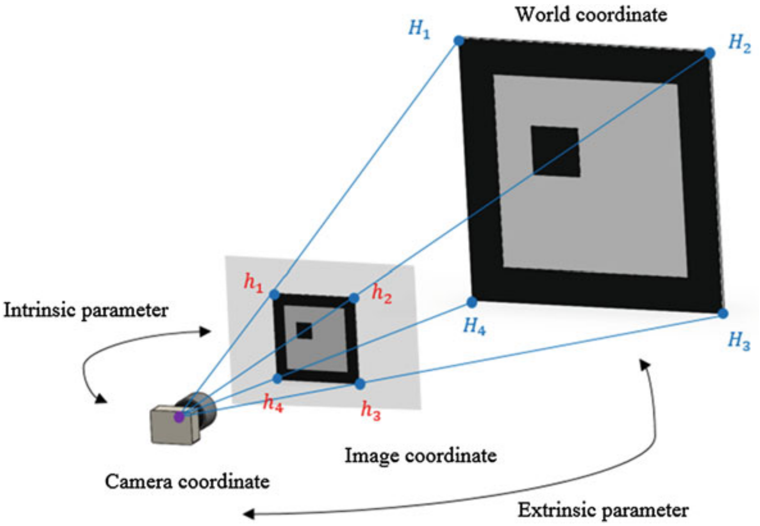


Fig. 2 Relationship among world, camera, and image coordinates

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{12} & -R_1^T T \\ r_{21} & r_{22} & r_{23} & -R_2^T T \\ r_{31} & r_{32} & r_{33} & -R_3^T T \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{1}$$

where f_x and f_y in the intrinsic parameters of the camera are focal length, and c_x and c_y are principal points of camera which are the center point of the image plane [5]. The intrinsic parameters are acquired from an extra camera calibration process. The variables r and T represent rotation matrix and translation vector, respectively. R_q ($q = 1, 2, 3$) is the row vectors of r matrix. By calculating r and T matrices with a known values of H_i, h_i and the intrinsic parameters, 6-DOF displacement can be estimated.

3 6-DOF Displacement Estimation with Pixel Data Transfer

The image processing when the camera captures the marker is conducted in the SBC. The procedure is performed in real time to detect the marker. Firstly, original images are transformed to gray image and it is converted to binary image. Binary image can be represented by digital 0 or 1 with a specific threshold value or adaptive threshold values. Consequently, total 12 edge points are extracted from three squares of the marker, and 4 edge points are extracted from one square bound that is caught by a blob algorithm in planar marker. Also, the position of the most inner square is recognized to know the yaw direction of the marker and gives the each point own id. All edge points are utilized for the matching points of the world coordinate. These marker points can be expressed more accurately and detailed pixel, called sub-pixels. In result, the marker generates 12 sub-pixels on one marker (Fig. 3).

In this paper, it has an aim to measure 6-DOF displacement between a camera which is installed on a shear pocket and a marker which is located on a shear pin when a crane driver is moving a PC member for fitting the shear pin on the beam. Since the crane driver who monitors the displacement and the PC blocks where marker and camera are installed are not close to each other, data transmission is necessary. Our system is composed of two systems, one is installed with camera capturing the marker image, extracting the sub-pixel of the edge points, and transmitting the data, and the other is located in the crane operating room and its main processor calculates 6-DOF displacement with the sub-pixel data. At this moment, 6-DOF displacement can be calculated from Eq. 1 because we already know the geometry information of the marker and the sub-pixel data which is projected in the image coordinate and intrinsic parameter.

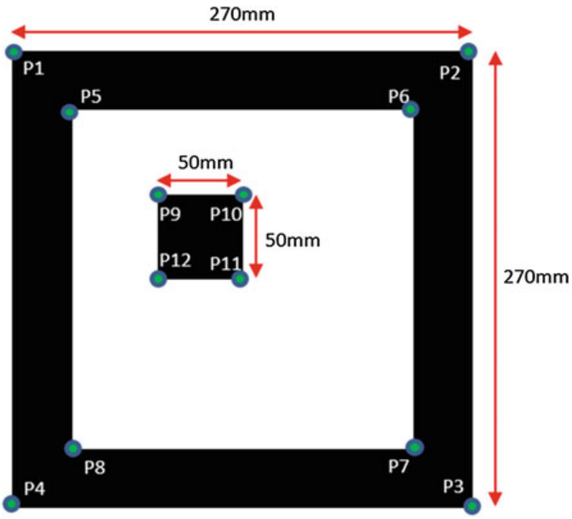


Fig. 3 12-pixel points in the marker image

4 Experimental Results

To prove the proposed system's validation, the experimental setup and results are represented in this section. The experiments are conducted within 30 m distance between two systems considering the distance between the PC member and the crane.

Figure 4 shows the configuration of the system setup. Figure 4 consists of a camera, a marker, and motorized motion stage. The camera is HD Pro Webcam C920 from Logitech, Co. with 1920 × 1080 pixels and 30 frame rates. The marker geometry is shown in Fig. 3. The motion stage provides artificial movement between the camera and the marker by giving linear translation along *X*-axis and rotation about *Y*-axis in this experiment. The distance from the camera and the marker is 50 cm. After the camera gets marker images, a SBC which is

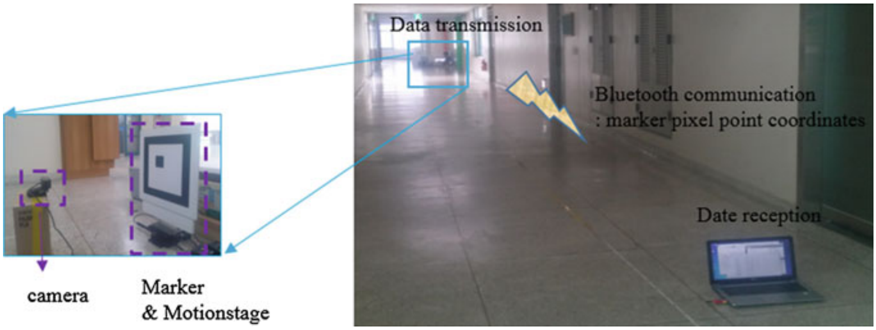


Fig. 4 Installation of experimental tests

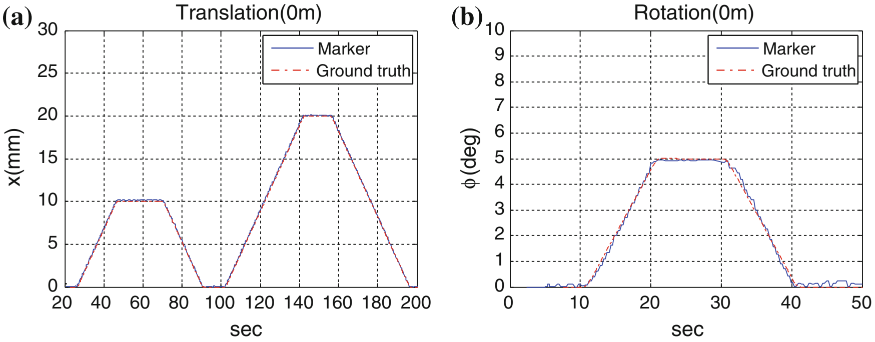


Fig. 5 Experimental result at communication distance 0 m for sub-pixels. **a** Translation 10 and 20 mm. **b** Rotation 5°

NUC5i3RYK from INTEL Co. with i3 core processor proceeds image processing. The SBC calculates 12 sub-pixel data from binary image converted from the original image, and then these pixel points are transferred to a main PC using Bluetooth. The Bluetooth system is parani-SD100U from SENA. Co with standard version 2.0 with 3 Mbps max transfer rate and 300 m max transfer distance. Finally, the main system calculates 6-DOF displacement results based on the 12 sub-pixel data on the image plane and the corresponding points on the known world plane geometry of the marker.

Marker is installed on the motion stage and moves along *X*-axis 10 and 20 mm linear translation (Figs. 5a, 6a, 7a and 8a), and about *Y*-axis 5° rotation (Figs. 5b, 6b, 7b and 8b). The root mean square error (RMSE) of the linear translation along *X*-axis is max. 0.61 mm and the error of rotational case about *Y*-axis is max. 0.26° (Tables 1 and 2). The results of 6-DOF pose estimation components have error

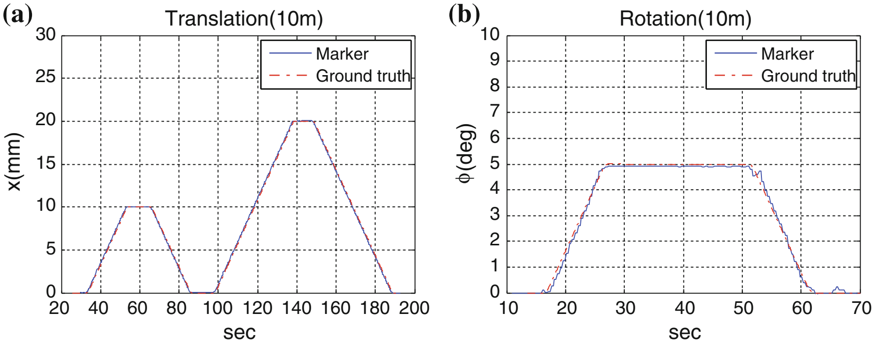


Fig. 6 Experimental result at communication distance 10 m for sub-pixels. **a** Translation 10 and 20 mm. **b** Rotation 5°

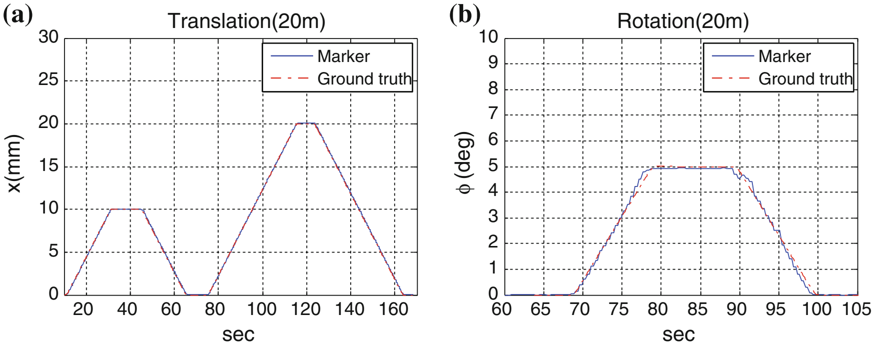


Fig. 7 Experimental result at communication distance 20 m for sub-pixels. **a** Translation 10 and 20 mm. **b** Rotation 5°

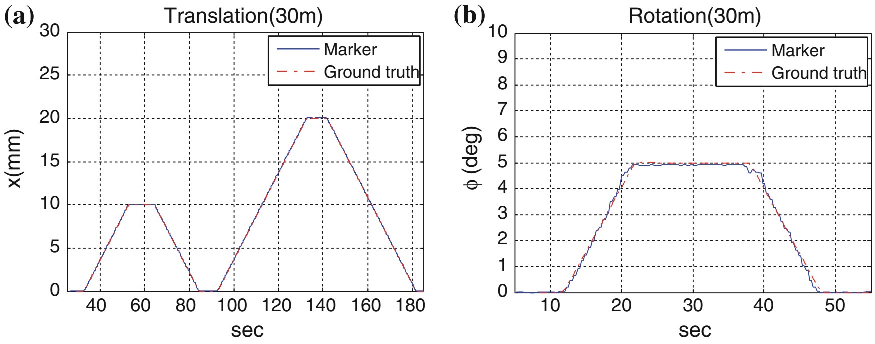


Fig. 8 Experimental result at communication distance 30 m for sub-pixels. **a** Translation 10 and 20 mm. **b** Rotation 5°

Table 1 RMSE between 6-DOF estimation results and ground truth generated from a motion stage with linear motion along *X*-axis 10 and 20 mm

	X (mm)	*Y* (mm)	*Z* (mm)	Linear av. (mm)	*θ* (°)	*ϕ* (°)	*ψ* (°)	Angular av. (°)
0 m	0.17	0.07	0.25	0.16	0.02	0.09	0.07	0.06
10 m	0.19	0.02	0.07	0.09	0.01	0.05	0.04	0.04
20 m	0.09	0.04	0.04	0.05	0.10	0.06	0.04	0.04
30 m	0.61	0.02	0.05	0.23	0.01	0.07	0.03	0.04

within a few millimeters and a few degrees compared with the motion stage that plays a role in the ground truth. The results of the translational and rotational test show that all components of 6-DOF track the ground truth quite well with low error within 1 mm and 1°.

Table 2 RMSE between 6-DOF estimation results and ground truth generated from a motion stage with linear motion about *Y*-axis 5°

	X (mm)	*Y* (mm)	*Z* (mm)	Linear av. (mm)	θ (°)	ϕ (°)	ψ (°)	Angular av. (°)
0 m	0.27	0.03	0.21	0.17	0.18	0.11	0.05	0.11
10 m	0.32	0.06	0.18	0.19	0.26	0.11	0.05	0.14
20 m	0.31	0.05	0.18	0.18	0.24	0.10	0.06	0.13
30 m	0.36	0.03	0.22	0.20	0.25	0.10	0.07	0.14

5 Conclusions

This paper proposes a planar marker and vision-based 6-DOF displacement estimation method. Once the camera gets an image of the planar marker, the single-board computer processes the image processing. After this operation, 12 sub-pixels on the image are calculated and these are transferred to the main processor by Bluetooth communication. The main system that is approximately 30 m apart from the SBC computes 6-DOF displacement with the 12 sub-pixels on the image and corresponding points in the world marker coordinate. The experiments are conducted using a motion stage for moving the planar marker like translational movement of 5 and 10 mm along X-axis, and rotational movement of 5° about Y-axis. The error compared with the motion stage is within 1 mm and 0.1°. The calculated 6-DOF displacement results can be seen on the screen of the crane worker during the construction so that he/she can adjust the PC member precisely and quickly.

Acknowledgment This research was supported by a grant (13SCIPA01) from Smart Civil Infrastructure Research Program funded by Ministry of Land, Infrastructure and Transport (MOLIT) of Korea government and Korea Agency for Infrastructure Technology Advancement (KAIA). The students are supported by Korea Ministry of Land, Infrastructure and Transport (MOLIT) under U-city Master and Doctor Course Grant Program.

References

1. Hieber, D.G., et al.: Precast Concrete Pier Systems for Rapid Construction of Bridges in Seismic Regions. Washington State Transportation Center (TRAC) (2005)
2. Tam, V.W.Y., Fung, I.W.H.: Social and environmental benefits of precast concrete technology. J. Prestressed Concr. Inst. **46**, 14–19 (2001)
3. Lee, J., Fukuda, Y., Shinozuka, M., Cho, S., Yun, C.-B.: Development and application of a vision based displacement measurement system for structural health monitoring of civil structures. Smart Struct. Syst. **3**, 373–384 (2007)

4. Lee, D.H., Jeon, H.M., Myung, H.: Vision-based 6-DOF displacement measurement of structures with a planar marker. Sens. Smart Struct. Technol. Civ. Mech. Aerosp. Syst. **8345**, 1–8 (2012)
5. Myeong, W., Choi, S., Myung, H.: Monte Carlo localization and multiple vision sensor based 6-DOF displacement measurement system for the rendezvous of PC bridge members. In: International Workshop on Structural Health Monitoring (IWSHM) (2015)

A Color Constancy Algorithm Using Photodetector Characteristics of a Camera for Indoor Scenes

Ue-Hwan Kim and Jong-Hwan Kim

Abstract According to a recent research, illumination invariant features could be available to guarantee robust and reliable performance of vision systems. The research incorporated the imaging sensor properties to the vision algorithm, which is a general case in robotics contexts. However, these features are only applicable for the outdoor scenes. This paper develops an algorithm that enhances the features reported in the previous research. The features work for the indoor scenes with the proposed algorithm. Experiments are conducted to verify the performance of the proposed algorithm. It is verified that the features enhanced by the proposed algorithm become robust to illumination fluctuations.

Keywords Illumination invariant transform · Color constancy · Indoor environments · Robotics application

1 Introduction

Illumination changes during a day and various light sources affect the performance of most vision systems, such as classification, object recognition, and localization [4, 12, 18]. Even a small change in the direction of the light source could hamper the vision system. For the robust and reliable operation of vision systems, illumination invariant imaging techniques are crucial. However, coping with a wide range of illumination conditions requires much knowledge of the environment, which results in increased computation.

Image processing algorithms based on image-point features have been proposed to deal with such illumination variations without entailing much computation. Scale

U.-H. Kim (✉) · J.-H. Kim
School of Electrical Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon,
Republic of Korea
e-mail: uhkim@rit.kaist.ac.kr

J.-H. Kim
e-mail: johkim@rit.kaist.ac.kr

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_41

invariant features (SIFT) [4], speeded up robust features (SURF) [1], and binary robust independent elementary features (BRIEF) [2] combine gradients and histograms of grayscale patches around the point of interest to generate illumination invariant features. Gross and Brajovic [8] proposed an image processing algorithm for illumination invariant face recognition. The algorithm extracts the illumination field from a single image and recovers the scene reflectance. Although these techniques overcome a certain level of illumination variations, they show some limitations for complicated situations where the direction and the spectrum of the light source vary significantly.

Incorporating the imaging sensor information improves the performance of vision algorithms. In general settings, the properties of the imaging sensor are not available. In robotics settings, however, one can fully exploit the knowledge of the sensor characteristics. By utilizing the spectral sensitivity function of a photodetector, the work of determining the color of a surface independent of the illuminant becomes simple, which is known as color constancy. Several methods [5, 11, 13] assume narrow photodetector responses in achieving color constancy. One of the methods [16, 17] approximates the spectral sensitivity function to a Dirac delta function and the sun to a blackbody. It is reported that the color constancy algorithm with the Dirac delta function approximation represents robust and improved performance for the outdoor scenes [15]. However, the method does not reveal any enhancement for the indoor scenes. For the indoor scenes, various light sources and shadows affect the camera response; hence, the assumptions made become unreliable.

This paper proposes an algorithm that enhances the conventional algorithm presented in [17]. As mentioned above, the conventional method does not effectively deal with the indoor scenes. The proposed algorithm adds supplementary processes to the conventional algorithm to make it robust to the illumination shifts even for the indoor scenes. The experiment results prove the enhanced performance of the proposed algorithm.

This paper is organized as follows: Sect. 2 describes the illumination invariant transform. In Sect. 3, the proposed algorithm is developed. Section 4 presents the experiment settings and the experimental results. Section 5 discusses the proposed algorithm. Finally, concluding remarks follow in Sect. 6.

2 Illumination Invariant Transform

Numerous factors take a role in the formation of an image, as shown in Fig. 1. The following equation illustrates the relationship between the image, $R^{x,E}$, which is the response of a camera with a spectral response function of $F(\lambda)$, and the key factors when the intensity of the illuminator with the output spectrum of $E(\lambda)$ is I:

$$R^{x,E} = \underline{a}^x \cdot \underline{n}^x I \int S^x(\lambda)E(\lambda)F(\lambda)\mathrm{d}\lambda, \tag{1}$$

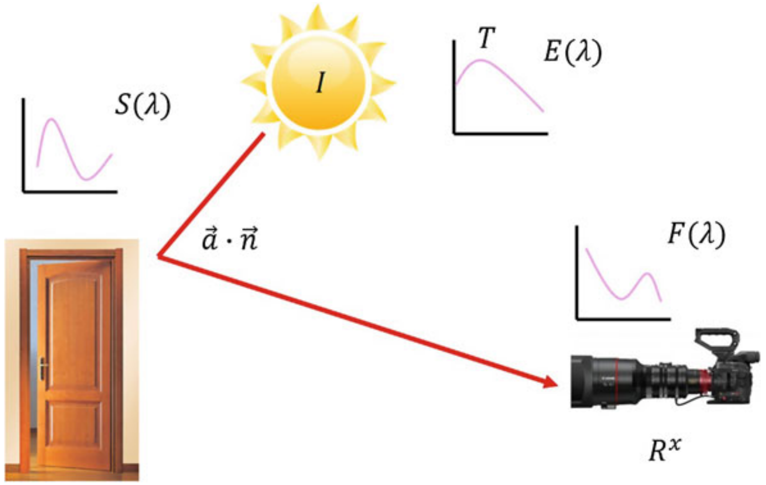


Fig. 1 Illustration of the key factors affecting the formation of an image in a scene where the sunlight is the dominant light source. The light from the sun with intensity I is reflected by an object whose surface reflectance is $S(\lambda)$ in the scene and received by a camera whose spectral sensitivity function is $F(\lambda)$. This paper models the spectral distribution of the sunlight, $E(\lambda)$, as that of a blackbody with the correlated color temperature T. The unit vectors a and n represent the geometric factor between the light source and the camera. From the knowledge of the spectral sensitivity function of the imaging sensor, $F(\lambda)$, and the response of the sensor $R^{x,E}$, the aim was to extract the material property $S(\lambda)$ without knowing other factors

where the unit vector $\underline{a}^x$ and $\underline{n}^x$ represent the direction of the light source and the direction of the surface normal of an object whose reflectance is S^x at a point x on the surface, respectively. It is reasonable to assume that the spectral sensitivity of the camera is narrow enough to be modeled as a Dirac delta function [6], where the camera is only sensitive to the light with a wavelength λ_i. Then, Eq. (1) is simplified to

$$R^{x,E} = \underline{a}^x \cdot \underline{n}^x I S^x(\lambda_i) E(\lambda_i). \tag{2}$$

By taking the logarithm of both sides of Eq. (2), different components are separated as follows:

$$log\{R^{x,E}\} = log\{GI\} + log\{E(\lambda_i)\} + log\{S^x(\lambda_i)\}, \tag{3}$$

where $G(= \underline{a}^x \cdot \underline{n}^x)$ denotes the geometric factor. Equation (3) consists of three parts: the geometry and intensity part, the surface property part, and the light source spectrum part. It is desirable to minimize the first and the third parts of Eq. (3) to visualize only the material-dependent property, $S(\lambda)$. Firstly, speculation on how to minimize the effect of the geometry and intensity part is made. The part is removed by subtracting the weighted sum of the two responses of the photodetector from the other response of the photodetector. This process creates a geometry and intensity independent feature P:

$$P = log(R_2) - \{\alpha log(R_1) + (1-\alpha)log(R_3)\}. \tag{4}$$

Next, the results from the studies which have shown that one could approximate the spectrum of the sunlight to that of a blackbody are utilized for the minimization of the third part [9, 10]. Using Planck‘s equation, the spectrum of a blackbody with a parameter known as the color correlated temperature (CCT), T, becomes

$$E(\lambda, T) = \frac{2hc^2}{\lambda^5}\frac{1}{e^{hc/k_B T\lambda} - 1}, \tag{5}$$

where h is the Planck‘s constant, k_B is the Boltzman‘s constant, and c is the speed of light. By applying the Wien approximation [7], Eq. (5) reduces to

$$E(\lambda, T) \approx c_1 \lambda^{-5} e^{-c_2/T\lambda}, \tag{6}$$

where $c_1 = 2hc^2$ and $c_2 = h_c/k_B$. Substituting Eqs. (6) into (4) gives

$$P \simeq log(c_1{\lambda_2}^{-5}S_2) - \{\alpha log(c_1{\lambda_1}^{-5}S_1) + (1-\alpha)log(c_1{\lambda_3}^{-5}S_3)\} \\ + \frac{c_2}{T\lambda_2} - \{\alpha\frac{c_2}{T\lambda_1} + (1-\alpha)\frac{c_2}{T\lambda_3}\}. \tag{7}$$

The feature P becomes independent of the CCT as well if

$$\frac{c_2}{T\lambda_2} - \{\alpha\frac{c_2}{T\lambda_1} + (1-\alpha)\frac{c_2}{T\lambda_3} = 0, \tag{8}$$

which simplifies to

$$\frac{1}{\lambda_2} = \frac{\alpha}{\lambda_1} + \frac{(1-\alpha)}{\lambda_3}, \tag{9}$$

which suggests α is uniquely determined with the simple knowledge of the peak spectral responses of each photodetector.

Originally, it is postulated that one feature space would be insufficient to distinguish a particular color [17]. However, we follow the approach of [15] which has reported that the one feature-based color space is sufficient to differentiate most materials in natural scenes.

3 Algorithm for Indoor Scenes

As stated in [15], the algorithm described in Sect. 2 does not work for the indoor scenes. The reason for this is that the major assumption made in developing the algorithm does not hold for the indoor cases. The sunlight is no longer the only light source and the photodetector response model cannot be simplified using the blackbody spectrum approximation. We propose a modification of the model to cope with indoor environments. We firstly include other possible light sources and then Eq. (2) becomes

$$R^{x,E} = GIS^{x}(\lambda_i)E(\lambda_i) + G'I'S^{x}(\lambda_i)E'(\lambda_i), \tag{10}$$

where the second term of the right-hand side denotes the effect of other light sources. This equation can be simplified in a product form by defining a disturbance constant k:

$$R^{x,E} = GIS^{x}(\lambda_i)E(\lambda_i)k. \tag{11}$$

It can be viewed that each spectral component of the sunlight corresponding to the components of the spectral sensitivity function of the imaging sensor is disturbed by the amount k. Applying the illumination invariant transform developed in Sect. 2 to images obtained at different moments results in

$$\begin{aligned} Time1 &: F_{t1} = log(R_2) - \{\alpha log(R_1) + (1-\alpha)log(R_3)\} + k_1 \\ Time2 &: F_{t2} = log(R_2) - \{\alpha log(R_1) + (1-\alpha)log(R_3)\} + k_2 \end{aligned} \tag{12}$$

Substituting the common component with P and using the matrix formulation for Eq. (12) generates

$$\begin{bmatrix} F_{t1} \\ F_{t2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} F \\ k_1 \\ k_2 \end{bmatrix}. \tag{13}$$

This under-determined system can be solved in various ways. We adopt the most simple and straightforward method, the least mean square. Then, the solution is obtained as follows:

$$\begin{bmatrix} F \\ k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{T} \left(\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{T} \right)^{-1} \begin{bmatrix} F_{t1} \\ F_{t2} \end{bmatrix}. \tag{14}$$

4 Experiment

To verify the performance of the proposed algorithm, we conducted carefully designed experiments. We recorded the response of a photodetector for the same scene every three hours in a day. The data for the experiment were acquired at 10 am, 1 pm, 4 pm, and 7 pm, respectively. We used the camera of a smartphone and no image processing technique was included in the acquisition of the data. The spectral sensitivity function of the camera was assumed to be the same as that of iphone 5 s [3]. The estimated α value is 0.4591 and the peak wavelengths are 450, 540, and 650 nm. Both outdoor and indoor scenes were considered for comparison. For

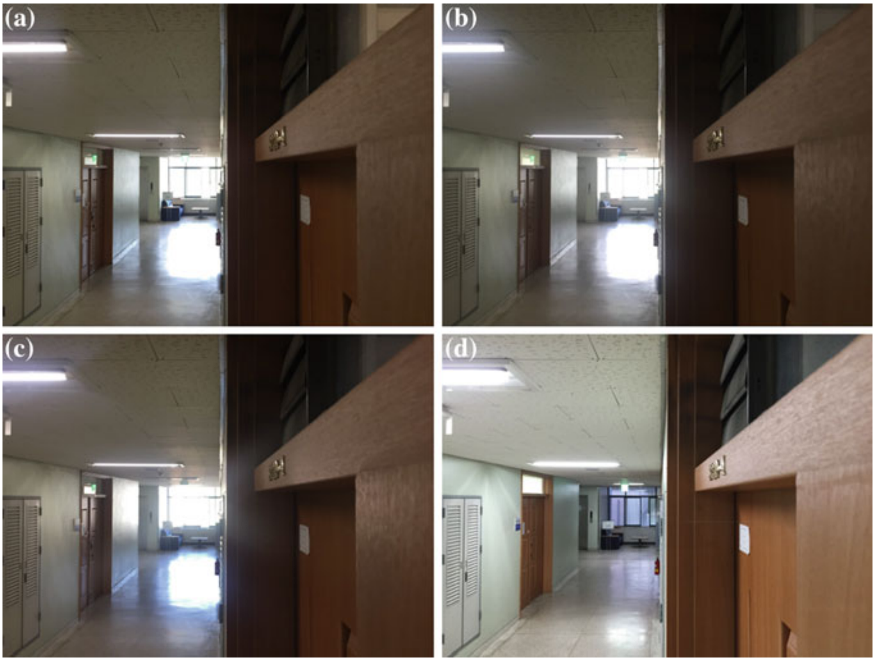


Fig. 2 Original images used for the experiment. **a** Indoor scene taken at 10 am. **b** Indoor scene taken at 1 pm. **c** Indoor scene taken at 4 pm. **d** Indoor scene taken at 7 pm

the outdoor scene, the sunlight was the only light source, whereas fluorescent lights appeared in the indoor scene all the times.

Figure 2 shows the original images of the indoor scene taken at different times during a day. As time passes, the direction of the sunlight changes and the illumination condition also varies accordingly. One can easily see that it is necessary to deal with the illumination variation during a day before processing the data for a computer-vision application.

Figure 3 displays the experimental results of the proposed algorithm. The illumination variations along the time series of the photodetector response are dramatically reduced. Compared to the original data set, the processed images are much more suitable for vision-based applications. In the following studies, the enhancement of the vision systems with the proposed algorithm should be verified. Such systems include the simultaneous localization and map building (SLAM) systems.

Figure 4 compares the proposed algorithm with the conventional algorithm. The difference is not obvious to bare eyes. This leads to a quantitative study of both algorithms. For the quantitative comparison, the sum of absolute difference (SAD) metric was employed. The pixel-by-pixel differences among the time series data were averaged for each data set. The proposed algorithm approximately reduces SAD of the conventional algorithm about thirty percent.

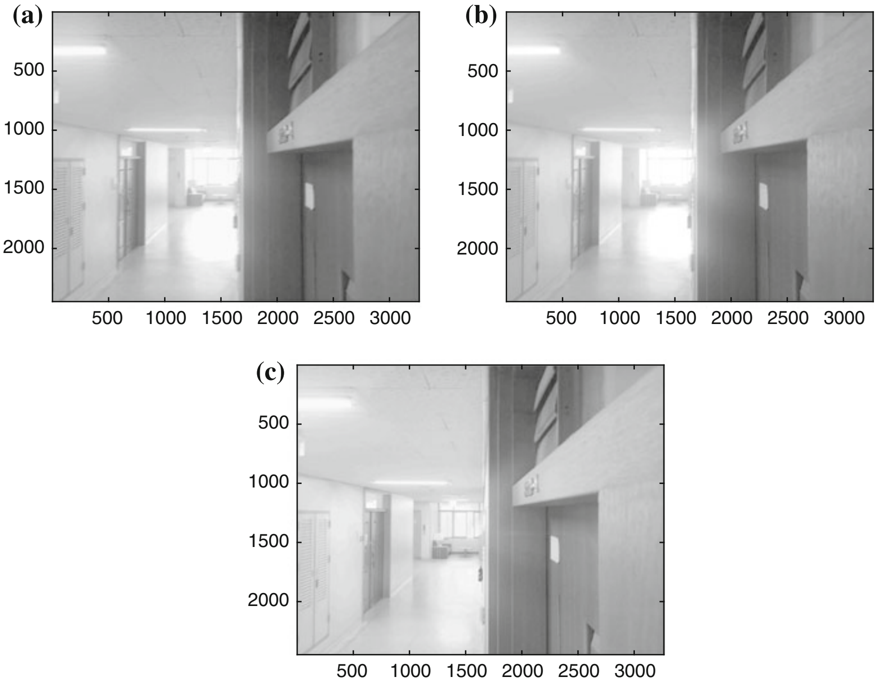


Fig. 3 The experiment results of the proposed algorithm. **a** The first output. **b** The second output. **c** The third output

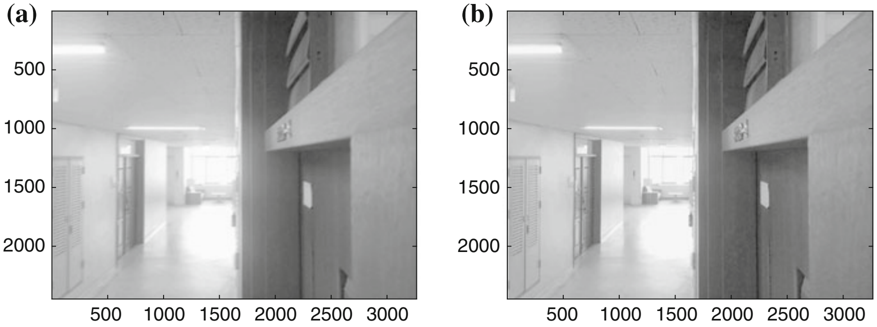


Fig. 4 Comparison of the experimental results of the proposed algorithm and the conventional algorithm. **a** The result of the proposed algorithm. **b** The result of the conventional algorithm

5 Discussion

Although the proposed algorithm improves the conventional algorithm, there are still rooms for further enhancements. First of all, the light spectrum model should be generalized to include any kinds of light sources. The current model only works for the cases where the light from the sun exists. The proposed algorithm cannot handle the indoor environments where there is no sunlight. One possible remedy is modeling the dominant light source as a Gaussian function. Then, the other minor light sources could be considered with the light disturbance parameter defined in this paper.

Secondly, other methods can be utilized to solve the under-determined system. In this paper, the simple and straight forward least mean square was employed. However, it shows some limitations as this method does not consider any possible constraints. After contemplating possible constraints for the finer formulation of the model, the under-determined system could be solved by optimization techniques. This would result in solutions closer to the true value.

Another thing to point out is related to the experiment methodology adopted in this study. The experiment was conducted in a minimal way for the verification of the performance of the proposed algorithm. The enhanced algorithm was tested in the environments where only a few parameters that could affect the results were varied. The data set collected for the experiment should have included more situations for thorough analysis of the proposed algorithm. Diverse weather conditions and different types of light sources should be considered in the further study.

Next, the proposed algorithm can be utilized for a real-time system such as a robotics system. In such a system, the operating agent should be able to recognize whether the scene the agent is looking at is what the agent has already learned. To realize this, the algorithm needs some modifications in a way that it can adaptively process the current scene and find out whether the current scene matches with a scene in its database. This modification would be made without entailing complicated thoughts.

Finally, we consider conceivable applications. As an illumination invariant transform is required for almost all of the vision systems, numerous types of vision systems can adopt the algorithm developed in this paper. One example is localization system. Recently, localization techniques are drawing much attention from robotics academy since it broadens the application area of robotics. The algorithm enhances the performance of the localization systems for both outdoor and indoor environments.

6 Conclusion

Illumination invariant feature plays a crucial role in the performance of most vision systems. Many attempts were made to achieve a true illumination invariant feature. One of the studies had accomplished this goal by making two assumptions: The sunlight spectrum can be approximated to that of a blackbody and the spectral response function of a imaging sensor is narrow enough to be modeled as a Dirac delta function. However, the existing algorithm was is only applicable to the outdoor scenes. The algorithm proposed in this paper has extended the conventional study. By including other light sources to the image formation model, the conventional algorithm has become applicable to the indoor environments. Experiments have been conducted to verify the performance of the proposed algorithm. The algorithm has shown improved performance although it requires further enhancement. In addition, future research issues have been discussed.

Acknowledgments This work was partly supported by Institute for Information& communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0101-15-0551, Virtual Creatures with Digital DNA) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2014R1A2A1A10051551).

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Computer VisionECCV 2006, pp. 404–417. Springer (2006)
2. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: binary robust independent elementary features. In: Computer VisionECCV 2010, pp. 778–792. Springer (2010)
3. Catalin, M.: Mobile phone camera possibilities for spectral imaging (Master's thesis, University of Eastern Finland, Jeonsuu, Finland). Retrieved from http://epublications.uef.fi/pub/ (2014)
4. Dissanayake, M.W.M.G., Newmau, P., Clark, S., Durrant-Whyte, H.F.: A solution to the simultaneous localization and map bnilding (SLAM) problem. Trans. IEEE Robot. Autom. **17**(3) (2001)
5. Ebner, M.: Color Constancy, Wiley Series in Imaging Science and Technology. Wiley (2007)
6. Finlayson, G.D., Hordley, S.D.: Color constancy at a pixel. JOSA A **18**(2), 253–264 (2001)
7. Finlayson, G.D., Hordley, S.D.: Color constancy at a pixel. J. Opt. Soc. Am. A **18**, 253–264 (2001)
8. Gross, R., Brajovic, V.: An image preprocessing algorithm for illumination invariant face recognition. In: Proceedings of the AVBPA, pp. 10–18 (2003)
9. Henderson, S., Hodgkiss, D.: The spectral energy distribution of daylight. Br. J. Appl. Phys. **14**(3), 125 (1963)
10. Hernandez-Andres, J., Romero, J., Nieves, J.L., Lee, R.L. Jr., et al.: Color and spectral analysis of daylight in Southern Europe. JOSA A **18**(6), 1325–1335 (2001)
11. Horn, B.K.P.: Determining lightness from an image. Comput. Graph. Image Process. **3**, 277–299 (1974)
12. Jedynak, B., Fleuret, F.: 3D Object recognition from geometric queries. In: Proceedings of the ImageCom 96, Bordeaux, France (1996)
13. Land, E.H., McCann, J.J.: Lightness and retinex theory. J. Opt. Soc. Am. **61**, 1–11 (1971)
14. Lowe, D.G.: Object recognition from local scale-invariant features. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999)
15. Maddern, W., Stewart, A.D., McManus, C., Upcroft, B., Churchill, W., Newman, P.: Illumination invariant imaging: applications in robust vision-based localisation, mapping and classification for autonomous vehicles. In: Proceedings of Workshop on Visual Place Recognition in Changing Environments, IEEE International Conference on Robotics and Automation (ICRA) (2014)
16. Ratnasingam, S., Collins, S., Hernandez-Andres, J.: Optimum sensors for color constancy in scenes illuminated by daylight. JOSA A **27**(10), 2198–2207 (2010)
17. Ratnasingam, S., Collins, S.: Study of the photodetector characteristics of a camera for color constancy in natural scenes. JOSA A **27**(2), 286–294 (2010)
18. Tuzel, O., Porikli, F., Meer, P.: Region covariance: a fast descriptor for detection and classification. In: Proceedings of the Ninth European Conference on Computer Vision (ECCV 06), vol. 2, pp. 589–600 (2006)

Real-Time Ball Detection and Following Based on a Hybrid Vision System with Application to Robot Soccer Field

Taher Abbas Shangari, Faraz Shamshirdar, Bita Azari, Mohammadhossein Heydari, Sourosh Sadeghnejad and Jacky Baltes

Abstract Circle detection is one of the most important problems in image processing since circular objects, such as balls, are observed frequently in many natural and artificial environments and problems such as light variations, occlusions, shadows, circle-shaped objects, and real-time processing have to be managed. The previous methods are either edge-based which generally suffer from processing burden or color-based and cannot deal with color variations suitably. This paper presents a real-time ball detection framework that uses both color information and shape information together to detect and track a ball robustly. The results demonstrate superiority of the proposed method over the previous. After classifying the color space, image segmentation, building regions, and extracting objects, to reduce the computation, it is focused on finding the green horizon to eliminate a part of the image, which is beyond the field border. Afterward some filters based on circle-fitting methods, image moments are applied to detect and track the ball. The results show that these filters are real-time, robust against occlusion and are able to track the ball even for distances more than six meters.

Keywords Circle detection · Real-time ball detection · Color and shape information · Circle-fitting method

T.A. Shangari · F. Shamshirdar · B. Azari · M. Heydari · S. Sadeghnejad (✉)
Bio-Inspired System Design Laboratory, Mechanical Engineering Department,
Amirkabir University of Technology (Tehran Polytechnic), No. 424, Hafez Ave.,
P. O. Box 15875-4413, Tehran 158754413, Iran
e-mail: s.sadeghnejad@aut.ac.ir
URL: http://autman.aut.ac.ir

J. Baltes
Autonomous Agents Laboratory, University of Manitoba, Winnipeg R3T 2N2, Canada
URL: http://aalab.cs.umanitoba.ca

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_42

1 Introduction

RoboCup is an important challenge in robotics. It acts as an important benchmark in developing useful and practical humanoid robots for society. An ongoing concern for RoboCup is the processing costs of the developed robotic frameworks [1, 2]. Vision algorithms are one of the most costly parts [3–5] of a humanoid robot framework.

Object detection and more especially ball detection are the most important problems in RoboCup soccer leagues [6–9]. Ball detection is defined as extracting ball candidates from several consecutive frames using color, shape, and size cues. Ball tracking is considered as applying some filters to track the motion of the ball in subsequent frames robustly.

Being able to accurately and robustly determine position and size of the balls and track them, may be the most critical issue in robot soccer games. These issues make the robot soccer games difficult by limited processing resources, which results in an inability to process a high-resolution image. Moreover, the ball is too small to be extracted from images only using color information.

There are some publications stating the problem of detecting and tracking balls. A large number of methods for ball detection and ball tracking have been studied in recent years in robotic fields. Color information and structural features are used to detect ball in [10]. In [4], the authors used Viola and Jones's algorithm for ball detection without color information. In spite of its popularity, the large amount of storage and computing power required by the Hough Transform, Circular Hough Transform, and Probabilistic Hough Transform and so on and the inaccuracy in the case of excessively noisy images are the major disadvantages of using it in real-time applications. Circle-extraction methods that have been developed also can be used to ball detection and tracking. Finally, many of the proposed algorithms work effectively but unfortunately suffer from two problems: Some are not suitable for real-time applications and some else do not work robustly.

Considering the mentioned problems, this paper presents a computationally efficient and robust system to track a ball by using color information and shape specifications from video images. A bunch of filters for ball detection and ball tracking is devised. In order to choose the fast, robust, economic filters for ball detection and tracking, the most well-known circle detectors such as Circular Hough Transform, Probabilistic Hough Transform, Levenberg-Marquardt least squares circle fitting [11], and Newton-Taubin [12] were tested and compared to each other. Two bunches of experiments were designed to test the proposed algorithm: The first bunch is applied on a stationary ball and the second on a ball moving through the field.

Hardware specification is composed of a Pentium IV 1.7 GHz, which is responsible for video acquisition of a monocular camera at a frame rate of 30 Hz, processing the video stream, image segmentation, object detection, and tracking. Because the computational resources needed to process and rectify the original images are limited, we developed a novel ball detection and tracking algorithm that consists of two types of filters.

The rest of the paper is organized as follows: First, our system classifies the color space, then segments the image, and unifies these segments. The second step is about building regions and extracting objects. In the third step, it is focused on finding the green horizon to eliminate a part of the image, which is beyond the field border. Finding green horizon, results in reduction of computation, needed for image processing. The previous operations are mentioned as a preprocessing step in the following. The next step is applying some filters to detect and track the ball. These filters work properly even if the ball is occluded or is not seen in several consecutive frames.

2 Preprocessing

To reduce the computation, needed for image processing, YUYV is chosen as the video stream format to work on, because it is supported by v4l2 Linux library and most of cameras. It allows reduced bandwidth for chrominance components, thereby typically enabling transmission errors or compression artifacts to be more efficiently masked by the human perception than using a "direct" RGB representation. Also it has a separate channel (Y or luma) to transmit luminance so light varying that is an annoying phenomenon in the RoboCup soccer fields would be under control.

Processing a 3-dimensional (3D) space is very resource consuming so a modified version of KD-tree algorithm is used to map colors from the 3-dimensional color space C to a set of colors S. A class label li ∈ S is assigned to every pixel Pi ∈ C. Each color component is represented by an 8-bit value and k = |C| = 5 stands for the number of defined class labels.

$$\begin{gathered} \mathrm{C} \rightarrow \mathrm{S} \\ \text{and } \mathrm{S} = \{\text{Green, White, Yellow, Blue, Orange}\}. \end{gathered}$$

The mapped image is segmented using vertical scan lines at first and then building the regions by unifying the segments. Considering current processing limitations, processing every pixel of a 0.48-megapixel image will result in significant frame-rate reduction. To prevent frame-rate reduction, only the pixels along a set of vertical scan lines are considered for initial processing. Doing so enables us to prevent camera resolution reduction and, as a result, increases the performance of later processes such as segmentation, object detection, object tracking, and localization. Having robot kinematics data, generating the scan lines equidistantly on the field plane is an option. Another way is to space the scan lines equidistantly on the image plane that can be done using a binary tree.

The pixels along the scan lines are traversed until a certain threshold of consecutive particular pixels is exceeded that is named color transition. In the following, the same process is done to detect the next color transition. The information stored in each color transition includes the (x, y) image coordinate of start color class label and end color class label. A segment is built up using these two pixels and the corresponding pixels in the next scan line.

The previous process reduces the segment resolutions. After unifying the related segments to build regions, the region bounds are extended; i.e., pixels that have the same label as a region and also are attached to the regions are added to the region.

After segmenting the image and building regions, green horizon determination will be done. Determining the green horizon reduces the computation needed for image processing, significantly. The robot will be able to look for the ball, the obstacles, and junctions beneath the horizon. Doing so needs processing the image from top of the image until a certain threshold of consecutive green pixels is exceeded. These green pixels are considered as the green horizon. The first feature, used to classify the regions, is color of the regions; e.g., white regions which have high eccentricity are candidate to be considered as a ball segment.

3 Image Filters for Ball Detection and Tracking

The monocular object detection and tracking system described in this paper is based on a hybrid vision system, formed by a color-based vision sub-system and a shape-based vision sub-system. It extracts balls based on color information, shape specifications, motion constraints, and difference images. The above-mentioned vision system can analyze the environment surrounding the robot, both in and out of the soccer field, even under varying lighting. The proposed approach consists of two major types of filters, categorized as world-coordinate filters and image-coordinate filters. World-coordinate filters are those which are applied on the 3D ball features and image-coordinate filters are those which are applied on 2D ball features. In the following, we introduce maximum distance and IrWrDiff as the world-coordinate filters. Newton-Taubin and image moments are introduced as the image-coordinate filters.

The ball detection and tracking scenario are as follow: The input image is preprocessed and the blobs are extracted. Retained blobs will be extended, features of the extended blobs are extracted, features are weighted, and then a parameter named validity is calculated based on the features and their corresponding weights. The output of mentioned processes is ball candidates. Ball candidate with the highest validity is taken as the best candidate; if validity of the best candidate is greater than a threshold, then it will be considered as a ball.

In the following, the filters are explained: Newton-Taubin is one of the filters that is used for ball detection. Applying Newton-Taubin on ball candidates, we get three parameters as the output: the blob center, blob radius, and standard deviation of the region. As the difference of mass centers is very small, it is divided by the radius of the ball candidate and then multiplied by its corresponding weight. These parameters are compared with the real ball and this filter is weighted based on the difference of these parameters.

Another filter is named aspect ratio; the closer to 1, the more weight it gets in calculating validity. To get this feature, the blob height is divided by blob width. The next is eccentricity filter. Eccentricity of every ball candidate is calculated and multiplied by its corresponding weight.

The next is maximum-distance filter. When mapping position and size of a ball candidate in the image onto that on the field, if the distance of the ball is less than MaxDis then it remains as a candidate, it will be removed from the candidates. MaxDis is distance of the robot to farthest point of the field border.

IrWrDiff filter deals with verifying the ball candidates mapped onto the field. It maps position and size of the ball candidate in the image onto that on the field and compares ball candidate size with the actual ball size. If the difference is less than a threshold, the ball candidate is retained; otherwise, it is not accepted as an actual ball.

The closer the ball is, the more weight it has in calculating validity.

4 Experiments

To test the proposed system, two experiments are designed. The first is to test robustness of the algorithm in the presence of occlusion, and the second is to examine how far the algorithm is able to detect the ball.

4.1 Robustness in the Presence of Occlusion

The algorithm is tested in various conditions and showed promising robustness in the presence of occlusion. Figure 1 demonstrates that even if 70 % of the ball is occluded the algorithm works properly.

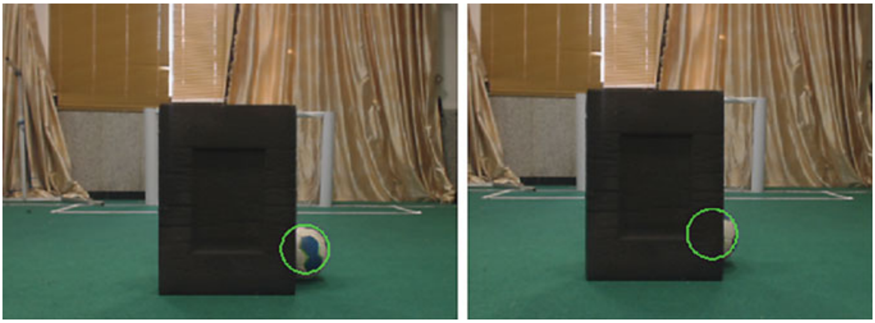


Fig. 1 The *left* image shows an about 30 % occluded ball that is detected by the proposed system and in the *right* image about 70 % of the ball is occluded

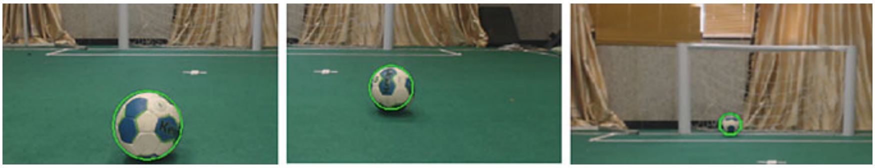


Fig. 2 From *left* to *right*, it is shown that the ball is detected for near, mid-, and far distances

4.2 Detection of Distant of the Balls

This section of experiments focuses on testing that how far the system is able to detect balls. Figure 2 shows that the system is able to detect the ball for near and far distances.

5 Conclusion

The algorithm system is economically implemented for real-time operation. Experimental results on real-time image sequences demonstrate that the proposed scheme is promising and effective. The system is able to detect and track the ball robustly even if the ball is occluded up to about 70 % or even more. Moreover, the system detected the ball in near and far distances up to 6 meters. The proposed algorithm for ball detection and tracking will be used in humanoid robot league of RoboCup 2015 competition in Hefei, China, as a robust and real-time system in the AUT-UofM Humanoid TeenSize joint team from Amirkabir University of Technology (Tehran Polytechnic), Iran and University of Manitoba (Canada).

Acknowledgment The authors would like to thank the Humanoid and Medical Robotic Laboratory (HuMeR Lab) of Amirkabir Robotic Institute of Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, and also Autonomous Agents Laboratory, University of Manitoba, Winnipeg, Canada. This research has been done in collaboration with both universities under the corporation protocol, started from 2014.

References

1. Baltes, J., Sadeghnejad, S., Seifert, D., Behnke, S.: RoboCup humanoid league rule developments 2002–2014 and future perspectives. In: The 18th Annual RoboCup International Symposium, Joao Pessoa, Brazil (2014)
2. Shangari, T.A., Shamshirdar, F., Heydari, M.H., Sadeghnejad, S., Baltes, J., Bahrami, M.: AUT-UofM humanoid TeenSize joint team; A new step toward 2050's humanoid league long term RoadMap. In: Paper Presented at the 3rd International Conference on Robot Intelligence Technology and Applications, Beijing, China (2014)

3. Mitri, S., Frintrop, S., Pervolz, K., Surmann, H., Nuchter, A.: Robust object detection at regions of interest with an application in ball recognition. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on 2005, pp. 125–130. IEEE (2005)
4. Treptow, A., Zell, A.: Real-time object tracking for soccer-robots without color information. Robot. Auton. Syst. **48**(1), 41–48 (2004)
5. Bader, M., Albero, M., Sablatnig, R., Simó, J.E., Benet, G., Novak, G., Blanes, F.: Embedded real-time ball detection unit for the yabiro biped robot. In: Intelligent Solutions in Embedded Systems, 2006 International Workshop on 2006, pp. 1–9. IEEE (2006)
6. D'Orazio, T., Ancona, N., Cicirelli, C., Nitti, M.: A ball detection algorithm for real soccer image sequences. In: Pattern Recognition, 2002. Proceedings. 16th International Conference on 2002, pp. 210–213. IEEE (2002)
7. Illingworth, J., Kittler, J.: A survey of the Hough transform. Comput. Vis. Graph. Image Proc. **44**(1), 87–116 (1988)
8. Leavers, V.: Which hough transform? CVGIP: Image Underst. **58**(2), 250–264 (1993)
9. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Commun. ACM **15**(1), 11–15 (1972)
10. Gönner, C., Rous, M., Kraiss, K.-F.: Real-time adaptive colour segmentation for the robocup middle size league. In: RoboCup 2004: Robot Soccer World Cup VIII, pp. 402–409. Springer (2005)
11. Budden, D., Fenn, S., Walker, J., Mendes, A.: A novel approach to ball detection for humanoid robot soccer. In: AI 2012: Advances in Artificial Intelligence, pp. 827–838. Springer (2012)
12. Taubin, G.: Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **13**(11), 1115–1138 (1991)

Toward Autonomous UAV Landing Based on Infrared Beacons and Particle Filtering

Vsevolod Khithov, Alexander Petrov, Igor Tishchenko and Konstantin Yakovlev

Abstract Autonomous fixed-wing UAV landing based on differential GPS is now a mainstream providing reliable and precise landing. But the task still remains challenging when GPS availability is limited like for military UAVs. We discuss a solution of this problem based on computer vision and dot markings along stationary or makeshift runway. We focus our attempts on using infrared beacons along with narrow-band filter as promising way to mark any makeshift runway and utilize particle filtering to fuse both IMU and visual data. We believe that unlike many other vision-based methods, this solution is capable of tracking UAV position up to engines stop. System overview, algorithm description, and its evaluation on synthesized sequence along real recorded trajectory are presented.

Keywords UAV • Fixed-wing • Autonomous landing • Computer vision • Particle filter • Infrared markers • Pattern detection • Real-time • Navigation • Sensor fusing

V. Khithov
Soloviev Rybinsk State Aviation Technical University, Rybinsk, Russia
e-mail: vskhitkov@gmail.com

A. Petrov
NPP SATEK Plus, Rybinsk, Russia
e-mail: petrov@nppsatek.ru

I. Tishchenko (✉)
Program System Institute of Russian Academy of Sciences, Pereslavl-Zalessky, Russia
e-mail: igor.p.tishchenko@gmail.com

K. Yakovlev
Institute for Systems Analysis of Russian Academy of Sciences, Moscow, Russia
e-mail: yakovlev@isa.ru

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_43

1 Introduction

Unmanned aerial vehicles (UAVs) are widely used for surveillance, aerial photography, and reconnaissance; payload carriers and military striking drones gain their popularity. Fixed-wing UAVs are known to have far longer flight times and distances than copter-like ones, but due to necessity of high-speed maintenance during all mission to keep lift their landing is a more challenging task. For fixed-wing UAV rescue it is a common practice to use parachute or catch them with net though disadvantages and limitations of such approach are obvious. Autonomous runway landing is commonly based on differential GPS. However, GPS cannot be very reliable especially for military drones. Anyway, autonomous landing means real-time and precise localization of UAV pose relative to landing site along with keeping trajectory required for safe landing. While latter is a problem of control system, a lot of research is faced to obtaining drone's pose above runway using plenty of sensors including IMU, altimeter, camera, laser sensors, etc. under hard real-time constraints due to high relative speed of UAV. The situation can also be compounded by demand of reliable action under different weather conditions, day and night.

Our method refers to the applications where tracking should be performed until a plane can switch to dead reckoning by sensors on its landing gear, and internal navigation system is capable of initially guiding a plane to the beginning of landing approach with precision of about a hundred meters, knowing its yaw, pitch, and roll with accuracy of several degrees due to IMU and magnetic sensor. We also assume runway as planar.

1.1 Related Works

There exist a lot of methods that use computer vision for runway detection by its edges or corners ([1–3] and many others). Such an approach is only suitable if some sort of asphalt concrete runway is used that has sharp boundaries or lane markings but not suitable for dirt runways.

Template search by correlation and edge-based tracking of landing site template image is utilized in [4], while in [5], mutual information criterion is considered for tracking. Such methods benefit from their versatility, but depend on landing site actual template availability and its parts' uniqueness. In [6, 7], SIFT features are used for matching between photo with known landing site position and video stream. All these methods suffer from scale problem—tracking is lost at too low altitude. In [8], scale problem is solved by using a concentric circle marker as landing site for multirotor UAV—when outer circles are not longer visible, inner ones remain. The idea can hardly be generalized for large fixed-wing drones.

In [9] with the help of single near-infrared beacon, the center of a landing deck is determined, and importance of correct camera exposure managing is shown.

Near-infrared technology has already reached hobby UAVs [10], though the effective range is rather small. In [11], one of the early works, thermal imaging camera is used to detect "hot spots" of aircraft carrier to guide UAV along glide-scope until low-power communication is allowed. Usage of mid-infrared (heated) beacons on a stationary runway is also discussed. Paper does not cover all the aspects of autonomous landing, but usage of thermal range is promising. Thermal imaging is also used by creators of TAMS system [12], but unlike works above, infrared stereo camera and processing unit are placed on the ground and obtained UAV pose is delivered to drone's control system by means of existing communication channel. It is worth nothing that existing commercial landing systems (not using computer vision) like OPATS [13] and UCARS [14] also have "active" ground equipment.

2 System Overview

We discuss autonomous landing method based on visual servoing where infrared beacons placed along runway (their images) are used as features for UAV pose recovery. Light emitters are selected for practical reasons: first, some light emitting is anyway necessary if we want to achieve operation in low light conditions using camera; second, such beacons can be quickly deployed on a temporary runway; third, distance of beacons' visibility is limited primarily not by their size but by intensity that result in (1) easier solving of scale problem on different distances and (2) extremely simple detection algorithm by threshold if we have enough light source power and manageable camera exposure. In practice, however, the method is not so easy, and the challenges are as follows: limited power of beacons, background illumination and simple shape of hotspots provide a lot of clutter to cope with, and the need to distinguish between identically looking beacon images.

System overview is illustrated in Fig. 1. UAV is equipped with frontal-view camera with narrow-band filter, IMU, magnetic sensor and altimeter, all synchronized. It is supposed that in GPS-denied environment UAV's navigation system is still capable of guiding UAV to the vicinity of runway. We use three coordinate frames for world, UAV and camera. Beacons are placed in straight lines with regular distances along both sides of a runway. Straight lines and regular structure are chosen for reason of simple deploying.

Camera position with respect to UAV and its intrinsic is ought to be known.

We look forward using MIR range (3–8 um) for beacon detection and tracking: mid-infrared beacons are just objects heated to several hundred degrees so they can be very cheap while sun is emitting far less in this range than in near infrared. Narrow-band filters for the range also exist. However, thermal sensor technology is far more expensive than NIR, so we tried both. In Fig. 2 (left), appearance of 30 W NIR lamp through dielectric IR filter from 350 m distance on sunny day is shown. In Fig. 2 (right), it is shown that 20-cm object heated by portable gas burner is visible from 700 m distance by thermal camera.

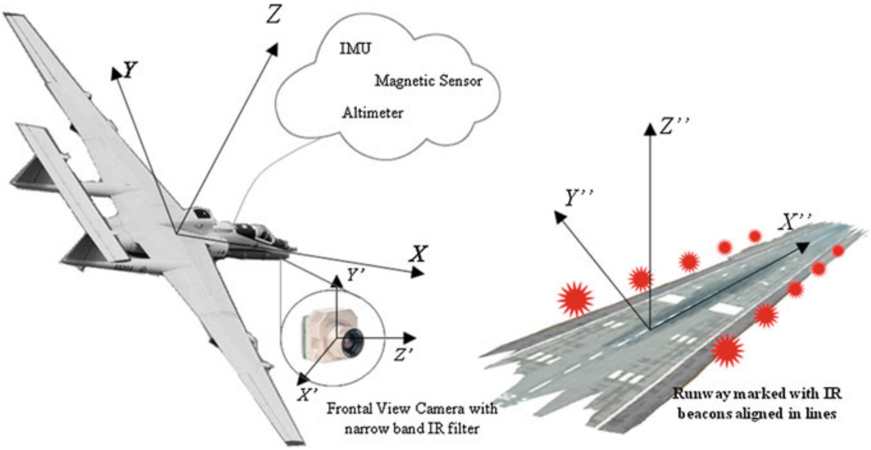

Z
Y
IMU
Magnetic Sensor
Altimeter
Z''
Y''
X''
X
Y'
Z'
X'
Frontal View Camera with narrow band IR filter
Runway marked with IR beacons aligned in lines

Fig. 1 Overview of landing system: UAV equipment, beacons configuration, and coordinate frames

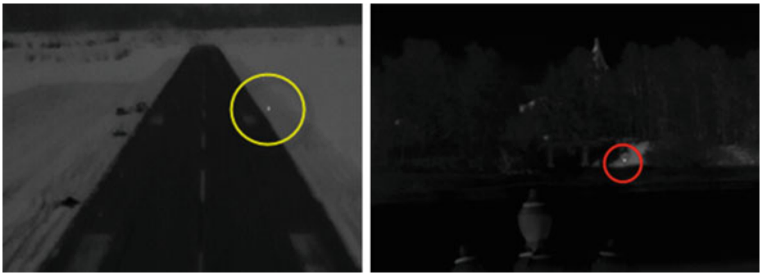


Fig. 2 *Left* 30 W NIR led lamp near runway visible through narrow-band IR filter in sunny day from 350 m; *Right* 20-cm object heated by portable gas burner is visible from 700 m distance on thermal camera image in daylight conditions

3 Algorithm Description

Particle filter is used to track UAV position relative to runway during descent. Let matrix RT_{cam} be camera 3D pose in UAV frame in homogenous coordinates, RT_{uav}—UAV 3D pose in world frame (unknown), K—camera intrinsic matrix, and p_i^0—ith beacon known position on the ground plane. Then we can obtain homography matrix that projects beacons' position p_i from ground plane to the image plane using the following Eq. (1).

$$p_i = H \times p_i^0,$$
$$H = K \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times RT_{cam}^{-1} \times RT_{uav}^{-1} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

So, particle filter is used to model probability of UAV pose to be RT_{uav} in the world frame. Generally, there are six degrees of freedom, but pitch and roll angles are known with high accuracy from IMU synchronized with camera. Final state vector is $(\overline{T}, \gamma, \theta, \varphi, s)$ where $\overline{T}$ is not a degree of freedom but denotes translation vector of UAV in world frame, γ denotes azimuth angles, θ, φ are UAV movement direction of azimuth and pitch, respectively, and s is movement speed.

Initialization. Particle filter is initialized with probability distribution known from UAV's internal navigation system with typical uncertainty of some hundred meters in planar coordinates (GPS-denied action), ten meters in elevation, and several degrees in azimuth.

Prediction. At each prediction step, the difference between previous and current IMU yaw measurement is added with slight Gaussian variance. $\overline{T}$ is changed according to the movement of direction and speed (2)

$$\overline{T}^{(t)} = \overline{T}^{(t-1)} + R(\theta, \varphi) \cdot s \cdot \Delta t. \tag{2}$$

where $R(\theta, \varphi)$ is rotation matrix, and Δt is time increment.

Measurement. Measurement lies in matching projected beacons' positions for each hypothesis with incoming image. Matching consists of three stages:

- extracting light source binary map M;
- applying distance transform D to binary map; and
- each particle's weight calculation based on a equation upon distance map pixels' values $D_{p[i]}$ under projected beacons' positions p_i.

When detecting the light source, one should reasonably try to make as much contrast to background as possible while keeping dynamic range. This can be achieved by selecting appropriate exposure [9]. The law to choose exposure on far distances is to have as many as possible saturated pixels, but do not have connected components of saturated pixels with area more than small predefined value. Moreover, as far as we know about beacons' emission power and camera sensitivity, there is a range of automatic tuning of this parameter depending on how close to landing site UAV is.

The following algorithm is used for light source detection as shown in (Fig. 3), which resembles [15]. We model faint light sources as local maxima of incoming gray-scale image pixel intensities that are above $k \approx 2.5$ standard deviations in this area.

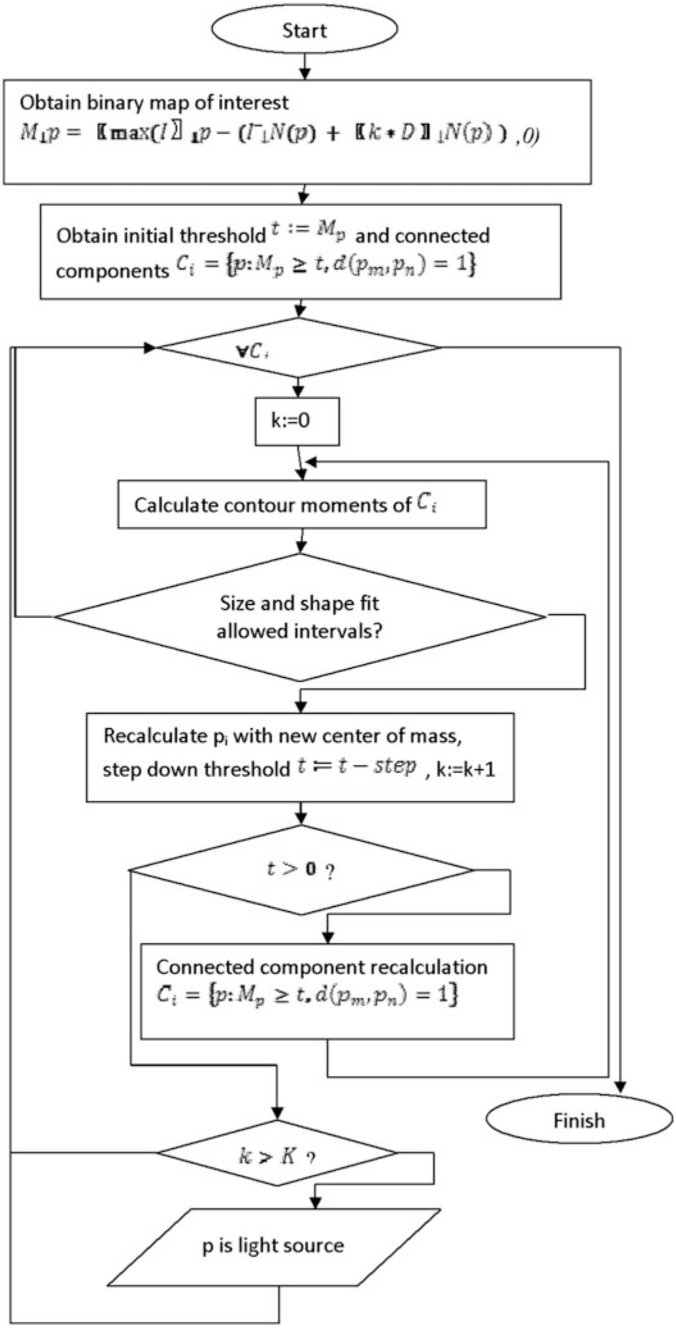

Start
Obtain binary map of interest
$M_{\downarrow}p = 〚\max(I〛_{\downarrow}p - (I\lceil_{\downarrow}N(p) + 〚k * D〛_{\downarrow}N(p))\ ,0)$
Obtain initial threshold $t := M_p$ and connected components $C_i = \{p: M_p \geq t, d(p_m, p_n) = 1\}$
$\forall C_i$
k:=0
Calculate contour moments of C_i
Size and shape fit allowed intervals?
Recalculate p$_i$ with new center of mass, step down threshold $t \coloneqq t - step$, k:=k+1
$t > 0$?
Connected component recalculation $C_i = \{p: M_p \geq t, d(p_m, p_n) = 1\}$
Finish
$k > K$?
p is light source

Fig. 3 Algorithm of light source detection. I$_p$—source image, $I_{m \in N}$—vicinity N of image, $D(I_{m \in N})$—standard deviation of pixel intensities in vicinity

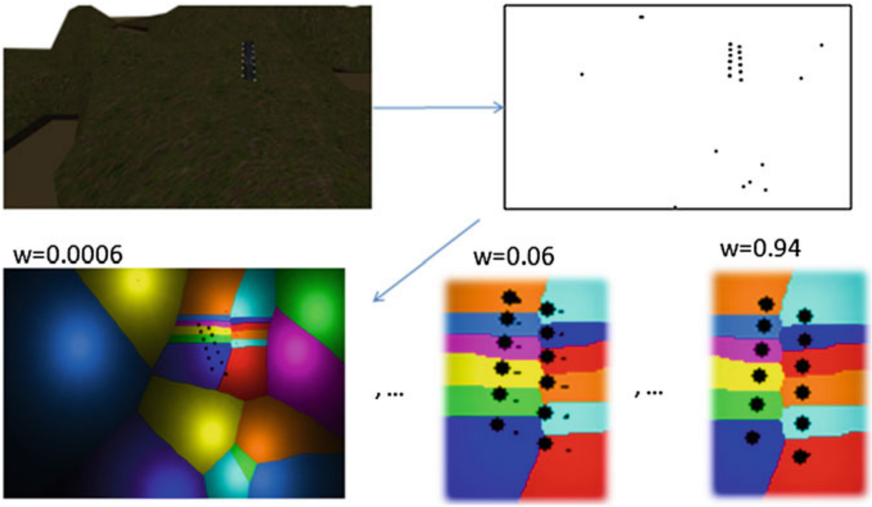


Fig. 4 Beacons and some outliers are detected on synthesized image, and distance transform is applied and multiple hypotheses are matched. *Black circles*: projections of three sets of beacons (each for one hypothesis) to distance transform map (labeling is not used). *Black dots* show detected beacons' positions (on last image are hidden below projected circles due to precise match). Matching weight for the cases is denoted by *w*

We assign weight to a hypothesis with the following Eq. (3):

$$w = \exp\left(\sum_{i=1}^{N} \frac{D_{p[i]}}{(D_{p[i]} + P)^{q}}\right), \tag{3}$$

where q, P are parameters.

The pipeline is illustrated in Fig. 4.

4 Evaluation

At the moment of paper writing, we have recorded real glide trajectory of UAV and its IMU signals during landing. We model a video input sequence by blender (Fig. 5). The goal of evaluation was to compare computed trajectory with real glide trajectory of UAV based on simulated image input and recorded IMU signals.

We use 16 markers and calibrated camera model. Our implementation works online on a personal computer with the x86 processor Pentium (R) Dual-Core CPU with 2.00 GHz. The comparable results are shown in Fig. 6 and detailed in Table 1.

The results presented in Table 1 show the accuracy of computed trajectory by our algorithm implementation across the distance from landing point (like angular variation).

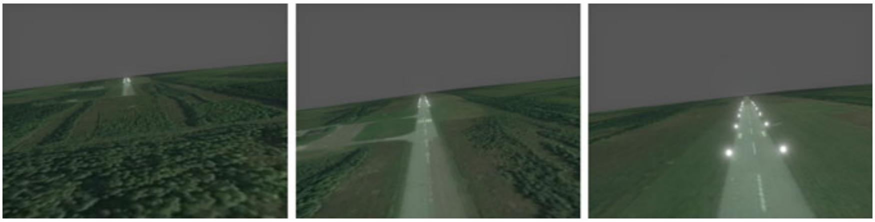


Fig. 5 Images from simulated video input

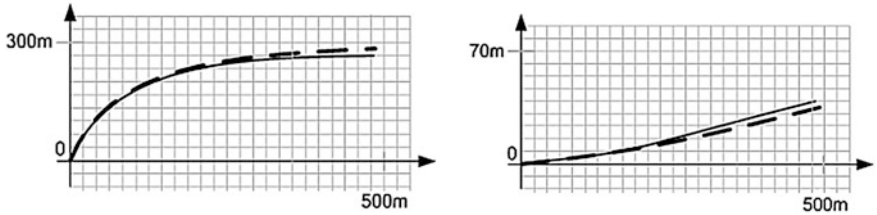


Fig. 6 Computed (*dashed*) and recorded glide trajectory. *Top view* on trajectory (*left* picture), *side view* on trajectory (*right* image)

Table 1 Compare recorded and calculated glide trajectory

Parameter	Value	Units
Maximum linear deviation across the distance from landing point	4, 12 on 500; 1, 52 on 100; 0, 15 on 10	Meters
Maximum orientation deviation across the distance from landing point	5, 3 on 500; 1, 5 on 100; 0, 1 on 10	Angles

5 Conclusion

Autonomous fixed-wing UAV landing task in GPS-denied environment is one of the actual tasks in UAV navigation problem area. We proposed a method of UAV navigation on landing stage based on computer vision, IMU sensor data fusion, and infrared dot markers. The described algorithm was evaluated in simulated environment and the comparable results are presented. Infrared dot markers are rather convenient for use on practice.

As a future directions of project development, we plan to evaluate our system on real data (like a left image in Fig. 2, but with MIR markers) and optimize some part of algorithm for onboard using. Also, we plan to investigate other markers for autonomous landing, e.g., flashing markers or laser beam markers.

Acknowledgments This work was supported by the Ministry of Education and Science of the Russian Federation (RFMEFI60714X0088) agreement for a grant on "Development of methods and means of processing and intelligent image analysis and flow of data obtained from a set of stationary and mobile sensors, using high-performance distributed computing for the tasks of monitoring the indoor placement and adjacent outdoor territories."

References

1. Williams, P., Crump, M.: Intelligent landing system for landing UAVS at unsurveyed airfields. In: 28th Congress of the International Council of the Aeronauti-cal Sciences, 23–28 Sept 2012, Brisbane, Australia Paper ICAS 2012-11.6.2
2. Laiacker, M., Kondak, K., Schwarzbach, M., Muskardin, T.: Vision aided automatic landing system for fixed wing UAV. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2971–2976, Nov
3. Zhuang, L., Han, Y., Fan, Y., Cao, Y., Wang, B., Zhang, Q.: Method of pose estimation for UAV landing. Chinese Optics Letters, vol. 10, (2012). doi:10.3788/col201210.s20401
4. Coutard, L., Chaumette, F.: Visual detection and 3D model-based tracking for landing on an aircraft carrier. In: 2011 IEEE International Conference on Robotics and Automation (ICRA). IEEE (2011)
5. Dame, A., Marchand, E.: Accurate real-time tracking using mutual information. In: IEEE International Symposium on Mixed and Augmented Reality, ISMAR'10, pp. 47–56, Seoul, Korea, Oct (2010)
6. Miller, A., Shah, M., Harper, D.: Landing a UAV on a runway using image registration. In: IEEE International Conference on Robotics and automation Pasadena, CS, USA (2008)
7. Zhang, X., Liu, X., Yu, Q.: Landing site locating of UAV by SIFT matching. Published in SPIE Proceedings, vol. 6625
8. Lange, S., Sunderhauf, N., Prozel, P.: Autonomous landing for a multirotor UAV using vision. In: SIMPAR 2008 International, pp. 482–491
9. Garratt, M., pota, H., Lambert, A., Eckersley-Maslin, S.: Systems for automated launch and recovery of an unmanned aerial vehicle from ships at sea. In: 22nd International UAV Systems Conference, Bristol, April (2007)
10. IR-LOCK Sensor for Precision Landing [online]. http://diydrones.com/profiles/blogs/ir-lock-sensor-fo
11. Yakimenko, O.A., Kaminer, I.I., Lentz, W.J., Ghyzel, P.A.: Unmanned aircraft navi-gation for shipboard landing using infrared vision. IEEE Trans. Aerosp. Electron. Syst. **38**, 1181–1200 (2002)
12. Kong, W., Zhang, D., Wang, X., Xian, Z., Zhang, J.: Autonomous landing of an UAV with a ground-based actuated infrared stereo vision system. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). doi:10.1109/IROS.2013.6696776
13. RUAG, "OPATS." [online]. https://www.ruag.com
14. Sierra Nevada Corporation. Automatic Recovery System. [online] http://www.sncorp.com/prod/atc/uav/default.shtml
15. Bertin, E. & Arnouts, S.: SExtractor: Software for source extraction. Astronomy and Astrophysics Supplement, vol. 117, pp. 393–404 (1996)

Local Obstacle Avoidance Using Obstacle-Dependent Gaussian Potential Field for Robot Soccer

Dong-Ok Kim, Da-Yeon Lee, Jae-Il Oh, Tae-Hoon Kang and Tae-Koo Kang

Abstract We address the vision-based local obstacle avoidance method. The proposed method is based on the obstacle-dependent Gaussian potential field (ODG-PF). Basically, the potential-field (PF) algorithm is a global path planning algorithm. Therefore, we modify the conventional PF to apply the local path planning condition which does not need any environment information. The proposed method makes the robot to avoid the obstacles according to the decision of the optimal path by itself. In particular, we can apply it to the robot soccer system to avoid the collision with the opposite player robots.

1 Introduction

A robot soccer is a good example application for multiple robot system in the real world. There are three control schemes for soccer robot control. The first scheme is a remote-brainless soccer robot system in which a host computer controls the robots by commanding robot's velocities like a radio-controlled car. The second scheme is a vision-based one which controls or manipulates the robots by processing information from the vision with a supervisor processor and sends commands to robots directly. So it is rather like a traditional supervisory system. The other scheme is a robot-based one. Wherein each autonomous robot can make a decision based on the information, it collects with its sensors and, if needed, can send information to others. Robots which can operate autonomously are designed for the robot-based one [1].

D.-O. Kim (✉) · D.-Y. Lee · J.-I. Oh · T.-H. Kang · T.-K. Kang
Department of Information and Telecommunication Engineering, Sangmyung University, Cheonan-si, Chungcheongnam-do 330-720, Republic of Korea
e-mail: ehddhr92@naver.com
URL: http://ittech.smuc.ac.kr

T.-K. Kang
e-mail: tkkang@smu.ac.kr

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_44

In this paper, we propose the vision-based local obstacle avoidance method. The proposed method makes the robot avoid the obstacles according to the decision of the optimal path by itself.

Generally, path planning method can be divided into the two parts: (1) global path planning and (2) local path planning. In the case of the global path planning, we need a condition that the path planner has to know the environment information including the location, position, and size of the obstacle in advance. On the contrary, the local path planning does not need the advance information regarding the obstacle avoidance, which has an advantage for obstacle avoidance under outdoor environment.

The proposed method is based on the obstacle-dependent Gaussian potential field (ODG-PF). Basically, the potential-field (PF) algorithm is a global path planning algorithm. Therefore, we modify the conventional PF to apply the local path planning condition which does not need any environment information. As the application of the proposed method, we can apply it to the robot soccer system to prevent from two robots colliding with each other.

In Sect. 2, we review the conventional PF algorithm and in the Sect. 3, and we introduce the ODF-PF algorithm and evaluate the proposed method using the robot.

2 Potential-Field Review

The idea of a potential field is taken from nature. For instance a charged particle navigating a magnetic field, or a small ball rolling in a hill, the idea is that depending on the strength of the field, or the slope of the hill, the particle, or the ball can arrive to the source of the field, the magnet, or the valley in this example [2].

In robotics, we can simulate the similar effect, by creating an artificial potential field that will attract the robot to the goal. By designing adequate potential field, we can make the robot exhibit simple behaviors. For instance, let us assume that there is no obstacle in the environment and that the robot should seek this goal. To do that in conventional planning, one should calculate the relative position of the robot to the goal, and then apply the suitable forces that will drive the robot to the goal [2].

In the potential-field approach, the potential field is defined across the entire free space, and in each time step, we calculate the potential filed at the robot position and then calculate the induced force by this field. The robot then should move according to this force. We can also define another behavior that allows the robot to avoid obstacles.

We make each obstacle generate a repulsive field around it. If the robot approaches the obstacle, a repulsive force will act on it, pushing it away from the obstacle. The two behaviors, seeking and avoiding, can be combined by combining

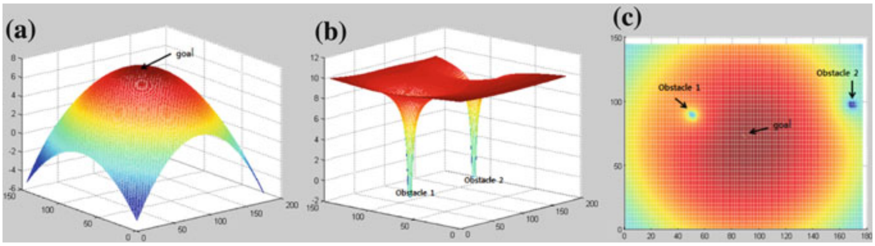


Fig. 1 Potential field: **a** attractive PF, **b** repulsive PF, and **c** total PF

the two potential fields, the robot then can follow the force induced by the new filed to reach the goal while avoiding the obstacle [2]. Figure 1 shows the sample images for the attractive PF, repulsive PF, and the total PF.

In mathematical expression, the overall PF is as follows:

$$U(\mathbf{q}) = U_{\text{goal}}(\mathbf{q}) + \sum U_{\text{obstacles}}(\mathbf{q}) \tag{1}$$

and the induced force is

$$F = -U(\mathbf{q}) = (U/x, U/y) \tag{2}$$

Robot motion can be derived by taking small steps driven the local force [3]. Typically, U_{goal} is defined as a parabolic attractor [3]:

$$U_{\text{goal}} = d(\mathbf{q}, \text{goal})^2 \tag{3}$$

where d is the Euclidean distance between the state $\mathbf{q}$ and the goal.

In addition, $U_{\text{obstacles}}$ is modeled as a potential barrier when the robot approaches the obstacles [3]:

$$U_{\text{obstacles}} = d(\mathbf{q}, \text{obstacle})^{-1} \tag{4}$$

where d is the Euclidean distance between the robot in state q and the closest point on the obstacle. The repulsive force is computed with respect to either the nearest obstacle or summed over all the obstacles in the environment.

Potential field method was developed as an online collision avoidance approach, applicable when the robot does not have a prior model of the obstacle, but senses then during motion execution [4]. Moreover, it is obvious that its reliance on local information can trap it in a local minimum as shown in Fig. 2.

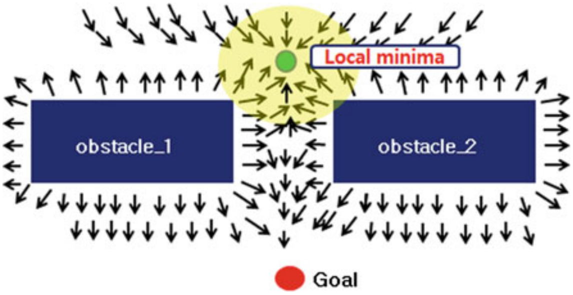


Fig. 2 Local minimum problem of the conventional PF

3 Obstacle-Dependent Gaussian Potential Field (ODG-PF)

In this section, we propose the obstacle-dependent Gaussian potential field (ODG-PF) to generate the optimal local path in each robot system. The ODG-PF can be divided into two terms: (1) attractive PF and (2) repulsive PF, which is similar structure with the conventional PF algorithm. Detailed method we proposed is described as follows.

3.1 Attractive Potential Field

The attractive PF is the virtual force guiding the robot to the target location. Because we cannot know the global position in the local robot systems, we have to make the initial direction which is generally set to the moving direction of the robot. From the initial position, the robot can return the expected global position after escaping the obstacle using the attractive PF. Figure 3 shows the relation between initial and current positions.

Based on Fig. 3, we can calculate the displacement with respect to the moving direction as follows:

$$\Delta\theta_i = |\theta_{\text{init}} - \theta_i| \tag{5}$$

From (5), we can obtain the final attractive PF as shown in (6).

$$U_{\text{att}}(\Delta\theta_i) = \gamma\,\Delta\theta_i, \quad \gamma > 0, \tag{6}$$

where γ is a weight variable.

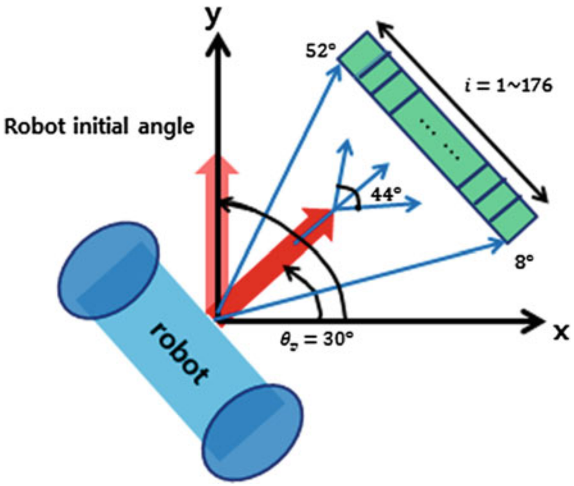


Fig. 3 Relation between the initial and current positions of the robot

3.2 *Repulsive Potential Field*

To calculate the repulsive PF, we define the width and height of an obstacle as ω_x and ω_y, respectively. Then, we can model a potential field to include the obstacle with Gaussian function as shown in Fig. 4. As shown in Fig. 4, we can regard the width and height as the standard deviation of the Gaussian model including the obstacle.

$$\sigma_x = \omega_x, \quad \sigma_y = \omega_y \tag{7}$$

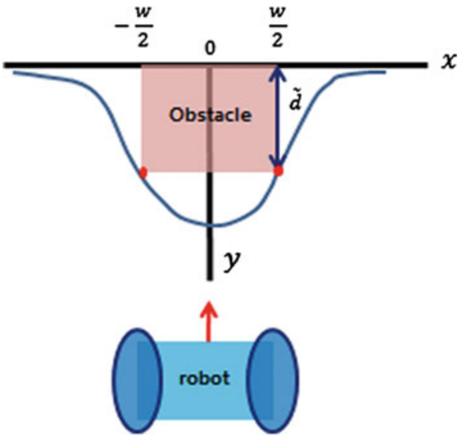


Fig. 4 Local potential-field model including the obstacle

In Fig. 4, we define a distance between an obstacle and the robot as d and a difference between the maximum range that the robot can detect and d as $\tilde{d}$.

Using the Gaussian model, we can define the repulsive PF as follows:

$$\tilde{d} = A * \exp\left(-\left(\frac{\left(\frac{\omega_x}{2}\right)^2}{2\omega_x^2} + \frac{\left(\frac{\omega_y}{2}\right)^2}{2\omega_y^2}\right)\right) \tag{8}$$

where A is a scale factor according to the distance between the robot and the obstacle.

From (8), we derive the equation as follows.

$$\tilde{d} = A * \exp\left(-\left(\frac{\left(\frac{\omega_x^2\omega_y^2}{2}\right) + \left(\frac{\omega_x^2\omega_y^2}{2}\right)}{4\omega_x^2\omega_y^2}\right)\right) = A * \exp(-\frac{1}{4}) \tag{9}$$

Based on (9), we can gain the scale factor A as shown in (10).

$$A = \tilde{d} * \exp(-\frac{1}{4}) \tag{10}$$

Finally, we define the repulsive PF as follows.

$$U_{rep} = \tilde{d} * \exp\left(\frac{1}{4} - \left(\frac{(x - x_0)^2}{2\omega_x^2} + \frac{(y - y_0)^2}{2\omega_y^2}\right)\right) \tag{11}$$

where (x_0, y_0) and (x, y) mean a center of the mass of the obstacle and a pixel position in the vision mounted on the robot system.

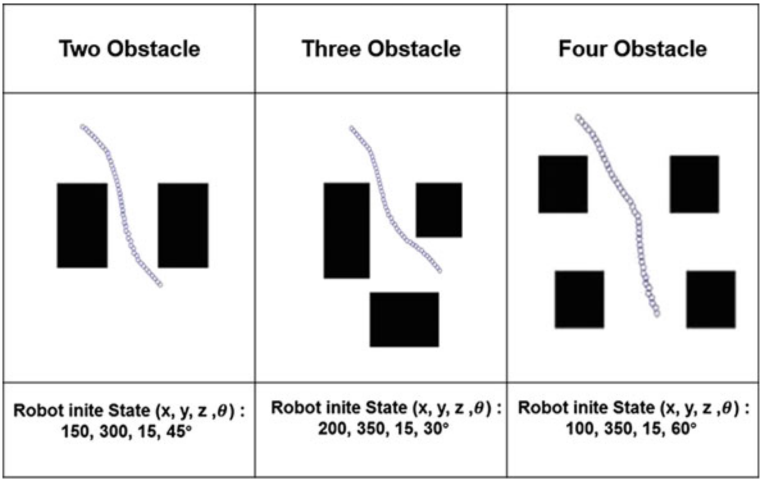
From (6) and (11), the ODG-PF U can be defined as (11).

$$U = \gamma\Delta\theta_i + \tilde{d} * \exp\left(\frac{1}{4} - \left(\frac{(x - x_0)^2}{2\omega_x^2} + \frac{(y - y_0)^2}{2\omega_y^2}\right)\right) \tag{12}$$

4 Experimental Results

4.1 Simulation Results

Evaluate the performance for the obstacle avoidance under a simulation environment. Figures 5 and 6 show the examples of the experimental environment for the multiple obstacles and the simulation result under three obstacles condition.



Two Obstacle	Three Obstacle	Four Obstacle
Robot inite State (x, y, z ,θ) : 150, 300, 15, 45°	Robot inite State (x, y, z ,θ) : 200, 350, 15, 30°	Robot inite State (x, y, z ,θ) : 100, 350, 15, 60°

Fig. 5 Local potential-field model including the obstacle

As shown in Figs. 5 and 6, the ODG-PF generates automatically the optimal path without occlusion to the obstacles. In Fig. 6, the green boxes present the detected obstacles and blue dots mean the target points for the moving direction from the viewpoint of the robot.

In addition, the graph presented in blue illustrates the status of the ODG-PF corresponding to the obstacles, and the first and the second images of the second column in each image show the attractive PF and repulsive PF in gray color, respectively. The second image of the first column illustrates the total ODG-PF in each image. As a result, the robot determines the moving direction to the brightest part corresponding to the lowest level of the ODG-PF.

The selected path from the ODG-PF provides the efficiency in terms of total length of the path and safety with respect to the obstacles.

4.2 *Experimental Result Using the Robot System*

To confirm the simulation results, we conduct the experiments using a real mobile robot. Figure 7 shows the experimental environment and a testing robot system.

To investigate the avoidance ability of the ODG-PF, we use the static obstacle like a wooden box and dynamic obstacle like a moving person. Figures 8 and 9 illustrate the results of obstacle avoidance under static and dynamic obstacle conditions. As shown in Fig. 8, the robot system moves by finding the optimal path

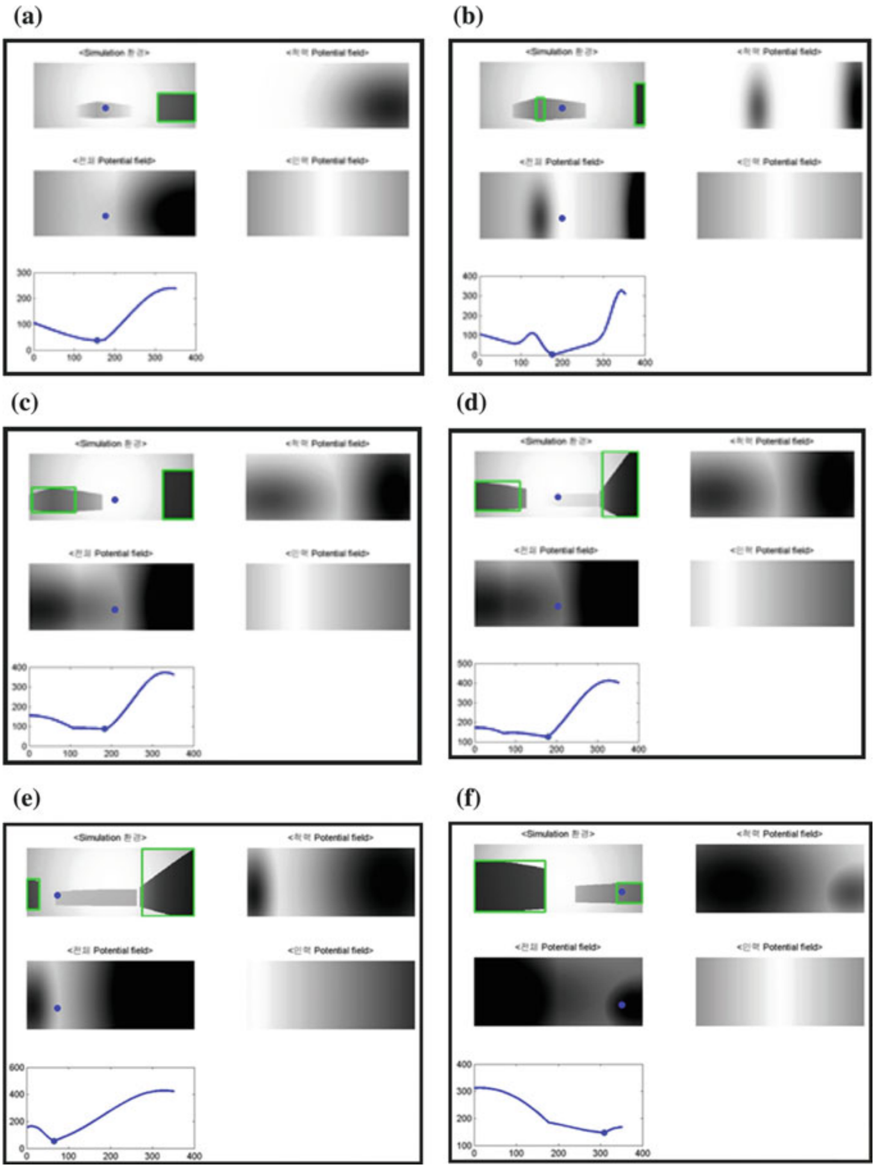


Fig. 6 Simulation result for the obstacle avoidance using ODG-PF

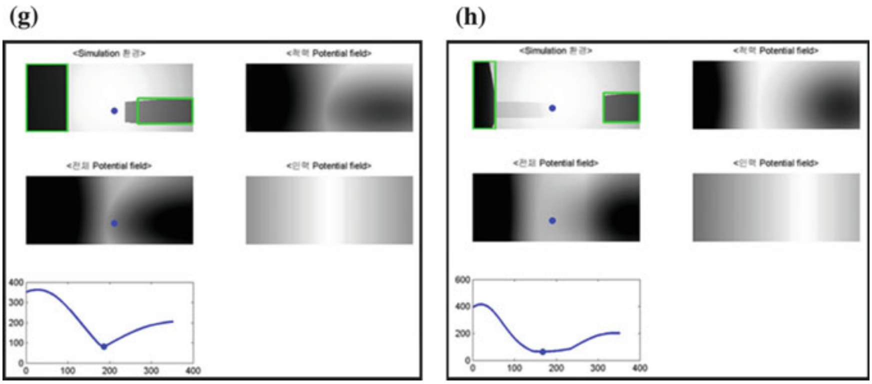


Fig. 6 (continued)

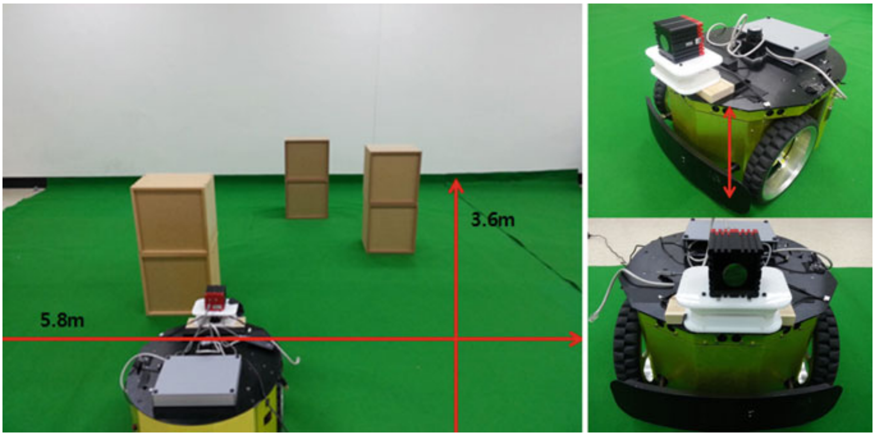


Fig. 7 Simulation result for the obstacle avoidance using ODG-PF

without collision against the obstacles. Especially, in the case of dynamic obstacle conditions, the robot system can modify the moving path in real time during the obstacle avoidance under static obstacle condition.

Using the ODG-PF, we can apply it to the robot soccer to move the ball without collision with the opposite robot player. Moreover, in the free-kick for penalty kick situation, the robot can find the trajectory of the ball to the goalpost using the ODG-PF.

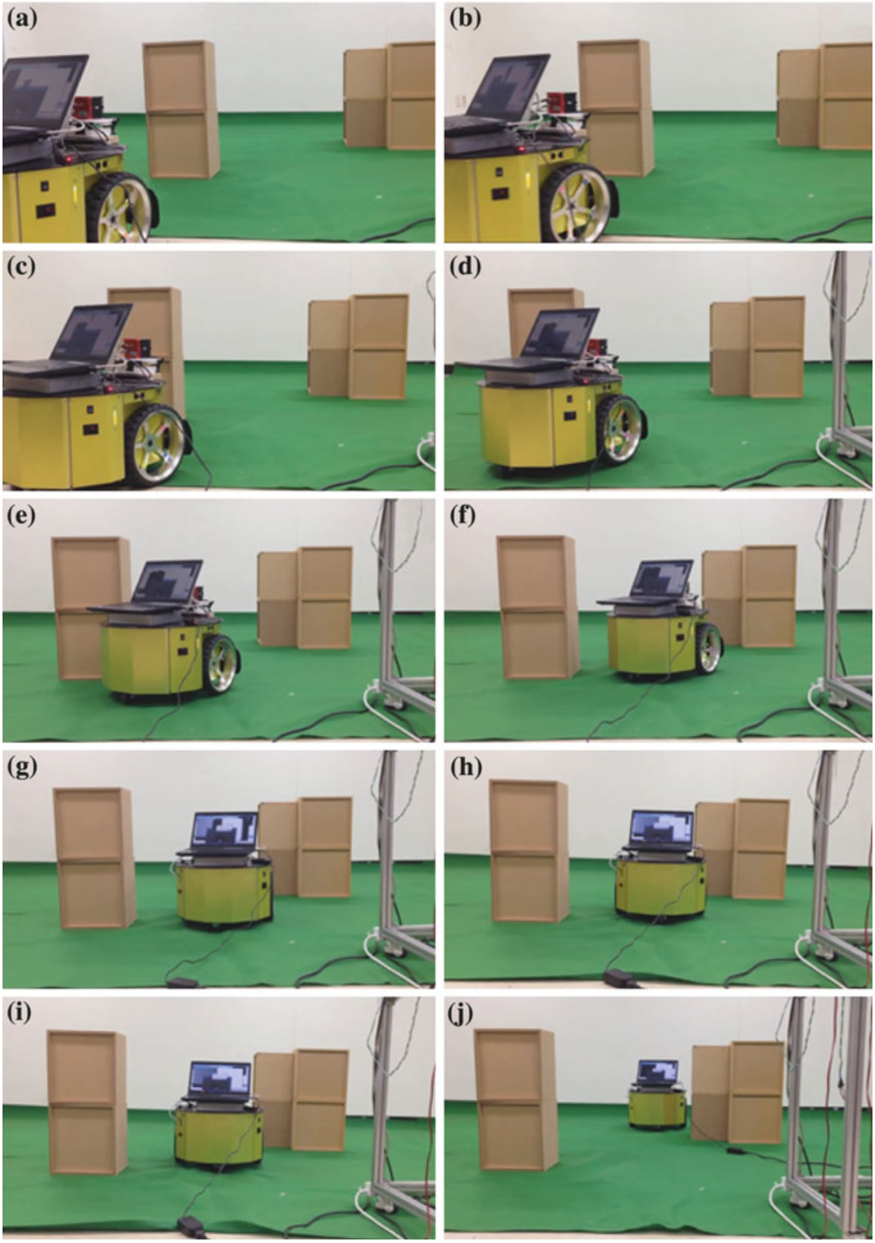


Fig. 8 Experimental result for the static obstacle avoidance using ODG-PF

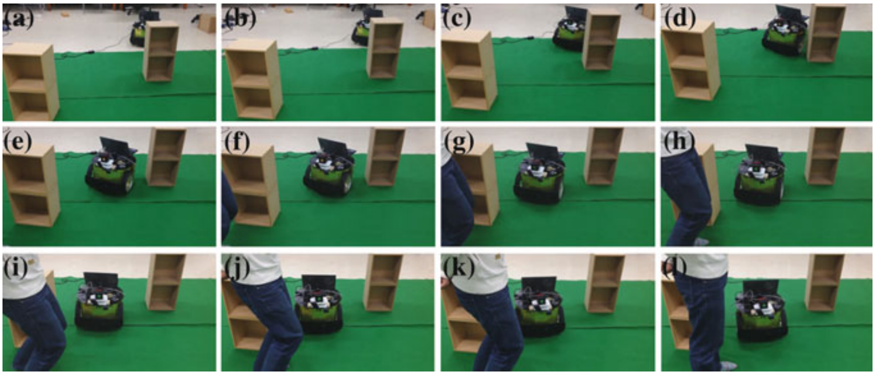


Fig. 9 Experimental result for the dynamic obstacle avoidance using ODG-PF

5 Conclusion

We address the vision-based local obstacle avoidance method. The proposed method is based on the obstacle-dependent Gaussian potential field (ODG-PF). Basically, the potential-field (PF) algorithm is a global path planning algorithm. Therefore, we modified the conventional PF to apply the local path planning condition which does not need any environment information. The proposed method makes the robot avoid the obstacles according to the decision of the optimal path by itself. In particular, we can apply it to the robot soccer system to avoid the collision with the opposite player robots.

References

1. Kim, J.-H., Shim, H.-S., Kim, H.-S., Jung, M.-J., Choi, I.-H., Kim, J.-O.: A cooperative multi-agent system and its real time application to robot soccer. Int. Conf. Robot. Autom. **1**, 638–643 (1997)
2. Goodrich, M.A.: Potential fields tutorial. http://borg.cc.gatech.edu/ipr/files/goodrich_potential_fields.pdf
3. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 500–505 (1985)
4. Fu-guang, D., Peng, J., Xin-qian, B., Hong-jian, W.: AUV local path planning based on virtual potential field. In: 2005 IEEE International Conference on Mechatronics and Automation, vol. 4, pp. 1711–1716 (2005)

Methods of Visual Navigation of the UAV Flying Over the Nonplanar District

Dmitry Stepanov and Igor Tishchenko

Abstract This article is devoted to development of methods of positioning of unmanned aerial vehicles (UAV) with use of video cameras and methods of computer vision. UAV flies over the district, having a relief. Main feature of proposed method is using of Gabor's wavelets to search of so-called reference points with known 3D-coordinates. Reference points are used from etalon images of the district (for example, satellite images) and looked for on the frame from the UAV's camera. Also, comparison of feature points from the frame and a satellite image detected by SURF algorithm is used. The system of modeling of flight of the virtual UAV which allows testing the developed algorithms is realized.

Keywords Visual navigation · Gabor's wavelets · UAV · Reference points · SURF · Feature points · Satellite images

1 Introduction

Scope of application of unmanned aerial vehicles (UAV) promptly extends. One of the main components of any aircraft is navigation system. Now, the following types of navigation equipment are applied:

- *GPS sensor*. Despite its compactness and simplicity of use, space-based navigation is unreliable in some cases (urban canyons, weather conditions, artificial hindrances, etc.).
- *Pressure altimeter*. The principle of action is based on decreasing of atmospheric pressure at increase in height of flight. The error depends on height of

D. Stepanov · I. Tishchenko (✉)
Program System Institute of Russian Academy of Sciences, Pereslavl-Zalessky,
Russian Federation
e-mail: igor.p.tishchenko@gmail.com

D. Stepanov
e-mail: mitek1989@mail.ru

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_45

flight and air temperature. Pressure altimeter has big delay at fast maneuvers in the vertical plane.

- *Radio altimeter, ultrasonic range finder, laser scanner, laser range finder*. All these devices are active navigation equipment and, therefore, their activity can be detected by instruments of radio tracking devices (that is inadmissible in case of use of the UAV in secret missions). There are also other drawbacks. For example, laser scanners give out unusable results in environments with glass objects. Also, for measurement of distances to remote objects is required to enough powerful laser, but it can be unsafe for human's eyes. Some scanners can be confused by sunlight, and are often heavy and expensive. Despite some drawbacks, such type of navigation equipment is widely used by many researches, with the use of additional tools or without it [1–6].
- *Inertial measurement unit (IMU)*. It consists of a three-axial gyroscope (for estimate of position) and the three-axial accelerometer (for estimate of orientation). The principle of action is based on double integration of values of acceleration. Such devices aren't used independently, as the error in calculations is accumulated in process of time. Also, IMU needs the periodic adjustment [1, 4, 7].

Thus, all the popular methods of UAV navigation have drawbacks. For decades, the problem of UAV navigation with the use of methods and algorithms of computer vision is studied.

1.1 Methods of Visual Navigation

Many researchers use stereo cameras for navigation flying and ground unmanned vehicles [3–5, 8]. But to calculate the three-dimensional coordinates of objects observed on the left and the right frame, two cameras should be placed at a sufficient distance from each other, this distance depends on altitude and distance to the observed object (if frontal view is used). If a UAV flies at high altitude, a camera with a long focal length should be used, but such cameras often have large size and large mass. Thus, the use of stereo cameras is possible only when UAV flies at low altitudes (for example, an autonomous landing) or indoors.

If single camera is used, that can estimate relative motion of UAV, by using the feature correspondences of consecutive image frames (or between frame and etalon image), but only up to an arbitrary scale factor. The fundamental matrix F describing the motion of the camera can be computed. Decomposing this matrix yields the relative rotation (matrix R) and translation motion of the camera (vector T), but we can only compute the translational direction of the motion, and not its magnitude. To resolve this scale ambiguity, scene knowledge is necessary.

For example, in paper [9] lines in the camera images are fitted to the edges of a 3D model of an office environment with known structure. Johnson [10] detected lines in a hallway, and used the assumption of a straight hallway to infer the vehicle pose. But these methods are not used in semistructured environments (forests and fields), their applicability is constrained to environments with specific features (cities, factories, roads).

Other approach is correlation-based matching: the sensed image is placed at every pixel location in the reference image, then, a similarity criteria is adopted to decide which location gives the best fit. Methods based on correlation can be implemented very efficiently and are suitable for real-time applications [11]. They can be applied also in areas with no distinct landmarks. However they are typically sensitive to geometric (scale, orientation) and luminance differences between the sensed and reference image [6].

The most common method in visual navigation is to compare images from UAV's camera and geolocated reference images of Earth's surface. Also the comparison of two or more consecutive frames of video to calculate the relative change in the position and orientation of the UAV between frames is used. If the observed scene (the area over which the UAV flies) is flat, or the height difference can be neglected, two images of the area will be connected by the homography matrix H, which can be decomposed into R and T, without any additional knowledge about scene [6, 12, 13]. But if the area has a terrain, such method is not applicable.

Thus, the problem of visual navigation of UAV during the flight over a known terrain with a relief at enough height has been little studied. The aim of our research is to develop methods and algorithms for visual navigation of UAV, which are able to work effectively in the presence of height differences on the underlying surface. At the current stage of research, we use 3D-modeling system to create a series of images that mimic the video on board the real UAV, because active experiments with real UAV (especially long flights over the extended territories) is not always possible. Also, GPS and IMU can't estimate the position and orientation of the UAV with absolute accuracy, so we can't give a reliable estimate of the accuracy of proposed UAVs' positioning algorithms using video.

2 Key Features of Our Approach to the Visual Navigation

The main idea of our method the visual navigation is the search of so-called reference point on UAV's frames. Reference points are taken from a geolocated satellite images. 3D-coordinates of points in the geocentric (Cartesian) coordinate system tied to the Earth's center are known. This reduces the problem of estimation the position and orientation of the UAV to the problem of the external camera calibration: calculating its position and orientation relative to the global coordinate system [14]. At the same time, we assume that the camera on the UAV is rigidly fixed and looking at the Earth's surface perpendicular to the direction of flight.

We can get the geocentric coordinates of reference points, if we know its geodetic coordinates (longitude, latitude, altitude). In reality, the spatial coordinates of points of the earth's surface can be calculated during the geodetic measurements, aerial or satellite imagery. In our research, we calculate 3D-coordinates of the reference points using elevation map.

To find reference points in UAV' frames, we use the capabilities of Gabor wavelets [15], which is used in many tasks of computer vision. In review [16], a successful experience of applying of Gabor filters in the problem of stereo correspondence is described. We use a similar approach: for each reference point we construct a descriptor by convolution the vicinity of the reference point with each of the 40 wavelets, which are defined by the following equations (parameters u and v define the form of each wavelet):

$$\begin{gathered} G(x, y, v, u) = \frac{k_v^2}{4\pi^2} \cdot e^{\left(-\frac{k_v^2}{4\pi^2} \cdot \frac{(x^2+y^2)}{2}\right)} \cdot \left(\cos(k_v \cdot (x\cos(\varphi_u) + y\sin(\varphi_u))) - e^{-2\pi^2}\right) \\ k_v = \frac{\pi}{2 \cdot \left(\sqrt{2}\right)^v}, \quad v = \{0, 1, 2, 3, 4\} \\ \varphi_u = u \cdot \frac{\pi}{8}, \quad u = \{0, 1, 2, 3, 4, 5, 6, 7\} \end{gathered} \tag{1}$$

Each descriptor (*Gabor jet*) is a 40-dimensional vector. Jet constructed for a reference point on the reference image is compared with the jets from the frame of video. We use cosine of the angle between the vectors as a measure of proximity between the jets:

$$S_1\left(\vec{A}, \vec{B}\right) = \frac{\sum_{i=1}^{40} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{40} A_i^2} \cdot \sqrt{\sum_{i=1}^{40} B_i^2}} \tag{2}$$

Thus, we find the coordinates of a reference point on UAV's frame. To solve the problem of external camera calibration, coordinates of at least three reference points are needed; using more points increases the accuracy of estimation the position and orientation in the presence of noise and minor errors in determining the coordinates of reference points.

Also, we use the assumption that the maximum speed of the UAV is limited and the orientation (R_{prev}) and position (T_{prev}) of UAV for the previous frame are known. This allows to calculate the approximate position of the reference point on the current frame and to reduce the scope of the search of reference points. But if the UAV is maneuvering, its orientation substantially change between the previous and the current frame (e.g., when the yaw can significantly change), so using the previous estimated values of orientation angles will lead to the fact that the predicted position of the reference point on the current frame will be at a considerable distance from its true position. Another problem lies in the fact that Gabor wavelets have invariance to geometric image transformations (scaling and rotation), but within narrow limits [16]. For example, if the yaw differs from the north direction more than 10°, attempts to find images of the reference points using Gabor wavelets failed.

To solve the above problems, we compute the fundamental matrix relating the current frame from UAV and satellite image. Next, we obtain a matrix R_{rel} of relative rotation between current frame and satellite image and transform this matrix into three angle of rotation: φ, ψ, θ (pitch, roll and yaw). We use θ for constructing a new rotation matrix $R_{\mathrm{curr\_app}}$, which allows to calculate the approximate position of a reference point on the current frame and for constructing matrix R_{rect}, which is used for geometrical correction of the current frame. After correction, precise localization of reference points using Gabor wavelets becomes possible.

Next, we describe the components system for UAV flight simulation and autonomous navigation. A detailed description of each component is the subject of a separate publication, so we limit ourselves to the general description.

3 Preparing of Initial Data for the Solution of Visual Navigation Task

Initially, we construct so-called pyramid for satellite images, each layer of the pyramid is divided into tiles using *gdal2tiles* utility, which is part of the *GDAL* library [17]. Satellite image that was used in the experiments has a resolution 10000 × 10000 pixels, spatial resolution—1 m per pixel. Availability of satellite image in several scales facilitates the matching points between the UAV frame and the reference image. Splitting into tiles eliminates the need to load data of the whole reference image into memory. To estimate the position and orientation, we use only tiles that fall within the field of view of the UAV's camera at the previous time. A suitable layer of the pyramid is chosen in view of the UAV's flight altitude at the previous time, the spatial resolution of each layer and the focal length of the UAV's camera.

We detect so-called feature points on each tile and calculate their descriptors using a SURF algorithm. Among the feature points found on the tile, we select a few points with the highest values of hessian. In the SURF, the value of hessian is used as a measure of "specialness" of feature points [18]. These few points will be used as reference points. For each of them we compute another descriptor consisting of various responses of the Gabor wavelets. Also we compute three-dimensional coordinates of reference point. Experiments have shown that to solve the problem of positioning only two reference points on each tile are needed.

Data associated with all features and reference points are saved on the hard drive. The procedure described in this section, are performed once.

4 Flight Simulation Through the Generation of the Virtual Video Sequence

User sets the desired trajectory of UAV, marking control points route on a small-size copy of the satellite image, control points are then connected by straight lines. A series of intermediate points are selected on each flight segment, these points correspond to the moments of time when the camera on the UAV took off the next frame of video. For simplicity, we assume that the UAV is moving at a constant speed. For each intermediate point we calculate position and orientation of UAV's camera, and generate a frame of using the terrain engine *osgEarth* [19]. In experiments, we generated frames with a resolution of 1024 × 768 pixels. Further, the frames are stored in the video file. The position and orientation of the UAV's camera, which are associated with each frame, are also stored.

5 Estimation of the Position and Orientation of the UAV's Camera Using Video Sequence

The task is to estimate the position and orientation of the camera in a global coordinate system using UAV's frame. Real values of the position and orientation are available, and this information is used to calculate the errors in estimation of coordinates of and navigation angles (pitch, roll and yaw).

Initially, we detect the feature points on UAV's frame and calculate their descriptors using SURF. Then we select the appropriate layer of the pyramid of satellite images, as well as a set of tiles from this layer, using the information on the UAV's position in the previous time. Descriptors of feature points from these tiles are matched with the descriptors of feature points found on the UAV's frame (we use Euclidean metric to calculate the distance between the descriptors). Figure 1 illustrates the problem of matching the feature points: on the top you see the UAV's frame, on the bottom—set of the tiles from one of the layers of satellite image, separated by white vertical and horizontal lines. White circle restricts the possible area of the earth's surface, which is embodied in the current frame (based on the UAVs position for the previous frame). Pairs of corresponding feature points are connected by white lines.

Next, we estimate fundamental matrix F using a 5-point algorithm [20], combined with RANSAC technique [21]. Matrix F ties the coordinates of the feature points on the UAV's frame and satellite image. From the matrix F we obtain the matrix R_{rel} and transform it into angles φ, ψ, θ. The angle θ and matrix R_{prev} are used to construct matrices $R_{\mathrm{curr\_app}}$ and R_{rect}.

Figure 2 shows an example of tile of satellite image with the reference point marked on it. Figure 3 shows an example of a frame from virtual UAV's camera, and Fig. 4—the result of the geometric correction. The approximate position of the

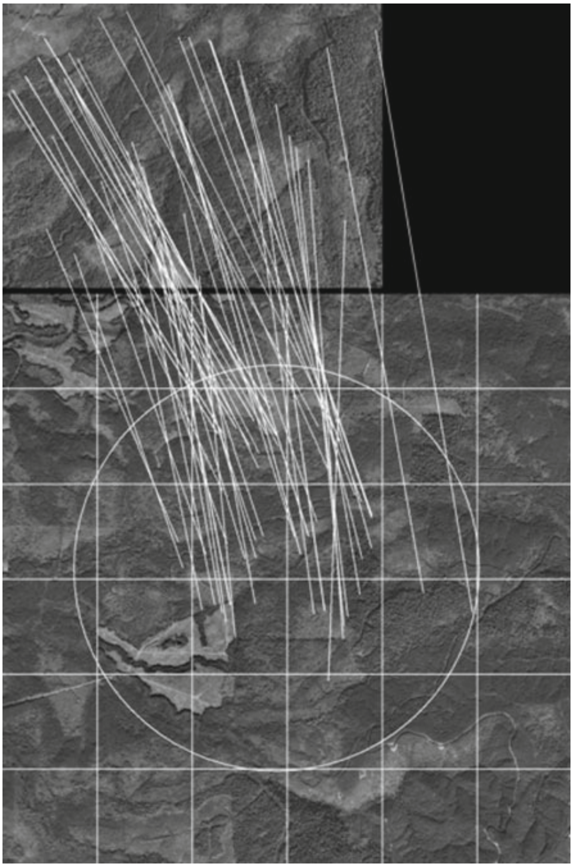


Fig. 1 Feature points matching between UAV's frame and satellite image

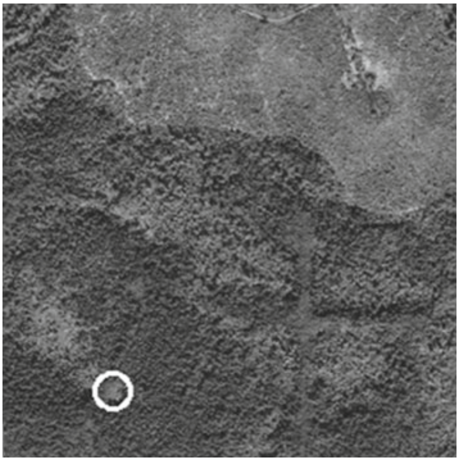


Fig. 2 Example of tile of satellite image with the reference point

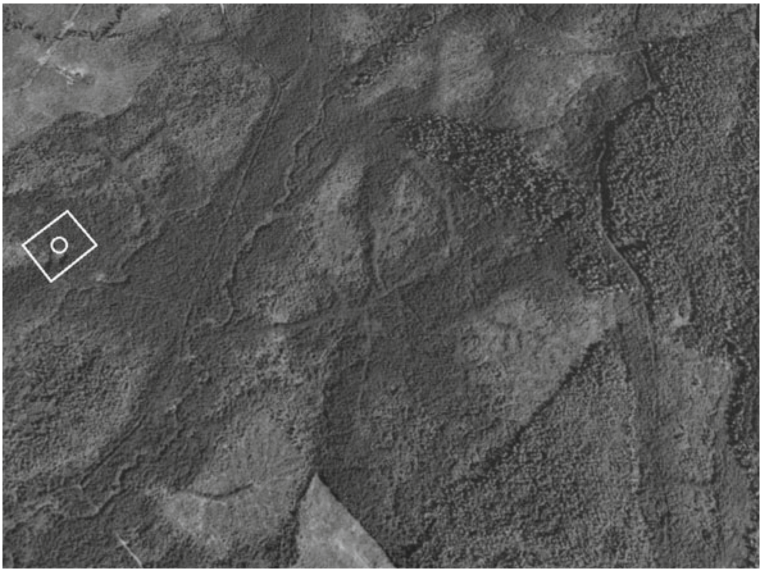


Fig. 3 Example of a frame from virtual UAV's camera

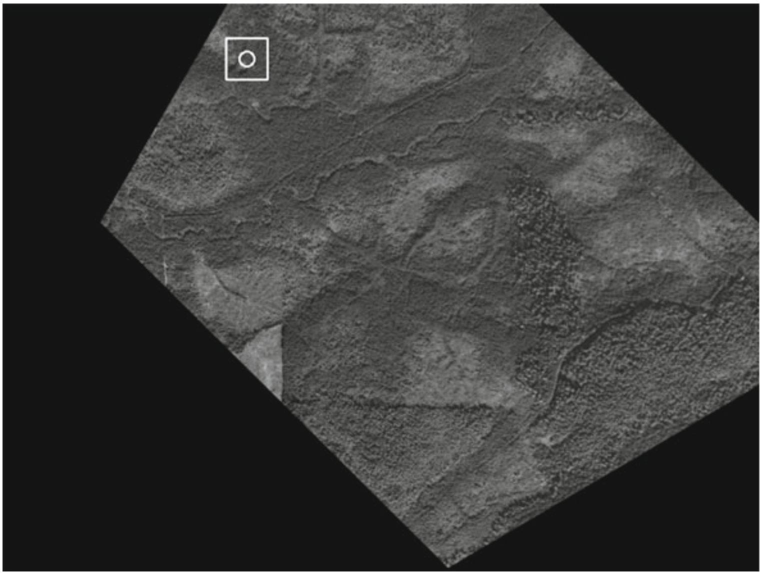


Fig. 4 Result of the geometric correction of UAV's frame

reference point is shown in Figs. 3 and 4 as white circles. The area, which was used for search of the reference point, is marked by white quadrilaterals.

The result of search of reference points is the set of pairs "2D-coordinates or reference points in UAV's frame—3D-corrdinates or reference points in global coordinate system”. These pairs are used for solving the problem of external camera calibration on the UAV. To solve it, we use capabilities of OpenCV library [14]. The described methods have been implemented in the programming language C++.

6 Experimental Results

In our experiments, we created various videos in which the virtual UAV was flying at different heights relative to the Earth’s surface. Virtual UAV could stay at a constant altitude and maneuver with increasing or decreasing the height and changing course. According to the results of testing of method of visual navigation in various videos, the maximum error in the positioning was 3 m. The method also allows calculating three navigation angles. The maximum error in determining the yaw angle was 0.2°, in determining the pitch and roll—0.01°.

7 Conclusions and Future Work

The proposed method of visual navigation of UAV in flight over a known non-planar terrain showed quite good results on synthetic data. In the related work [22], authors described a system of visual navigation for flights over flat terrain, in which the reference images and the UAV’s frames significantly different from each other. SIFT algorithm [23] was used to detect feature points and compare the images, and this algorithm is based on a methods, similar to the SURF. Therefore, we assume that in the case of using the real data from UAV, the proposed method of visual navigation will show good results. Currently, our research team is carrying out the experiments on the use of the proposed method on real models of UAVs (quadrocopters); results will be presented in the following publications. On board the UAV, we use a single-board microcomputer Raspberry Pi (latest models have quad-core processors). Also, we optimize the program code in order to achieve the performance of our algorithms in real-time.

Acknowledgments This work was supported by the Ministry of Education and Science of the Russian Federation, agreement № 14.607.21.0012 for a grant on “Conducting applied research for the development of intelligent technology and software systems, navigation and control of mobile technical equipment using machine vision techniques and high-performance distributed computing”. Unique identifier: RFMEFI60714X0012.

References

1. Grzonka, S., Grisetti, G., Burgard, W.: A fully autonomous indoor quadrotor. In: IEEE Transactions on Robotics, vol. 28(1), pp. 90–100. IEEE Press (2012)
2. Collier, J., Trentini, M., Giesbrecht, J., McManus, C., Furgale, P., Stenning, B., Barfoot, T.D., Kotamraju, S., Se, V., Jasiobedzki, P., Shang, L., Chan, B., Harmat, A., Sharf, I.: Autonomous navigation and mapping in GPS-denied environments at defence R&D Canada. In: Proceedings of NATO Symposium SET 168: Navigation Sensor and Systems in GNSS Denied Environments, p. 12. Izmir, Turkey (2012)
3. Piratla, V., Malode, S.B., Saini, S.K., Jakhotia, A., Sao, A.K., Rajpurohit, B.S., Haobijam, G.: Autonomous navigation in GPS denied indoor environment using RGBD sensor, kinect. In: Fourth Symposium on Indoor Flight Issues, p. 12 (2012)
4. Bachrach, A., Prentice, S., He, R., Roy, N.: RANGE—robust autonomous navigation in GPS-denied environments. J. Field Robot. **29** (2011)
5. Grzonka, S., Grisetti, G., Burgard, W.: Autonomous indoors navigation using a small-size quadrotor. In: Workshop Proceedings of International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR), pp. 455–463. Venice, Italy (2008)
6. Conte, G., Doherty, P.: Vision-based unmanned aerial vehicle navigation using geo-referenced information. EURASIP J. Adv. Sig. Process. 18 (2009)
7. Nwe, T.T., Htike, T., Mon, K.M., Naing, Z.M., Myint, Y.M.: Application of an inertial navigation system to the quad-rotor UAV using MEMS sensors. Proc. World Acad. Sci.: Eng. Technol. **44**, 578–582 (2008)
8. Andert, F., Adolf, F.: Online world modeling and path planning for an unmanned helicopter. Auton. Robots **27**(3), 147–164 (2009)
9. Kemp, C.: Visual control of a miniature quad-rotor helicopter. PhD thesis, p. 147. University of Cambridge (2006)
10. Johnson, N.: Vision-assisted control of a hovering air vehicle in an indoor setting. Master's thesis, p. 155. Brigham Young University (2008)
11. Ahuja, S.: Correlation based similarity measures—summary https://siddhantahuja.wordpress.com/tag/sum-of-absolute-differences-sad/
12. Michaelsen, E., Kirchhof, M., Stilla, U.: Sensor pose inference from airborne videos by decomposing homography estimates. In: Altan M.O. (ed.) International Archives of Photogrammetry and Remote Sensing, vol. 35, Part B3, p. 6 (2004)
13. Vargas, M., Malis, E.: Visual servoing based on an analytical homography decomposition. In: Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, pp. 5379–5384. Seville, Spain (2005)
14. Bradski, G., Kaehler, A.: Learning OpenCV. O'Reilly Media, p. 576 (2008)
15. Kamarainen, J.-K.: Invariance properties of gabor filter based features— overview and applications. IEEE Trans. Image Process. **15**(5), 12 (2006)
16. Bardsley, D.: Stereo vision for 3d face recognition, year 1 annual review, PhD report. University of Nottingham, p. 36 (2005)
17. GDAL, translator library for raster geospatial data formats. http://gdal.org
18. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. Comput. Vis. Image Underst. (CVIU) **110**(3), 346–359 (2008)
19. osgEarth—C++ geospatial SDK and terrain engine. http://osgearth.org/
20. Stewenius, H., Engels, C., Nister, D.: Recent developments on direct relative orientation. ISPRS J. Photogrammetry Remote Sens. **60**(4), 284–294 (2006)
21. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Comm. ACM **24**(6), 381–395 (1981)

22. Cesetti, A., Frontoni, E., Mancini, A., Ascani, A., Zingaretti, P., Longhi, S.: A visual global positioning system for unmanned aerial vehicles used in photogrammetric applications. J. Intell. Rob. Syst. **61**(1), 157–168 (2011)
23. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60** (2), 91–110 (2004)

Porting Experiment of Robotic Machining Application Using ORiN SDK and Design of 3D Printer-Like Interface

Fusaomi Nagata, Shohei Hayashi, Shingo Yoshimoto, Akimasa Otsuka, Keigo Watanabe and Maki K. Habib

Abstract Up to now, industrial robots with an open architecture have been applied to several tasks requiring high skills such as machining, sanding, and polishing. As one of the notable results, the open architecture allows system engineers at user side to design a feedback control system to develop robotic applications. However, porting of an application software into different makers' industrial robots is not easy even though having an open architecture, because the specifications of interface between PC-based controllers and robots are not standardized yet. In this paper, easy transplantation of a robotic application is tried by using ORiN (Open Robot/Resource interface for the Network) middleware interface. The application is the trajectory-following controller with vibrational motion used in a machining robot. First, the removability of undesirable cusp marks on a flat surface is examined through machining experiments. Then, portability of the controller into a small-sized educational robot VE026A with ORiN SDK (Software Development Kit) is evaluated on a simulation environment called WINCAPS III. Finally, a 3D printer-like interface is proposed for the machining robot to be controlled based on STL data. The design, implementation, and experimental results are presented.

F. Nagata (✉) · S. Yoshimoto · A. Otsuka
Tokyo University of Science, Yamaguchi, Japan
e-mail: nagata@rs.tus.ac.jp
URL: http://www.ed.yama.tus.ac.jp/nagata/

S. Hayashi
COSMOS Pharmaceutical Corporation, Fukouka, Japan

K. Watanabe
Okayama University, Okayama, Japan
e-mail: watanabe@sys.okayama-u.ac.jp

M.K. Habib
American University in Cairo, Cairo, Egypt
e-mail: maki@aucegypt.edu

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_46

1 Introduction

Activities for the standardization in industrial automation are reported in order to enhance portability, reusability, and compatibility of elaborate robotic applications [1–3]. Over the past 40 years, industrial robots have been progressed due to the teaching and playback method. From about 20 years ago, industrial robots with an open architecture have been developed from various technical points of view [4–6] and applied to several tasks requiring high skills such as machining, sanding, polishing. For example, as one of the notable results, the open architecture allows system engineers to realize desirable teachingless operation for robotic applications. However, transplantation, i.e., porting of an application software to other makers' industrial robots, is not easy even though having an open architecture, because the specifications of interface between PC controllers and robots are different, respectively. That is the reason why the standardization of industrial robots' interfaces is now expected and desired from system engineers at user side. Recently, ORiN (Open Robot/Resource interface for the Network) is gathering attention as a promising middleware for the standardization of robotic interfaces [7–9].

Figure 1 illustrates the comparison of the proposed two processes (b), (c) with the conventional process (a). Already proposed trajectory-following control of an end-effector attached to the flange of the robot is a basic but important strategy for developing a robotic application without using any robot languages (see Fig. 1b). In this paper, first of all, our past developed systems based on the concept of the process (b) are reviewed. Then, a novel trajectory-following control method with vibrational motion used in a machining robot is introduced [10] and is applied to the machining on a flat polystyrene material. The vibrational motion is considered to efficiently remove undesirable cusp marks which come out in machining process using a ball-end mill. The removability of undesirable cusp marks on a flat surface is evaluated. Then, portability of the controller into a small-sized educational robot VE026A with ORiN SDK (Software Development Kit) is investigated on a simulation environment called WINCAPS III, by comparing with the interfaces used in different makers' industrial robots [11]. The ORiN SDK is available on the VE026A.

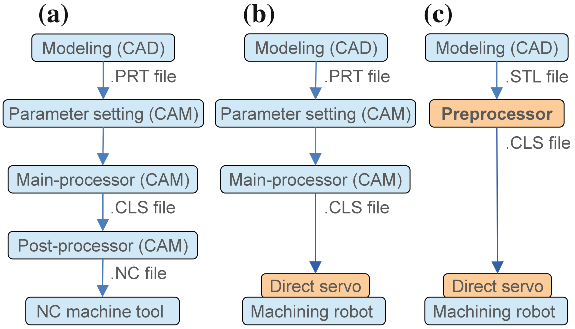


Fig. 1 Comparison of the proposed two processes with the conventional process. **a** Conventional process. **b** Proposed process 1. **c** Proposed process 2

Furthermore, a 3D printer-like interface called robotic preprocessor is introduced for the machining robot. The robotic preprocessor converts STL data into CL data as the proposed process 2 shown in Fig. 1c. Although the STL originally means stereolithography which is a file format proposed by 3D systems and recently is supported by many CAD and CAD/CAM software, it is also known as standard triangulated language in Japan. The STL is widely spread to the rapid prototyping mainly using a 3D printer which is a typical additive manufacturing system. The STL deals with a triangular representation of a 3D surface geometry [12, 13]. The robotic preprocessor allows the machining robot to be controlled along continuous triangular polygon mesh included in STL data. The design, implementation, and experimental results are presented.

2 Past Developed Robotic Systems

In this section, our past developed robotic systems using open architecture controllers are briefly introduced [14]. Although the open architecture concept allows us to efficiently develop a robotic application software without using a conventional device called a teaching pendant and a robot language, the specifications for developers provided by each robot maker had to be used firmly and minutely.

2.1 Mold Polishing Robot Developed Based on JS10

Figure 2 shows the polishing robot for PET bottle blow molds developed based on JS10 with six DOFs provided by Kawasaki Heavy Industries (KHI), LTD. in 1997. The JS10 equipped with a PC-based controller having a concept of open architecture. Six joint angles $\boldsymbol{\theta} \in \Re^{6\times1}$ [rad] could be controlled by giving relative joint angles

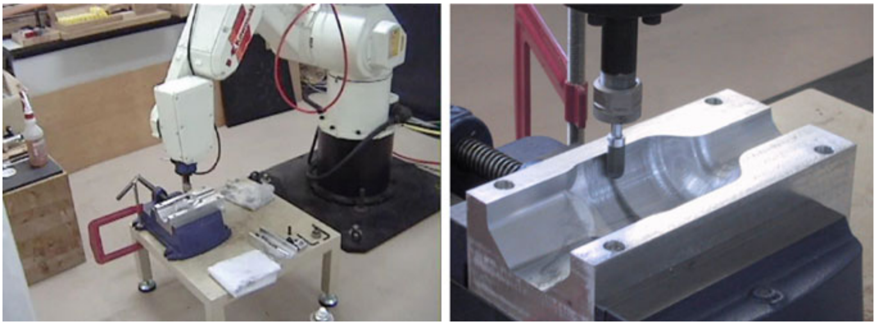


Fig. 2 Polishing robot for PET bottle blow molds developed based on JS10 provided by Kawasaki Heavy Industries, LTD. in 1997

$\Delta\boldsymbol{\theta} \in \Re^{6\times 1}$. API functions for forward kinematics ($\boldsymbol{\theta} \rightarrow$ homogeneous transformation matrix $\boldsymbol{T} \in \Re^{4\times 4}$) and inverse one ($\boldsymbol{T} \rightarrow \boldsymbol{\theta}$) were technically provided for us by KHI. In order to be able to set the sampling period shorter than 4 ms, Windows multimedia timer was used. The minimum resolution is 1 ms. In this system, the orientation of a ball-end abrasive tool, i.e., normalized direction vector, was always fixed to $[0\,0\ -1]^T$ as shown in Fig. 2 to efficiently abrade the contour of the ball-end abrasive tool. The radius of the ball-end tip was 5 mm. Cutter location data called CL data forming a zigzag path or a spiral one were generated by CAD/CAM Unigraphics and were used for the desired trajectory of the ball-end abrasive tool to perform teachingless operation. The unhandiness of the API functions was associated with the desired trajectory in Cartesian space that had to be converted to $\Delta\boldsymbol{\theta}$ using the inverse kinematics.

2.2 Robot Sander Developed Based on FS20

Figure 3 shows the robot sander for wooden workpieces with curved surface [15]. The system was developed based on FS20 with six DOFs provided by KHI in 2000, whose open architecture controller had almost the same software specification as JS10's one. In this system, the orientation of an air-driven sanding tool was controlled to be normal to surface by referring the direction components in CL data. In addition, the orientation could be manually controlled via a force-feedbacked joystick [16].

2.3 Mold Polishing Robot Developed Based on Motoman UP6

Figure 4 shows another mold polishing robot for PET bottle blow molds developed based on Motoman UP6 provided by Yaskawa Electric Corp. in 2002. Fortunately,

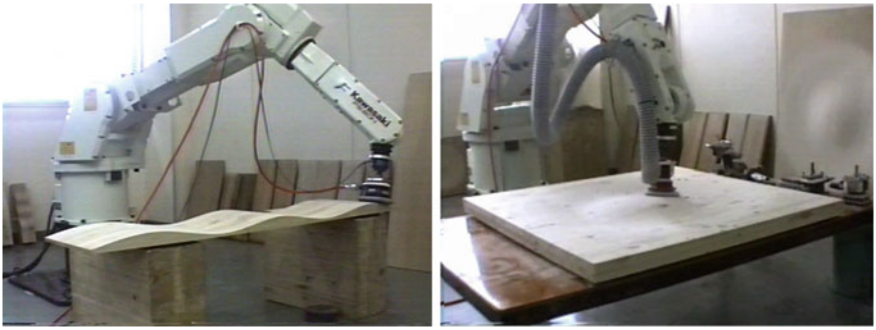


Fig. 3 Robot sander for wooden workpieces developed based on FS20 provided by Kawasaki Heavy Industries, LTD., in 2000

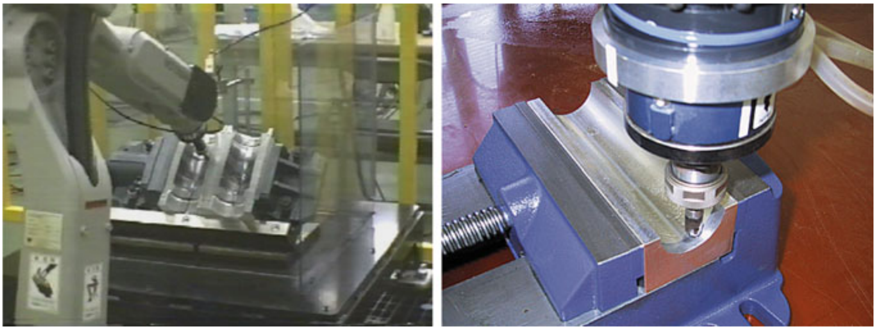


Fig. 4 Mold polishing robot for PET bottle blow molds developed based on Motoman UP6 provided by Yaskawa Electric Corp. in 2002

the robotic controller had a similar concept as FS20's one, so that the porting of the robotic application was not complicated. One of the improved points in API functions was the ability to control the tip of the robot arm directly by giving a relative homogeneous transformation matrix $\boldsymbol{\Delta T}$ to the Cartesian-based servo system for every sampling period of 10 ms. The matrix is written by

$$\Delta \boldsymbol{T} = \begin{pmatrix} \Delta n_x & \Delta o_x & \Delta a_x & \Delta p_x \\ \Delta n_y & \Delta o_y & \Delta a_y & \Delta p_y \\ \Delta n_z & \Delta o_z & \Delta a_z & \Delta p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

The position and orientation of the tool tip could be controlled using Eq. (1) without considering the corresponding six joint angles, i.e., without computing the inverse kinematics.

2.4 Air Blow Robot Developed Based on PA10

Figure 5 shows an industrial robot PA10 with seven DOFs provided by Mitsubishi Heavy Industries, Ltd. When controlling the arm tip, the position and orientation matrix of the tool tip had to be given to the servo system, which was written by the following recursive equation.

$$\boldsymbol{T}[10k] = \boldsymbol{T}\,[10(k-1)] + \Delta \boldsymbol{T}(k) \;\; (k = 1, 2, 3 \ldots) \tag{2}$$

where $\boldsymbol{T}[10k]$ is the homogeneous transformation matrix absolutely representing the position and orientation of the arm tip in robot coordinate system when the sampling

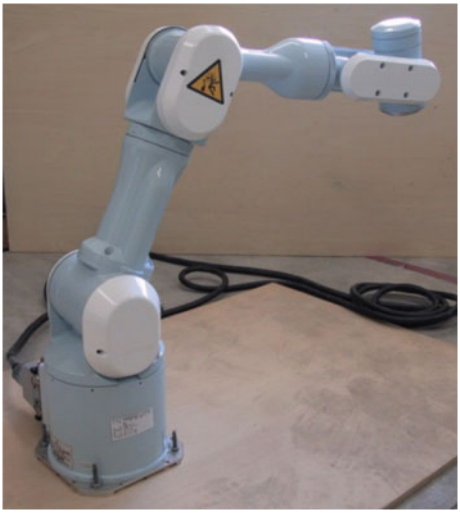


Fig. 5 Open architectural industrial robot PA10 provided by Mitsubishi Heavy Industries, Ltd., applied for painting of a relief wall in 2005

period is set to 10 ms. As can be seen, PA10 is controlled by the absolute command $\boldsymbol{T}[10k]$ in robot coordinate system. The PA10 was applied to the painting process for artistic relief wall at the Interior Design Research Institute, Fukuoka Industrial Technology Center in 2005 [17].

2.5 Machining Robot Developed Based on RV1A

A machining robot is introduced for foamed polystyrene materials [18]. Figure 6 shows the machining robot being developed based on a small industrial robot RV1A which is provided by Mitsubishi Electric Corporation. RV1A is usually used in engineering education fields. The machining robot will be used for engraving foamed polystyrene materials. When the position and orientation of the arm tip are controlled, a command vector $[\boldsymbol{x}^T(k)\,\boldsymbol{o}^T(k)]^T$ is transmitted to the robotic servo controller through Ethernet using UDP protocol. $\boldsymbol{x}(k) = [x(k)\ y(k)\ z(k)]^T$ [mm] is the position vector in robot absolute coordinate system. $\boldsymbol{o}(k) = [\phi(k)\ \theta(k)\ \psi(k)]^T$ [rad] is the orientation vector of the arm tip. Note that the orientation of the arm tip must be given with roll angle $\phi(k)$, pitch angle $\theta(k)$ and yaw angle $\psi(k)$. One of the unhandy points in developing the robotic application was that a communication program had to be implemented by using Windows Sockets API called Winsock. This implementation was a complicated and time-consuming work.

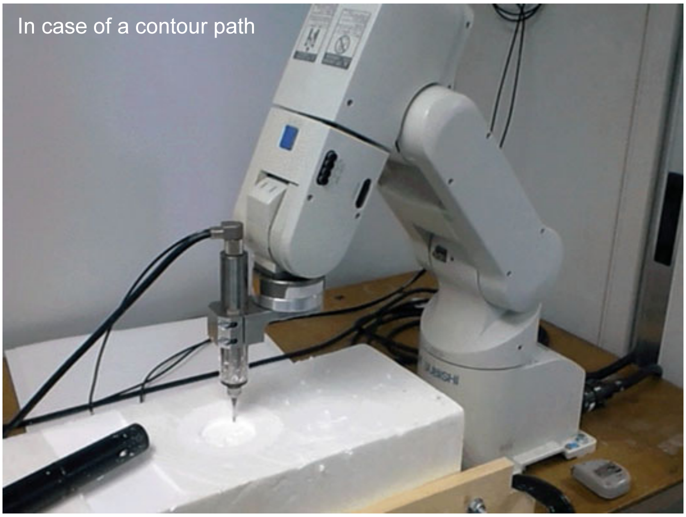


Fig. 6 Machining robot being developed based on a small industrial robot RV1A provided by Mitsubishi Electric Corp. in 2015

2.6 Trajectory-Following Controller Basically Used in Past Developed Robotic Systems

These robotic applications developed for articulated industrial robots have innovated a common important function for realizing teachingless operation desired from users. The function is the position and orientation control based on CL data called CAD/CAM-based trajectory-following controller [15]. The trajectory-following controller could be realized by simultaneously pursuing desired position and orientation of the arm tip of each robot. As an instance, Fig. 7 shows the block diagram of the trajectory-following controller incorporated in the robot sander shown in Fig. 3. $\boldsymbol{v}_t(k) \in \Re^{3\times1}$, $\boldsymbol{v}_n(k) \in \Re^{3\times1}$, and $\boldsymbol{v}_p(k) \in \Re^{3\times1}$ are velocities to control position/orientation, contact force, and pick feed, respectively. $\boldsymbol{x}_d(k)$, $\boldsymbol{o}_d(k)$, and $f_d(k)$ are desired position vector, orientation vector and force norm, respectively. $\boldsymbol{x}(k)$ and $\boldsymbol{f}(k)$ are position/orientation vector and force vector measured by respective sensors. $\boldsymbol{S}_p \in \Re^{3\times3}$ is the switch matrix to set active directions of position feedback control for accurate pick feed motion.

One of the serious problems in utilizing industrial robots provided by different robot makers is that the transplantation of a robotic application software among different makers' industrial robots is not easy even though having an open architecture because the software specifications of interface between PC-based controllers and robots are not standardized sufficiently. That is the reason why the interface standardization among different makers' industrial robots is now expected and desired from users.

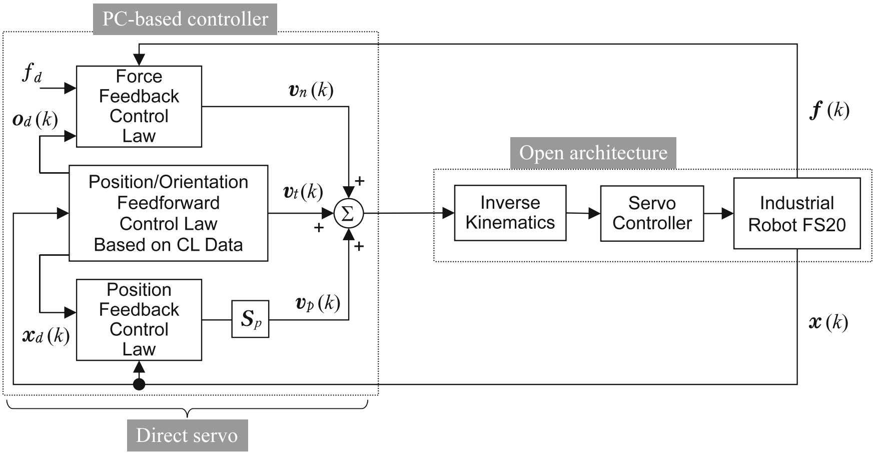


Fig. 7 Block diagram of CAD/CAM-based trajectory-following and force controllers used in the robot sander shown in Fig. 3

3 About ORiN

ORiN means Open Robot/Resource interface for the Network and provides a standard network interface for factory automations, robots, and other devices [19]. After the Japan Robot Association established the ORiN in 2002, the ORiN Forum has produced unified methods for standardization. Recently, ISO20242-Part4 including applications based on ORiN was officially approved as an international standard. For example, a robotic application software developed based on ORiN is able to not only obtain sensor information easily but also transmit control input to the servo system, without considering the difference of specifications among robot companies concerning hardware type, communication protocol, data format, and so on.

It is expected that if a robotic application, e.g., the trajectory-following controller, is developed using API functions provided by ORiN SDK, the application can be straightforwardly ported into other makers' industrial robots with ORiN middleware interface. Figure 8 illustrates an image of the advantage of using ORiN, in which a promising standardization can be seen.

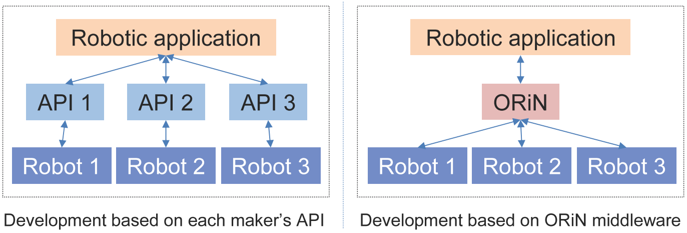


Fig. 8 Advantage of portability of robotic application through ORiN middleware interface

4 Trajectory-Following Control with Vibrational Motion

In this subsection, a trajectory-following controller with vibrational motion is proposed to efficiently remove undesirable cusp marks which appear in machining process using a ball-end mill. Figure 9 illustrates the image of the undesirable cusp marks when a zigzag path is applied. Figure 10 shows an example of base trajectory generated based on position components $\boldsymbol{p}(i-1), \boldsymbol{p}(i)$ in CL data and the vibrational motion vectors $\boldsymbol{v}_v(l-1)$, $\boldsymbol{v}_v(l)$, and $\boldsymbol{v}_v(l+1)$ online yielded in the orthogonal directions along original CL data at the discrete time $k-1$, $k+7$, and $k+15$. l is the sequential number of vibrational motion vector. When the vibrational motion is not applied, an endmill's tip moves from $\boldsymbol{p}(i-1)$ to $\boldsymbol{p}(i)$ with a given velocity v_t. On the other hand, if the vibrational motion is generated, the endmill's tip follows the trajectory as $\boldsymbol{x}_d(k-3)$, i.e., $\boldsymbol{p}(i-1) \rightarrow \boldsymbol{x}_d(k-2) \rightarrow \boldsymbol{x}_d(k-1) \rightarrow \boldsymbol{x}_d(k) \rightarrow \cdots \rightarrow \boldsymbol{x}_d(k+20) \rightarrow \boldsymbol{x}_d(k+21) \rightarrow \boldsymbol{x}_d(k+22) \rightarrow \cdots \rightarrow \boldsymbol{p}(i)$. Note that, for example, $\boldsymbol{x}_d(k)$ and $\boldsymbol{x}_d(k+4)$ are located at the same point on the round-trip path via $\boldsymbol{x}_d(k+2)$, i.e., $\boldsymbol{v}_v(l-1)$.

Here, how to calculate $\boldsymbol{v}_v(l)$ is explained. First of all, $\boldsymbol{v}_v(l)$ is perpendicular to normal direction vector $\boldsymbol{o}_d(k)$ obtained from CL data, so that the following relation is given.

$$v_{vx}(l)\, o_{dx}(k) + v_{vy}(l)\, o_{dy}(k) + v_{vz}(l)\, o_{dz}(k) = 0 \tag{3}$$

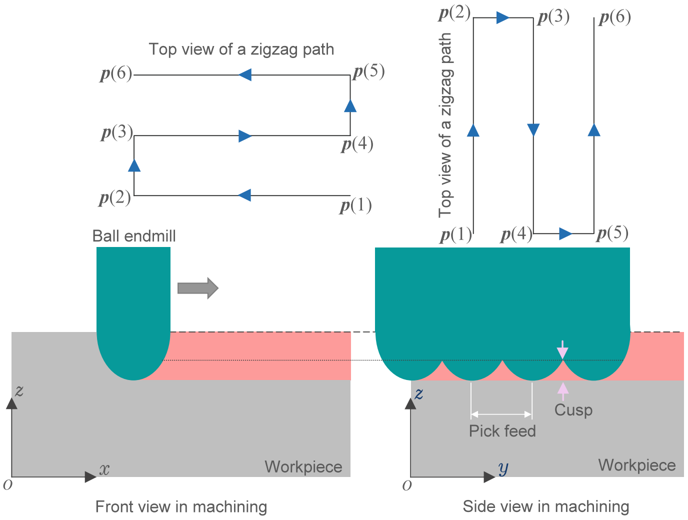


Fig. 9 Image of undesirable cusp marks caused by the trajectory of a ball-end mill in removal machining process

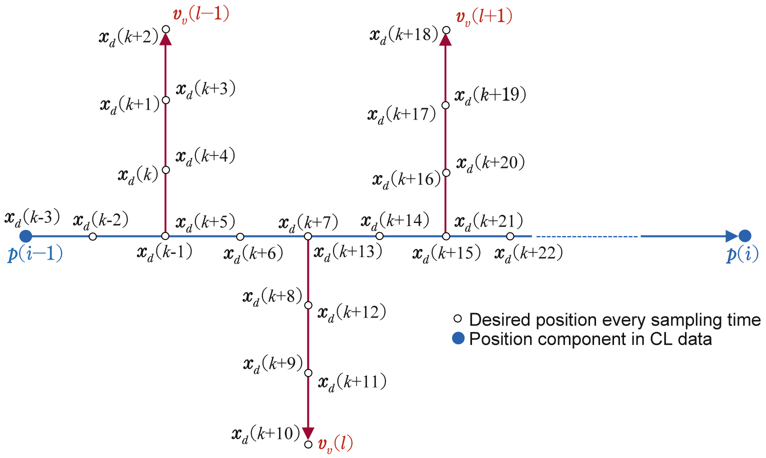


Fig. 10 Vibrational motion vector generated in the orthogonal direction along original CL data

Also, $\boldsymbol{v}_v(l)$ and tangent direction vector $\boldsymbol{t}(k)$ are orthogonal each other, so that

$$v_{vx}(l)\ t_x(k) + v_{vy}(l)\ t_y(k) + v_{vz}(l)\ t_z(k) = 0 \tag{4}$$

Because both $\boldsymbol{o}_d(k)$ and $\boldsymbol{t}(k)$ are known in advance, $\boldsymbol{v}_v(l)$ can be determined by simultaneously solving Eqs. (3) and (4). Further, by using a sign SIGN(l), a scalar K_v, and a random number $N_r(l)$ ($0 \le N_r(l) \le 1$), the vibrational motion vector is finally generated as

$$\tilde{\boldsymbol{v}}_v(l) = \mathrm{SIGN}(l) K_v N_r(l) \frac{\boldsymbol{v}_v(l)}{\|\boldsymbol{v}_v(l)\|} \tag{5}$$

where SIGN(l) is given by

$$\mathrm{SIGN}(l) = \begin{cases} 1 & \text{if } \ l = \text{odd number} \\ -1 & \text{otherwise} \end{cases} \tag{6}$$

As can be seen, the magnitude of the vibration vector depends on K_v and $N_r(l)$. $\tilde{\boldsymbol{v}}_v(l) \in \Re^{3\times 1}$ is a velocity vector to yield small and ununiform vibrational motion, which is given to a ball-end mill's tip alternately changing the direction with a given frequency.

Figure 11 shows machining results of a formed polystyrene material with a flat surface, in which zigzag paths without and with vibrational motion were given to the trajectory-following controller in the machining robot. It is observed that undesirable cusp marks could be efficiently suppressed. In the next section, the trajectory-following controller with vibrational motion is tried to be ported into a small articulated robot VE026A with ORiN middleware interface.

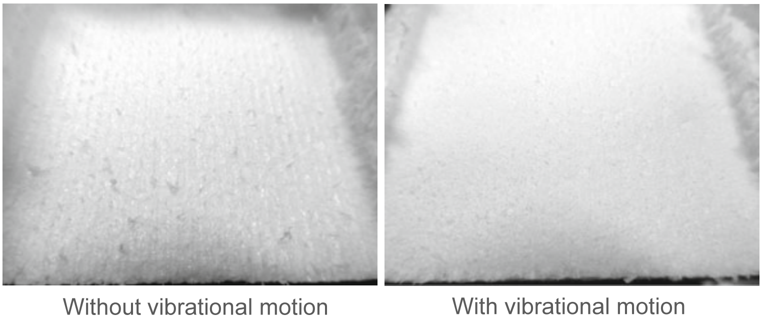


Fig. 11 Machining results of a flat foamed polystyrene without and with the proposed vibrational motion control

5 Transplantation of Trajectory-Following Controller to VE026A with ORiN Interface

The robotic applications introduced up to here are developed based on each robot maker's interface, so that the transplantation, i.e., porting among different makers' robots is complicated and time-consuming. Recently, ORiN is gathering attention as a promising middleware for standardization of robotic interface.

In this section, a porting of the CAD/CAM-based trajectory-following controller with vibrational motion is tried to evaluate ORiN SDK. Figure 12 shows the actual small-sized articulated robot VE026A and the simulation model on WINCAPS III available for ORiN SDK provided by DENSO WAVE Incorporated [20, 21]. WINCAPS III is an offline and online programming software and provides a 3D simulation function in a virtual environment.

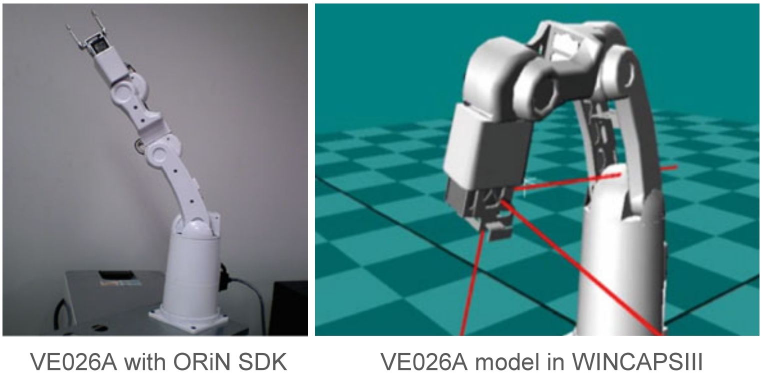


Fig. 12 Six-DOF-articulated robot arm VE026A and the simulation model incorporated with ORiN interface provided by DENSO WAVE Incorporated

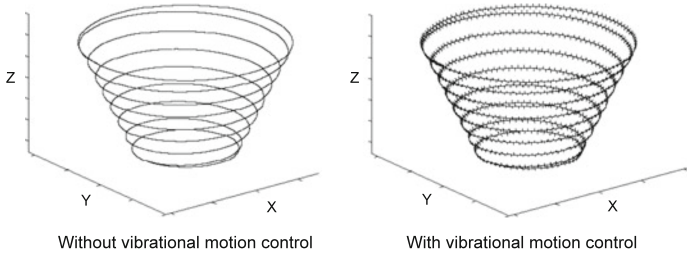


Fig. 13 Example of position control results without and with vibrational motion along a spiral path

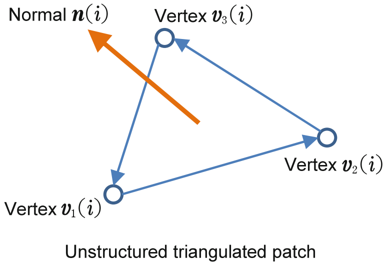
Standard programming interface called CAO (Controller Access Object) engine works as common interface and functions between the application and the robot. Among methods provided by CAO engine, CaoGetPose( ) was used to get the current position and orientation $[\boldsymbol{x}^T(k)\ \boldsymbol{o}^T(k)]^T$ of the arm tip. Also, CaoMove( ) was used in order to give the desired position and orientation $[\boldsymbol{x}_d^T(k)\ \boldsymbol{o}_d^T(k)]^T$ to the VE026A every sampling period. In other words, it is expected that we have only to basically use CaoGetPose( ) and CaoMove( ) in order to transplant the trajectory-following controller with vibrational motion into the VE026A.

After the implementation on Windows Visual C++ with a little effort, it was confirmed that the tip of the robot arm could be successfully controlled with a command vector similar to the case of the machining robot developed based on RV1A. It should be noted that the orientation was dealt with $\boldsymbol{o}(k) = [\phi(k)\ \theta(k)\ \psi(k)]^T$ [deg], i.e., not radian but degree in case of ORiN SDK. Figure 13 shows examples of the control results without and with vibrational motion control. It was further confirmed that the ORiN SDK enabled the application software, i.e., the trajectory-following controller with vibrational motion, to be easily transplanted into the VE026A with a little effort [11].

6 3D Printer-Like Interface for a Machining Robot

Our current interest is to enable the machining robot to run based on STL data consisting of unstructured triangulated patches as shown in Fig. 14. In this section, a preprocessor is proposed to convert STL data into CL data. If the machining robot works following the converted CL data; consequently, it can be said that the robot is controlled based on the STL data. Our proposed robotic machining preprocessor consists of two processes, i.e., essential process and smart one, to realize a 3D printer-like interface.

Fig. 14 Unstructured triangular patch used in STL file


Normal $\boldsymbol{n}(i)$
Vertex $\boldsymbol{v}_3(i)$
Vertex $\boldsymbol{v}_2(i)$
Vertex $\boldsymbol{v}_1(i)$
Unstructured triangulated patch

6.1 Essential Process

The STL means stereolithography which is a file format proposed by 3D systems and recently is supported by many CAD and CAD/CAM softwares. STL is known as standard tessellation language using multiple unstructured triangular patches as shown in Fig. 14, in which $\boldsymbol{n}(i) \in \Re^{3\times1}$ is normal vector, $\boldsymbol{v}_1(i)$, $\boldsymbol{v}_2(i)$ and $\boldsymbol{v}_3(i) \in \Re^{3\times1}$ are position vectors, respectively. i denotes the sequential number of the patches in a STL file. The contents of triangulated patches in a binary file format are composed of 80 characters header, number of triangulated patches, i pairs consisting of normal vector to triangle face, position vectors of three vertexes, and attribute byte count, which are written as

char[80]	//Header
uint32	//Number i of triangles included in a STL file
float[3]	//Normal vector $\boldsymbol{n}(1)$
float[3]	//Vertex vector $\boldsymbol{v}_1(1)$
float[3]	//Vertex vector $\boldsymbol{v}_2(1)$
float[3]	//Vertex vector $\boldsymbol{v}_3(1)$
uint16	//Attribute byte count
float[3]	//Normal vector $\boldsymbol{n}(2)$
float[3]	//Vertex vector $\boldsymbol{v}_1(2)$
float[3]	//Vertex vector $\boldsymbol{v}_2(2)$
float[3]	//Vertex vector $\boldsymbol{v}_3(2)$
uint16	//Attribute byte count
⋮	⋮
float[3]	//Normal vector $\boldsymbol{n}(i)$
float[3]	//Vertex vector $\boldsymbol{v}_1(i)$
float[3]	//Vertex vector $\boldsymbol{v}_2(i)$
float[3]	//Vertex vector $\boldsymbol{v}_3(i)$
uint16	//Attribute byte count

As can be seen from Fig. 14, three "GOTO" statements for CL data can be generated from ith triangulated patch as

GOTO/Vertex vector $\boldsymbol{v}_1^T(i)$, Normal vector $\boldsymbol{n}^T(i)$
GOTO/Vertex vector $\boldsymbol{v}_2^T(i)$, Normal vector $\boldsymbol{n}^T(i)$
GOTO/Vertex vector $\boldsymbol{v}_3^T(i)$, Normal vector $\boldsymbol{n}^T(i)$

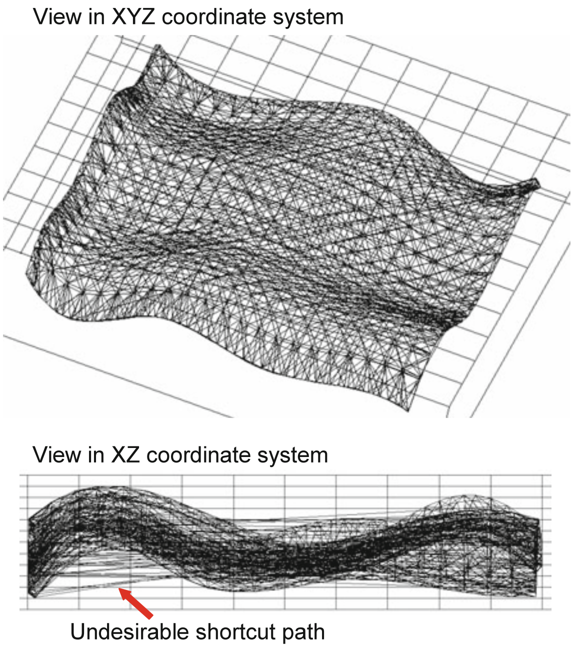
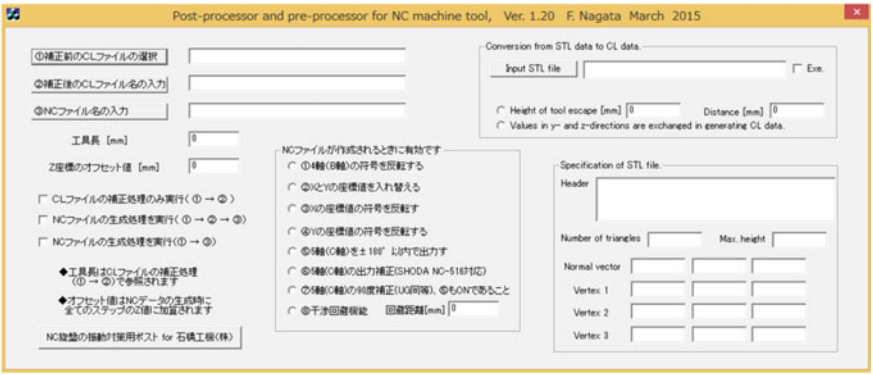

View in XYZ coordinate system
View in XZ coordinate system
Undesirable shortcut path

Fig. 15 An example of CL data directly converted from STL data

This is the essential process to convert STL data into CL data. Figure 15 shows an example of output of converted CL data (STL data → CL data), in which the original STL data are generated by CAD/CAM Creo. The CL data are used as unique desired trajectory based on continuous triangular polygon mesh. As can be seen, if the conversion is conducted without any device and consideration, undesirable shortcut paths frequently appear. This means that sometimes two adjacent triangulated patches saved in STL file do not neighbor each other but are far away in the actual coordinate system. When the STL data are used for additive layered machining with a 3D printer, there is no problem. However, if the CL data converted from the STL data are applied to the removal machining with a ball-end mill, serious mis-removal cuttings tend to occur according to the frequency. This means that parts to be left are forced to be cut.

6.2 *Smart Process*

To overcome the problem of mis-removal cutting, a smart process is additionally proposed in the robotic machining preprocessor. Figure 16 shows the Windows dialogue designed for the robotic machining preprocessor and post-processor. There are two parameters for the smart process in the robotic machining preprocessor. One is the height h [mm] of tool escape. The other is the distance $d(i) = \|\boldsymbol{v}_1(i) - \boldsymbol{v}_3(i-1)\|$ [mm] from the vertex $\boldsymbol{v}_3(i-1)$ to the $\boldsymbol{v}_1(i)$. If $d(i)$ is longer than the distance set in the



Post-processor and pre-processor for NC machine tool, Ver. 1.20 F. Nagata March 2015

①補正前のCLファイルの選択

②補正後のCLファイル名の入力

③NCファイル名の入力

工具長 [mm] 0

Z座標のオフセット値 [mm] 0

□ CLファイルの補正処理のみ実行(①→②)

□ NCファイルの生成処理を実行(①→②→③)

□ NCファイルの生成処理を実行(①→③)

◆工具長はCLファイルの補正処理
(①→②)で参照されます

◆オフセット値はNCデータの生成時に
全てのステップのZ値に加算されます

NC旋盤の振動対策用ポスト for 石橋工機(株)

NCファイルが作成されるときに有効です

○ ①4軸(B軸)の符号を反転する

○ ②XとYの座標値を入れ替える

○ ③Xの座標値の符号を反転す

○ ④Yの座標値の符号を反転する

○ ⑤5軸(C軸)を±180° 以内で出力す

○ ⑥5軸(C軸)の出力補正(SHODA NC-516対応)

○ ⑦5軸(C軸)の90度補正(UG同等)、⑥もONであること

○ ⑧干渉回避機能 回避距離[mm] 0

Conversion from STL data to CL data.

Input STL file □ Exe.

○ Height of tool escape [mm] 0 Distance [mm] 0

○ Values in y- and z-directions are exchanged in generating CL data.

Specification of STL file.

Header

Number of triangles Max. height

Normal vector

Vertex 1

Vertex 2

Vertex 3

Fig. 16 Windows dialogue designed for robotic machining preprocessor to process STL data

Fig. 17 Smart path generated through the proposed preprocessor

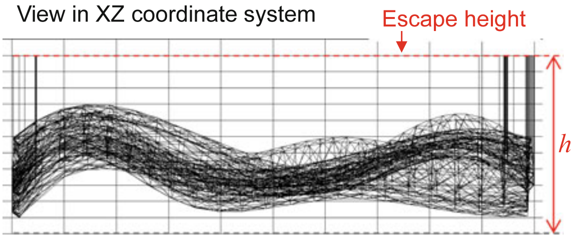
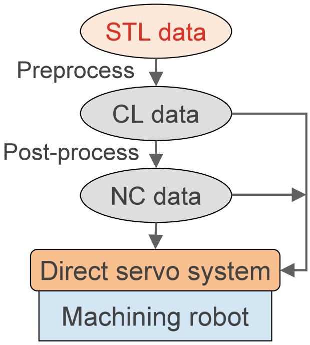


Fig. 18 Direct servo system available for NC data, CL data, and STL data



dialogue, then the preprocessor inserts two positions in upper direction for escaping. The first position is for the escape just upper direction. Also, the second position is for the movement from the current patch to the next one. Figure 17 shows an example of a smart path written with the style of CL data converted from original STL data using the proposed preprocessor. As can be seen, it is confirmed that the path is successfully processed so as to prevent mis-removal cuttings.

As shown in Fig. 18, the proposed machining robot can run based on not only NC data and CL data but also STL data due to the results introduced in this section.

7 Conclusions

In this paper, our past developed robotic applications were first introduced, and the necessity of standardization of robotic interfaces provided by different makers was described for easier transplantation of a robotic application. Then, a novel trajectory-following control with vibrational motion implemented in a machining robot was explicated and applied to the removal machining of a flat polystyrene material. The removability of undesirable cusp marks on the flat surface was examined through actual experiments. Further, portability of the controller into a small-sized educational robot VE026A with ORiN SDK was examined on a simulation environment called WINCAPS III. Consequently, higher reusability and portability due to using ORiN SDK were confirmed through the design and experiments. Finally, a robotic preprocessor was proposed for the machining robot to realize a 3D printer-like interface by converting STL data into CL data. The robotic preprocessor allowed the machining robot to be controlled based on the information included in STL data without using any CAM process.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 25420232.

References

1. Shin, S.: What are the standards in industrial automation sector? In: Proceedings of ICCAS-SICE 2009, pp. 945–946 (2009)
2. Borangiu, T., Anton, F.D., Anton, S.: Open architecture for robot controllers. In: Proceedings of 2010 IEEE 19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD), pp. 181–186 (2010)
3. Ying, Z., Tianmiao, W., Hongxing, W., Miao, L.: Robot software architecture based on IPv6. In: Proceedings of 6th IEEE Conference on Industrial Electronics and Applications, pp. 1666–1671 (2011)
4. Liandong, P., Xinhan, H., Arif, M.: A PC-based open architecture controller for robot. Inf. Technol. J. **3**(3), 296–302 (2004)
5. Garcia, J.G., Ortega, J.G., Nicto, L.N., Garcia, A.S.: Design and validation of an open architecture for an industrial robot. In: Proceedings of 2007 IEEE International Symposium on Industrial Electronics, pp. 2004–2009 (2007)
6. Connolly, C.: KUKA robotics open architecture allows wireless control. Ind. Robot: Int. J. **35**(1), 12–15 (2008)
7. Mizukawa, M., Koyama, T., Inukai, T., Noda, A., Kanamaru, N., Noguchi, Y., Otera, N.: Proposal of open-network-interface for industrial robots (ORiN) and its experimental evaluation. In: Proceedings of 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 689–694 (2001)
8. Mizukawa, M., Matsuka, H., Koyama, T., Inukai, T., Noda, A., Tezuka, H., Noguchi, Y., Otera, N.: ORiN: open robot interface for the network—the standard and unified network interface for industrial robot applications. In: Proceedings of the 41st SICE Annual Conference, pp. 925–928 (2002)
9. Mizukawa, M., Sakakibara, S., Otera, N.: Implementation and applications of open data network interface ORiN. In: Proceedings of the 43rd SICE Annual Conference, pp. 1340–1343 (2004)

10. Nagata, F., Hayashi, S., Nagatomi, T., Otsuka, A., Watanabe, K., Habib, M.K.: Robotic trajectory following controller with a capability for generating micro vibrational motion along curved surface. In: Proceedings of the 41st Annual Conference of the IEEE Industrial Electronics Society (IECON2015), pp. 765–770 (2015)
11. Nagata, F., Hayashi, S., Nagatomi, T., Otsuka, A., Watanabe, K.: Trajectory following controller with vibrational motion developed based on ORiN middleware interface. In: Proceedings of the 3rd International Conference on Industrial Application Engineering 2015, pp. 59–64 (2015)
12. Szilvasi-Nagy, M., Matyasi, G.: Analysis of STL files [J]. Math. Comput. Model. **38**(7/9), 945–960 (2003)
13. Iancu, C., Iancu, D., Stancioiu, A.: From CAD model to 3D print via "STL" file format. Fiability Durab. **1**(5), 73–81 (2010)
14. Nagata, F., Watanabe, K.: Controller Design for Industrial Robots and Machine Tools: Applications to Manufacturing Processes. Woodhead Publishing, UK (2013)
15. Nagata, F., Kusumoto, Y., Fujimoto, Y., Watanabe, K.: Robotic sanding system for new designed furniture with free-formed surface. Robot. Comput.-Integr. Manuf. **23**(4), 371–379 (2007)
16. Nagata, F., Watanabe, K., Kiguchi, K., Tsuda, K., Kawaguchi, S., Noda, Y., Komino, M.: Joystick teaching system for polishing robots using fuzzy compliance control. In: Proceedings of 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 362–367 (2001)
17. http://www.fitc.pref.fukuoka.jp/kenkyu/report/h17/17-15.pdf (in Japanese)
18. Nagata, F., Otsuka, A., Nakamura, K., Watanabe, K.: Proposal of a machining robot for foamed polystyrene materials. In: Proceedings of 2nd International Conference on Electrical, Control and Computer Engineering (InECCE2013), pp. 279–284 (2013)
19. ORiN FORUM, http://www.orin.jp/e/
20. VE026A ORiN Option Users' Guide, Version 1.0.0, DENSO WAVE Inc., Dec 14th (2012)
21. ORiN2 Programming Guide, Version 1.0.12.0, DENSO WAVE Inc., Sept 7th (2012)

Rotation Vector Sensor-Based Remote Control of a Mobile Robot via Google Glass

Xi Wen, Yu Song, Wei Li and Genshe Chen

Abstract Google Glass, as a representative product of wearable Android devices, provides a new way of Human Machine Interaction. In this paper, we propose a method for the control of a Surveyor mobile robot via a Google Glass. In doing it, we establish Wi-Fi communication between the Google Glass and the mobile robot, wear the Google Glass to observe robot states and its surroundings, and navigate the mobile robot through head movements detected by rotation vector sensor mounted on the Google Glass. The method allows us to completely free hands in navigating the robot without need for a computer monitor. In order to demonstrate the flexibility of the proposed method, we control the robot to go through a maze in a simulated environment via the Google Glass.

The work is supported in part by The National Natural Science Foundation of China (No. 61473207) and The State Key Laboratory of Robotics at Shenyang Institute of Automation (Grant No. 2014-Z03).

X. Wen · Y. Song · W. Li (✉)
School of Electrical Engineering and Automation, Tianjin University,
Tianjin 300072, China
e-mail: wli@csub.edu

X. Wen
e-mail: wenxi203120@tju.edu.cn

Y. Song
e-mail: songyu@tju.edu.cn

W. Li
Shenyang Institute of Automation, Chinese Academy of Sciences,
Shenyang 110016, China

W. Li
Department of Computer & Electrical Engineering and Computer Science,
California State University, Bakersfield, CA 93311, USA

G. Chen
Intelligent Fusion Technology, Inc, Germantown, MD 20876, USA
e-mail: gchen@intfusiontech.com

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_47

1 Introduction

In April 2012, the Google Inc. launched Google Glass (Google Project Glass), called "expanding reality" glass. Recently, the Google Inc. announced that the new model of Google Glass will be coming. It is a wearable computer with an optical head mounted display (OHMD) to display information dissimilarly to a smart phone in a small screen size. Wearers communicate with Google Glass via natural language voice commands or via gestures using the touchpad located on the side of the Google Glass. What's more, it has some functions similar to a smart phone, for example, powerful information processing, achieving photographs and videos, navigation ability, Wi-Fi Internet access, processing text scripts, and sending e-mails, etc. Without doubt, Google Glass will become a new Human Machine Interaction (HMI) device. Currently, Google Glass has been actively used in some fields. Through Google Glass, a doctor observed the patient's electrocardiogram (ECG) to avoid distraction [1] or conveniently recorded the operation process [2]. Google Glass was also used as a guidance system for manual assembling tasks to help free hands and achieve good results [3]. For community research, sociologists collected information society through Google Glass [4]. In the field of education, an instructor used Google Glass to demonstrate physics experiments [5].

In this paper, we investigate the remote control of mobile robot systems via Google Glass. Based on live video feedback, an operator makes the corresponding head-movement to send the control command to a Surveyor robot in a virtual environment designed by the Webots simulator [6]. Our system consists of two sections: the Google Glass section and Robot section, as shown in Fig. 1. In order to demonstrate the performance of the proposed Google Glass-based remote controller, we design a maze task for the robot, and discuss the operating conditions of the robot at different speeds. Following Introduction, Sect. 2 designs the user interface of the Google Glass-based control system. Section 3 introduces how to use Webots to build a simulation environment. Section 4 describes the experimental

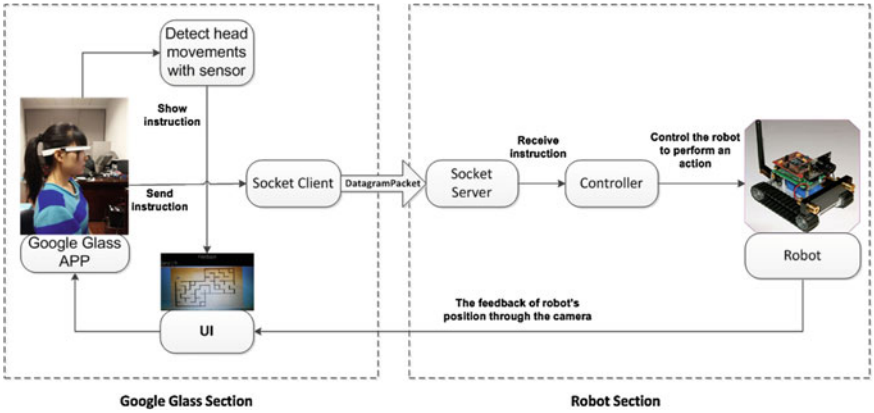


Fig. 1 The Google Glass-based remote control system

process and shows the experimental results. Section 5 compares the Google Glass rotation vector sensor-based control system with a touchpad-based control system and discusses further research.

2 Google Glass-Based Control Scheme

2.1 Google Glass

Google Glass runs on the operating system Android 4.4.2 (KitKat) with an OMAP 4430 SoC1.2 GHz Dual Core (ARMv7) processor manufactured by the Texas Instruments. It has 2 GB of memory, 16 GB of storage (12 GB available) and a 5-megapixel camera and supports both Wi-Fi and Bluetooth [7]. It is composed by six sections, respectively: a GPS, a camera, a prism, a speaker, a microphone, and a battery as shown in Fig. 2.

The unique design of Google Glass provides a user with different control experience compared to a smart phone or tablet. The main user interface on Google Glass is Timeline that is a set of 640 × 360 pixel cards organized into several sections, for example “Home”, “Past” and “Present & Future”. In addition, Google Glass supports three types Glassware: Mirror API Glassware, GDK Glassware and Hybrid Glassware, respectively. In this paper, we develop the control system using a GDK Glassware [7].

2.2 The Principle of Rotation Vector Sensor

Google Glass supports a variety of sensors, such as accelerometer, gyroscope, geomagnetic sensor, direction sensor, and rotation vector sensor. They are divided into two types: software-based sensor and hardware-based sensor respectively. In this paper, the rotation vector sensor is used to detect the operator’s head movements to resolve operator’s control intents [7].

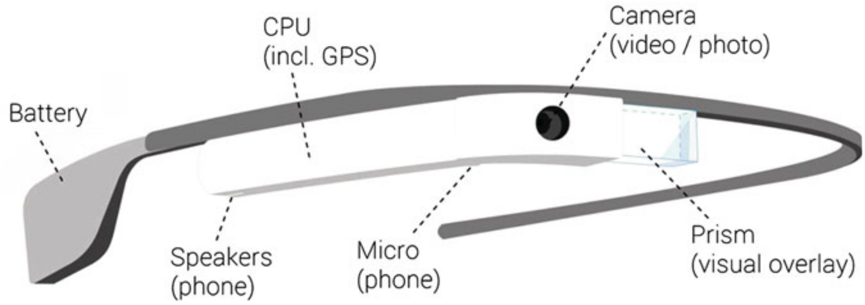


Fig. 2 Structure of Google Glass

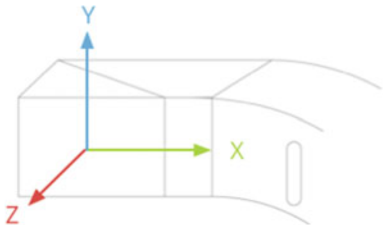


Fig. 3 The Google Glass sensor coordinate system

The rotation vector sensor, a software-based sensor, is able to calculate device rotation angle [8]. It is composed of an acceleration sensor, a geomagnetic sensor and a gyro. The Google Glass sensor coordinate system shown in Fig. 3 is relative to the Google Glass display [7]. We estimate the direction of the operator's head movement by the angle difference of the rotation matrix [8].

2.3 Remote Control Scheme

We develop an Android application programmed in Java and design a user interface for the control scheme based on the Google Glass's rotation vector sensor. The control scheme consists of two sections: the user interface and the client communication.

The user interface as shown in Fig. 1 presents live video to the operator, which displays real-time scenes that the Google Glass camera captures. We set up the control commands associated to head movements: moving forward, turning left, turning right and pause, which are respectively corresponding with head-up, head-turning-left, head-turning-right, and head-down. When detecting a movement, the controller sends its corresponding command to the robot. The client communication is developed by socket communication based on Java platform through Wi-Fi network. The communication ports serve as a server and a client, respectively. Google Glass acts as the control command client port.

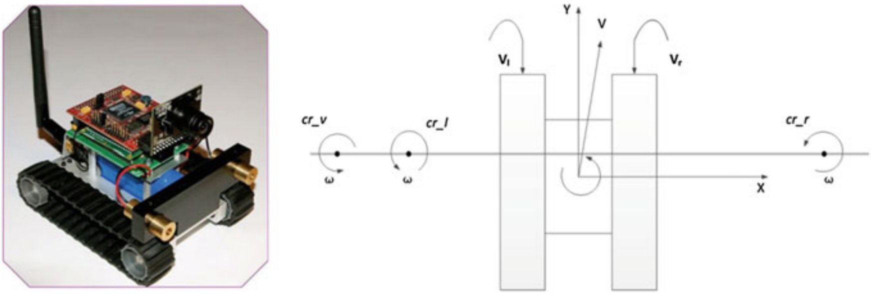


Fig. 4 SRV-1 robot and its attached coordinated system

3 Surveyor Robot

The Surveyor SRV-1 Blackfin Robot used in our system is an Open Source Wireless Mobile Robot with a video camera, which is suitable for research, education, and exploration. SRV-1 carries out Wi-Fi communication via Lantronix Matchport WLAN 802.11 g radio module. A mass of third party software support make it widely applied to many fields. For example, the Webots simulator offers the simulated model of the SRV-1 [9].

3.1 Surveyor Kinematic Model

The robot Surveyor SRV-1 is a differential wheeled robot. Its kinematic model is given in [10]

$$v_d = \frac{v_r - v_l}{x_{cr\_r} - x_{cr\_l}} \left(\frac{x_{cr\_r} + x_{cr\_l}}{2} \right). \tag{1}$$

$$v_x = \frac{v_r - v_l}{x_{cr\_r} - x_{cr\_l}} y_{cr\_v}. \tag{2}$$

$$v_y = \frac{v_r + v_l}{2} - v_d. \tag{3}$$

$$\omega = \frac{v_r - v_l}{x_{cr\_r} - x_{cr\_l}}. \tag{4}$$

where v_x and v_y are the linear speeds of the robot on the X and Y coordinate axes, ω is the angular speed of the robot, v_r and v_l are the linear velocities of the right and left wheels, x_{cr_r} and x_{cr_l} are the X coordinates of the instantaneous rotation centers of the right and left wheels, and y_{cr_v} is the Y coordinate of the instantaneous rotation center of the robot, as shown in Fig. 4. The inverse kinematics is given by:

$$v_l = v_y + x_{cr\_l}\omega. \tag{5}$$

$$v_r = v_y + x_{cr\_r}\omega. \tag{6}$$

3.2 Robot Controller

We developed a simulation module by upgrading the simulation model of the Surveyor robot with a robot controller offered by Webots to carry out the evaluation. It offers, which can be upgraded by users. Webots is a three-dimensional mobile robot simulator originally developed as a research tool for investigating various control algorithms of mobile robots. This software provides a powerful development environment for modeling, programming and simulation of mobile robots.

The module of our system is made up of three sections: the communication server, the robot controller, and the Surveyor robot graphical model. We receive the Data Packet through the server from Google Glass, parse the instruction and control the robot via the robot controller in Webots.

4 Experiment

First, the operator wearing the Google Glass started the App through the "Start a Task" voice command, inputted the server IP address, and entered into the console interface. The Google Glass calibration completed once the screen displayed "Calibration Successful". Second, the operator loaded the controller for the World and opened the simulation switch in Webots. Afterward, the operator finished the preparatory work of experiment.

In order to verify the characteristics of our system, we designed a maze experiment in which the operator navigated the mobile robot via Google Glass. The maze structure is illustrated in Fig. 5. We conducted three series of experiments which are divided into S1, S2 and S3 according to robot speeds. The speed of each group is increasing in proportion as listed in Table 1. During the entire experiment, the operator only needed to make the corresponding head movement to drive the robot in the correct path to reach the destination. It is obvious that increasing the speed of robot reduces the execution time but makes the control more difficult. In the experiments, we also recorded the consuming-time and length of the path traversed, as listed in Table 2.

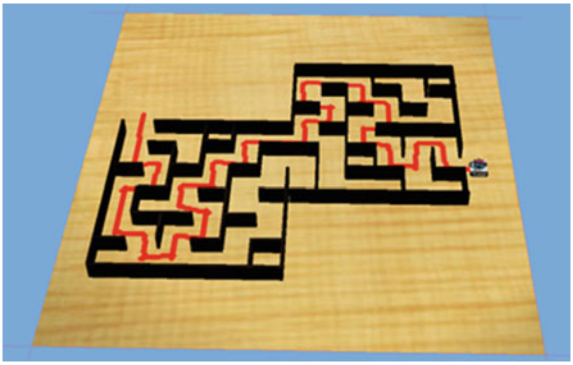


Fig. 5 The maze and trajectory

Table 1 Velocity distribution

Speed	Forward (m/s)	Left (m/s)	Right (m/s)	Pause (m/s)
S1	0.05	0.05	0.05	0.00
S2	0.1	0.1	0.1	0.00
S3	0.2	0.2	0.2	0.00

Table 2 Time and trajectory length

Speed	Time				Path-length (m)
	Forward (s)	Left (s)	Right (s)	Pause	
S1	145.18	32.74	31.78	–	7.24
S2	72.43	17.06	17.17	–	7.24
S3	42.50	15.47	13.95	–	8.50

5 Discussion

In this paper, we propose a novel control scheme based on the Google Glass rotation vector sensors. Compared to the ordinary device, it makes the operator generate a sense of interaction with the "expanding reality" glass.

By analyzing the experimental data, we find that increasing the speed reduces the task consuming-time approximately in proportion, but the path length is roughly constant. This conclusion well reflects the effectiveness and stability of our system. In addition, we compare this system with a touchpad one and draw a histogram shown in Fig. 6. We note the system based on rotation vector sensor is not inferior to the system based on touchpad. What's more, the performance yielded by the rotation vector sensor-based controller is better than one yielded by the touchpad-at high speed. It fully reflects the flexibility of this system. There are also three advantages about the system proposed in this paper, as followings:

(1) The selection of instructions is more consistent with the operator's habits;
(2) The instruction switch is more natural;
(3) The robot control achieves the goal of completely liberating hands.

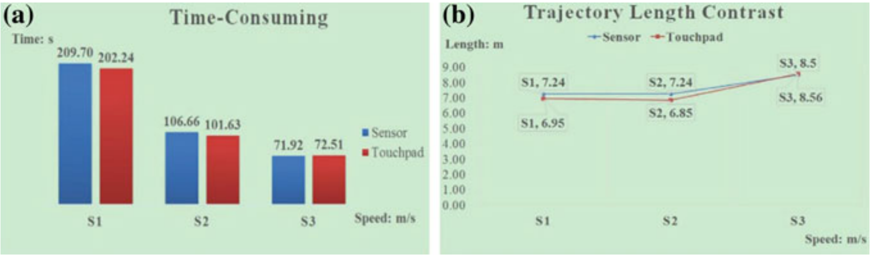


Fig. 6 The comparative results

All of advantages show the system based on the rotation vector sensor is a stable, successful and feasible system.

In the future, we will further explore the application of Google Glass in the robot research, which makes using Google Glass to control the robot more intelligent and more comfortable, and then we will apply this system to control the real robot.

References

1. Jeroudi, O., Christakopoulos, G., Kotsia, A., Kypreos, A., Rangan, B., Banerjee, S., Brilakis, E.: Accuracy of remote electrocardiogram interpretation with the use of Google Glass technology. Am. J. Cardiol. **115**, 374–377 (2015)
2. http://archopht.jamanetwork.com/article.aspx?articleid=2210534
3. Wille, M., Scholl, P., Wischniewski, S., Laerhoven, K.: Comparing Google Glass with Tablet-PC as guidance system for assembling tasks. In: 11th International Conference on Wearable and Implantable Body Sensor Networks Workshops, pp. 38–41. IEEE Press, Zurich (2014)
4. Paxton, A., Rodriguez, K., Dale, R.: PsyGlass: capitalizing on Google Glass for naturalistic data collection. Behav. Res. Methods **47**, 608–619 (2015)
5. Weppner, J., Lukowicz, P., Hirth, M., Kuhn, J.: Physics education with Google Glass physics experiment app. In: ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, pp. 279–282. Association for Computing Machinery, Inc., Seattle (2014)
6. https://www.cyberbotics.com/
7. https://developers.google.com/glass/design/
8. http://wear.techbrood.com/guide/topics/sensors/sensors_motion.html
9. http://www.robotreviews.com/reviews/surveyor-srv-1-blackfin-robot-review
10. Neamtu, D., Fabregas, E., Wyns, B., Keyser, R., Dormido, S., Inoescu, C.: A remote laboratory for mobile robot applications. In: 18th International Federation of Automatic Control World Congress, pp. 7280–7285. IFAC Secretariat, Schlossplatz 12, A-2361 Laxenburg, A-2361, Australia, Milano (2011)

A Comparative Study of Wi-Fi and Bluetooth for Signal Strength-Based Localisation

Ryoma J. Ohira, Tommi Sullivan, Andrew J. Abotomey and Jun Jo

Abstract With the growing need for indoor localisation solutions, this paper investigates the practical applications of wireless networking technologies based on the empirical study. By comparing between the two most widely used wireless technologies, aims to identify which technology, between Wi-fi and Bluetooth, is more capable in RSS-based localisation. Field experiments were conducted in order to collect the data to model the propagation of the two technologies. This study demonstrates that, through comparing the derived models to empirical data, Bluetooth has the potential to improve indoor localisation methods due to its more accurate model.

1 Introduction

GPS is one of the best localisation techniques created to discover a position outdoors; however, GPS has major difficulties for indoor localisation. With extensive time being spent as research into indoor localisation, a significant increase in accuracy has been achieved. Nonetheless, one of the biggest hurdles in indoor localisation systems is the reliability and accuracy of deriving the distance from a signal between two nodes.

Local positions can be calculated through various methods that can be categorised into received signal strength (RSS), angle of arrival (AOA) and propagation

R.J. Ohira · T. Sullivan · A.J. Abotomey · J. Jo (✉)
School of Information Communication and Technology,
Griffith University, Queensland, Australia
e-mail: j.jo@griffith.edu.au

R.J. Ohira
e-mail: ryoma.ohira@griffithuni.edu.au

T. Sullivan
e-mail: tommi.sullivan@griffithuni.edu.au

A.J. Abotomey
e-mail: andrew.abotomey@griffithuni.edu.au

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_48

of time. Time of arrival (TOA) uses the round-trip time (RTT) to calculate the distance of a transmitter from a receiver. This requires clocks with a very high degree of accuracy and precision as a difference of 1.0 μs results in a difference of 300 m [1]. Time difference of arrival (TDOA) is similar to TOA but calculates the difference between the arrival times of a transmitted signal. This requires only the synchronisation between the receivers. In an AOA approach, a receiver can calculate its position through measuring the direction of the incoming signals of transmitters with known position. This allows for the device to use triangulation but requires an antenna array [2].

These localisation techniques have been used with various devices including light detection and ranging (LiDAR), GPS, and radar. However, wireless technologies such as Wi-fi and Bluetooth are becoming more prevalent in everyday use, and increasing in numbers rapidly [3]. Unfortunately, these technologies are often unable to rely on the parameters required for the localisation methods stated above [4]. With the widespread adoption of wireless technologies, such as Wi-fi (IEEE 802.11) and Bluetooth (IEEE 802.15), these technologies provide a unique solution where the many available nodes may be utilised where traditional global positioning systems (GPSs) are unavailable or impractical. This paper extends the research by Sullivan et al. [5] to investigate the differences and similarities between Wi-fi and Bluetooth and their practical applications for localisation through RSS.

2 The Wireless Technologies for This Research

As both Wi-fi and Bluetooth are low cost and have been adopted by many household devices, this study will compare which of these two technologies is more effective for indoor localisation. Both technologies are standardised by the IEEE with Wi-fi under the 802.11 set of protocols and Bluetooth under the 802.15 set of protocols [6]. While the two operate within the 2.4 Ghz band, the differentiating factor between the two is the broadcasting power and the power consumption. WizFi210 and Mini Beacon were chosen as transmission nodes for this investigation.

2.1 WizFi210

At full transmission power, the Wi-fi shield transmits at +8 dBm. Due to the active nature of Wi-fi, it is expected that it would use more power than the Mini Beacon. Furthermore, increased output power will enable the Wi-fi connection to have improved signal quality for data transfer. The specifications are outlined below in Table 1.

Table 1 WizFi210 specifications [7]

Specifications	Description
Radio protocol	IEEE 802.11b/g/n
RF frequency	2.4 Ghz (2.497 Ghz)
Power consumption	Standby: 34.0 μA, Receive: 125.0 mA, Transmit: 135.0 mA
RF output power	8dBm ± 1dBm
Power source	3.3v
Dimensions (excluding antenna)	59 x 54 mm
Range indoors	20 m
Range outdoors	>20 m
Supported data rates	11, 5.5, 2, 1 Mbps

Table 2 Mini Beacon specifications [8]

Specifications	Description
Radio protocol	IEEE 802.15b/g/n
RF frequency	2.4 Ghz (2.400–2.483 Ghz)
Power consumption	50 μA
RF output power	0 dBm ± 1dBm
Power source	2.0–3.6 v
Dimensions (excluding antenna)	36 × 16 mm
Range outdoors	50–90 m

2.2 Mini Beacon

As opposed to Wi-fi, Bluetooth low-energy (BLE) beacons are designed to broadcast a transmission at regular intervals. This technology can be used for limited data transfer; however, for this research, this will be used for investigating signal propagation. This specific BLE beacon is unable to respond to ping requests and is only able to broadcast. The specifications for the Mini Beacon are outlined below in Table 2.

3 Path Loss Model

As a signal propagates through space, it experiences a loss in signal strength. Path loss (*PL*) is a measure of the average attenuation of this loss in signal strength. This is defined as in Eq. 1 [9].

$$PL(\mathrm{dB}) = 10\log\frac{P_t}{P_r} \tag{1}$$

where P_t and P_r are the transmitted power and received power, respectively. In free space, Pr can be expressed using the Friis free-space equation [10]:

$$P_r = (d)\frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{2}$$

where G_t and G_r are the transmission gain and received gain by their respective antennas. L is the system losses that are not associated with the propagation such as losses at the antenna. An L value of 1 indicates that there are no losses by the system. λ is the wavelength of the carrier in metres.

As this model cannot account for a d (distance) value of 0, many propagation models use a reference point (d_0) for a close-in distance [11], usually at 1 m. The general path loss model uses γ to reflect the noise exponent of a signal propagation. This model is defined as in Eq. 3 and is used to model the propagation in this investigation.

$$PL(d) = PL(d_0) + 10\gamma\log\left(\frac{d}{d_0}\right) + X_\sigma \tag{3}$$

Free space is represented using a γ value of 2 where as a high noise environment could have a value of 4. X_σ denotes the random variable with a standard deviation of σ. This is used to model shadowing but will be given a value of 0 for this investigation.

4 Data Analysis

In order to ensure clean data, the experiment was set up with an Arduino unit with the WizFi210 module in order to have a Wi-fi node running with the least possible system noise. The Mini Beacon was deployed as is. These signals were read by a notebook computer with an Edimax USB 802.11n Wi-fi adapter and an Orico Bluetooth adapter. Both of these adapters were attached to 3 m USB 2.0 leads in order to minimise the noise generated by the notebook computer itself. The experiment itself was conducted outdoors on flat terrain with no obstructions to line of sight between the transmitter and receiver. This environment was chosen to minimise environmental noise as there were no other electronic devices or above ground cables. Readings for both devices were collected over 2 days in mid-October with no precipitation and a stable temperature of 24° Centigrade for both days. The receiver and transmitter modules were raised 1 m from ground level with the transmitter being placed 0.5 m away from the receiver and moved away by

0.5 m increments for each reading. To ensure that the two transmitting devices would not interfere with each other's signals, the WizFi210 and Mini Beacon were deployed and tested separately.

4.1 WizFi210 Analysis

Figure 1 plots all the readings from the Wi-fi device with the darker dots indicating more occurrences of a specific RSSI at a given distance. What should be noted is manner in which the Wi-fi readings loosely follow a logarithmic curve despite the small spread in data readings. This suggests that there is a consistent and significant amount of noise in the signal itself.

Using the median RSSI from each set of readings (i.e. every 50 cm), the γ value for each distance was calculated from the average noise exponent across all distances. This figure, calculated to 4.1854, confirms the previous hypothesis of high noise. The model in Fig. 2 can be derived from using the median noise exponent and the median RSSI at 1 m. Furthermore, the graph also illustrates the confidence interval (CI) for all the readings which indicates the reliability of the data produced. The narrow CI range indicates that the readings are closely clustered, as stated previously.

The high noise exponent is likely to be attributed to the system noise of the hardware itself.

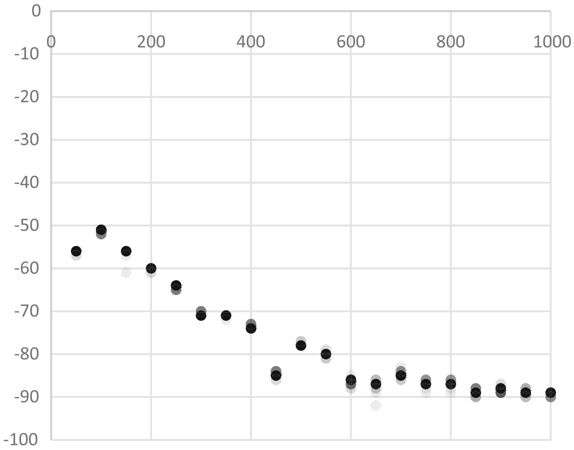


Fig. 1 Wi-fi readings

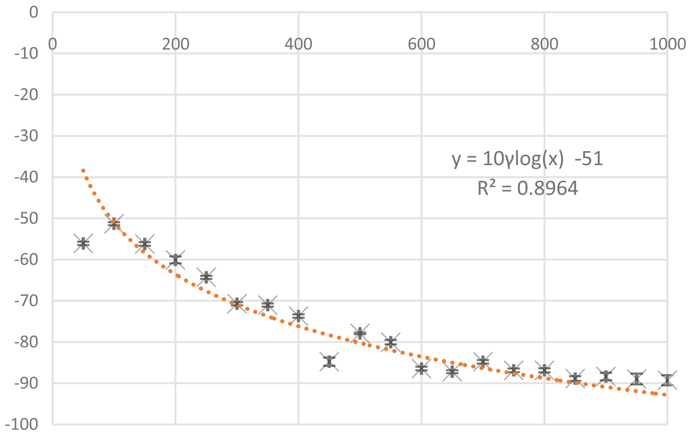


Fig. 2 Dispersion graph of Wi-fi

4.2 Mini Beacon Analysis

Figure 3 illustrates the data collected from the Bluetooth tests. While the data suggest a number of outliers, there is a relatively high concentration around a logarithmic trend.

With an average value of 2.56825, the test can be seen to have very little noise affecting its reading. Using this γ value, the relationship can be modelled as shown in Fig. 4.

Fig. 3 Bluetooth readings

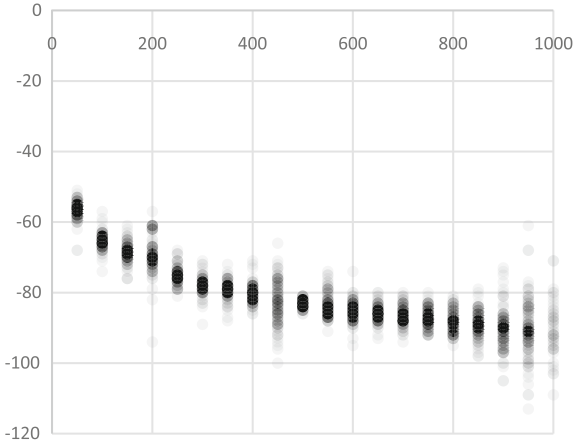
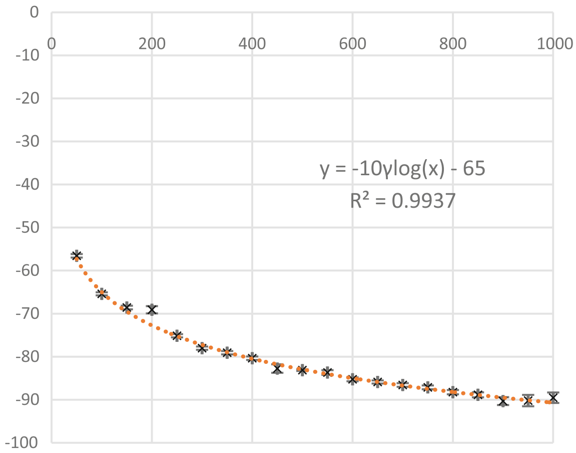


Fig. 4 Dispersion graph of Bluetooth



4.3 Path Loss

Both the Wi-fi and the Bluetooth devices follow the path loss model but with varying degrees of noise. It can be seen that from the median noise exponents are contained within Table 3, the Wi-fi device experiences a greater amount of noise to the Bluetooth device.

The noise exponent of the model allows for any estimations to take into considerations the noise from the system itself and its surrounding environment. It does not take into consideration the spread of the data itself which is defined as X_σ Eq. 3. This can be seen by comparing the actual distances to the estimated distances produced by the two models. This is done by using the median RSSI reading at each distance for the two models. Table 4 illustrates the accuracy of the two models. It can be seen that the estimations for Mini Beacon was almost consistently more accurate with an average of 4.61 % error as opposed to the 20.16 % for the WizFi210. It should also be noted that both are accurate to within 100 cm for readings up to 800 cm. The variance in the RSSI readings further than this point may result in a higher degree of error.

Table 3 Noise exponent for Wi-fi and Bluetooth

Device	Noise exponent
Wi-fi	4.1854
Bluetooth	2.5682

Table 4 RSSI to distance error rates for Wi-fi and Bluetooth

Actual distance (cm)	Wi-fi error	Bluetooth error
50	+81.66 cm (163.33 %)	–5.38 cm (10.75 %)
100	0 cm (0 %)	0 cm (0 %)
150	–18.34 cm (12.22 %)	–6.86 cm (4.58 %)
200	–35.93 cm (17.96 %)	–43.44 cm (21.72 %)
250	–45.54 cm (18.22 %)	–4.89 cm (1.95 %)
300	+0.5 cm (0.17 %)	+20.76 cm (6.92 %)
350	–49.5 cm (14.14 %)	+0.85 cm (0.24 %)
400	–45.57 cm (11.39 %)	–16.24 cm (4.06 %)
450	+199.15 cm (44.26 %)	+52.19 cm (11.6 %)
500	–58.33 cm (11.67 %)	+2.19 cm (0.44 %)
550	–56.96 cm (10.36 %)	–0.71 cm (0.13 %)
600	+85.87 cm (14.31 %)	+0.81 cm (0.14 %)
650	+74.66 cm (11.49 %)	+7.17 cm (1.1 %)
700	–50.85 cm (7.26 %)	+18.81 cm (2.69 %)
750	–25.34 cm (3.38 %)	–31.19 cm (4.16 %)
800	–75.34 cm (9.42 %)	–13.77 cm (1.72 %)
850	–41.06 cm (4.83 %)	+9.98 cm (1.17 %)
900	–134.36 cm (14.93 %)	+40.64 cm (4.52 %)
950	–141.06 cm (14.85 %)	+78.87 cm (8.3 %)
1000	–191.06 cm (19.11 %)	–59.36 cm (5.94 %)
Average	–26.37 cm (20.16 %)	+2.52 cm (4.61 %)

5 Conclusion

This study investigated the practical usability of both Wi-fi and Bluetooth for localisation purposes. While Wi-fi is capable of higher rates of data transmission, results indicate that Bluetooth is a more reliable source for local positioning by an average of 15.55 %. While the two operate with many similar technologies and on the same 2.4 Ghz bandwidth, the lower error rates for the Bluetooth beacons, along with its smaller hardware footprint, suggest that the hardware of the device itself contributes significantly to a signal's noise. When considering these factors and its low power consumption, Bluetooth is a more practical solution for localisation.

References

1. Jami, I., Ali, M., Ormondroyd, R.F.: Comparison of methods of locating and tracking cellular mobiles. In: IEE Colloquium on Novel Methods of Locating and Trackin of Cellular Mobiles and Their System Applications. London (1999)

2. Peng, R., Sichitiu, M.L.: Angle of arrival localization for wireless sensor networks. Sensor Ad Hoc Commun. Networks **1**(1), 374–382 (2006)
3. Wireless Broadband Alliance: Global Developments in Public WiFi. WBA Industry Report, Singapore (2011)
4. Hernandez-Perez, E., Navarro-Mesa, J.L., Martin-Gonzalez, S., Quintana-Morales, P., Ravelo-Garcia, A.: Path loss factor estimation for RSS-based localiztion algorithms with wireless sensor networks. In: 19th European Signal Processing Conference, Barcelona (2011)
5. Sullivan, T., Jo, J., Lennon, M.: A usability study for signal strength based localisation. Robot Intell. Technol. Appl. 3 **345**, 35–44 (2015)
6. Bahl, P., Padmanabhan, V.N.: RADAR: an in-building RF-based user location and tracking system. In: INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (2000)
7. WIZnet: WizFi210/220 User Manual, 6 May 2013. http://www.wiznet.co.kr/wp-content/uploads/wiznethome/WiFi%20Module/WizFi_210_220/Document/WizFi210_DS_V120E.pdf
8. Mini Beacon, Mini Beacon User Manual, Mini Beacon, 2014
9. Bertoni, H.L.: Radio Propagation for Modern Wireless Systems. Prentice Hall Professional Technical Reference, Upper Saddle (2000)
10. Andersen, J.B., Rappaport, T.S., Yoshida, S.: Propagation measurements and models for wireless communication channels. IEEE Commun. Mag. **33**(1), 42–49 (1995)
11. Tapan, S.K., Zhong, J., Kyungjung, K., Adbellatif, M., Magdalena, S.-P.: A survey of various propagation models for mobile communication. IEEE Antennas Propag. Mag. **45**(3), 51–82 (2003)

Cloud-Based Robots and Intelligent Space Teleoperation Tools

**Tomáš Cádrik, Peter Takáč, Jaroslav Ondo, Peter Sinčák,
Marián Mach, František Jakab, Filippo Cavallo
and Manuele Bonaccorsi**

Abstract Despite an idea of robotic system teleoperation is a relatively old concept, here we present its enhancements heading to an interconnection of teleoperation and collecting relevant information from the environment where robots act. This environment should be an intelligent space featured with various devices and sensors, which allows to obtain, preprocess and stores data in the cloud. Those data should provide relevant information for teleoperator or directly for robots, which act autonomously. For this purpose, we developed cloud-based tools, named Telescope v2. It is a platform-independent system for remote monitoring and controlling various systems. In this paper, we introduce this system, its abilities, and compare it with its network-based ancestor, Telescope v1. We analyze measure-

T. Cádrik · P. Takáč · J. Ondo · P. Sinčák (✉) · M. Mach · F. Jakab
Center for Intelligent Technologies, Department of Cybernetics and Artificial Intelligence,
Faculty of Electrical Engineering and Informatics, Technical University of Kosice, Košice,
Slovakia
e-mail: peter.sincak@tuke.sk
URL: http://www.ai-cit.sk

T. Cádrik
e-mail: tomas.cadrik@tuke.sk

P. Takáč
e-mail: peter.takac.3@tuke.sk

J. Ondo
e-mail: jaroslav.ondo@tuke.sk

M. Mach
e-mail: marian.mach@tuke.sk

F. Jakab
e-mail: frantisek.jakab@tuke.sk

F. Cavallo · M. Bonaccorsi
The BioRobotics Institute, Scuola Superiore Sant'Anna, Pontedera, Pisa, Italy
e-mail: f.cavallo@sssup.it

M. Bonaccorsi
e-mail: m.bonaccorsi@sssup.it

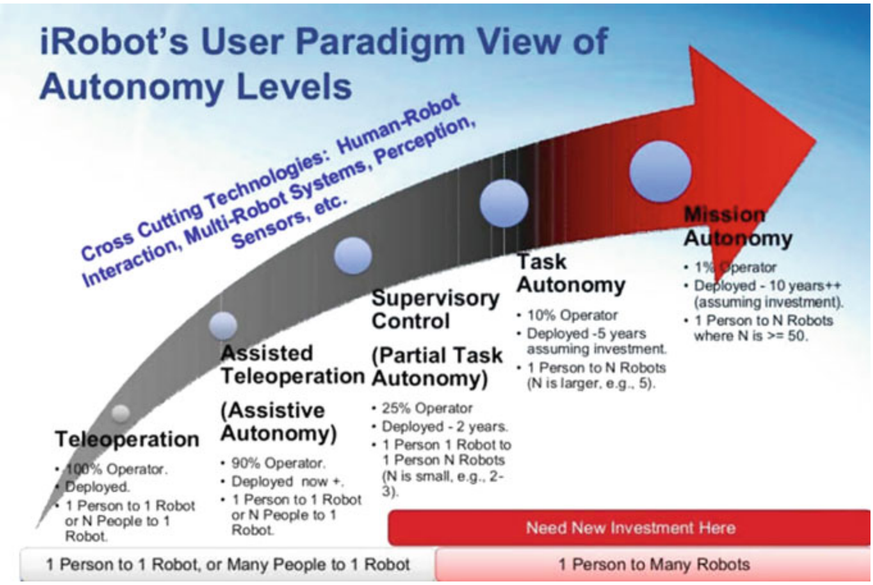
J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 4*,
Advances in Intelligent Systems and Computing 447,
DOI 10.1007/978-3-319-31293-4_49

ments of latency and response time when our new system is used for teleoperation in different places equipped with various Internet bandwidths.

1 Introduction

Teleoperation is an important factor of today's service and social robots. The basic aim of teleoperation is being able to operate a robot in remote or non-approachable environment. The human person operating a remote robot is using all the information to be able to control the robot by all the means of technology. The latest results in cloud computing are giving more and more conditions to fulfill the idea of remote brain concept [1]. The practical aim is during teleoperating a robotic system "would be nice to learn" and offload the human which is teleoperating a robot from more routine work and this is possible by creating intelligent agent on the cloud environment which is doing gradual replacement of human which is operating a robot toward anonymous robots. The Fig. 1 shows the concept of the US IROBOT Company of creating an intelligent machine using teleoperation and engaging learning procedure into the process. The application potential of this approach is rather large, since a number of autonomous machines are expected to appear on the market in the near future.

One of the major problems related to intelligent machines are the difficulties in measuring the autonomy of the system. There is no universal approach to this



iRobot's User Paradigm View of Autonomy Levels

Cross Cutting Technologies: Human-Robot Interaction, Multi-Robot Systems, Perception, Sensors, etc.

Teleoperation
- 100% Operator.
- Deployed.
- 1 Person to 1 Robot or N People to 1 Robot.

Assisted Teleoperation (Assistive Autonomy)
- 90% Operator.
- Deployed now +.
- 1 Person to 1 Robot or N People to 1 Robot.

Supervisory Control (Partial Task Autonomy)
- 25% Operator
- Deployed - 2 years.
- 1 Person 1 Robot to 1 Person N Robots (N is small, e.g., 2-3).

Task Autonomy
- 10% Operator
- Deployed -5 years assuming investment.
- 1 Person to N Robots (N is larger, e.g., 5).

Mission Autonomy
- 1% Operator
- Deployed - 10 years++ (assuming investment).
- 1 Person to N Robots where N is >= 50.

Need New Investment Here

1 Person to 1 Robot, or Many People to 1 Robot

1 Person to Many Robots

Fig. 1 The approach of teleoperation toward mission autonomy operation [2]

problem. We can state that it is a task- or mission-oriented approach, and therefore, we write it as follows:

$$\mathbf{GTI = HTI + MTI} \tag{1}$$

where

GTI (global task intelligence) is always value 1 and is a sum of human intelligence "HTI (human task intelligence) from interval $\langle 0,\ 1\rangle$ and machine intelligence MTI (machine task intelligence) from interval $\langle 0,\ 1\rangle$. Also, we can define a machine task intelligence autonomity (MTIA)" as follows:

$$\mathbf{MTIA = MTI/HTI} \tag{2}$$

So, when MTIA is 0, we are describing a manual, fully human-made process, since HTI is 1 and MTI is 0. If MTIA is a very large number, HTI is very small close to "0" and MTI is close to "1." This can be considered as an autonomous mission where a human is the only observer. The further consideration of machine intelligent quotients (MIQ) related to machines have been studied in the past [3]. The MIQ should be domain-oriented and could be used in future for commercial advantage by selling various machines to humans.

The goal of teleoperation is to allow a human to control a robot in a situation where it is inconvenient or unsafe to place a human and difficult to program a robot to perform autonomously complex operations [4]. Nowadays, teleoperation tasks variate from very basic like a toy car teleoperation by remote controller, teleoperation of firefighter drones equipped with cameras, up to high-tech teleoperation task in the field of medicine, army, or space exploration. Many of those tasks demand the robot to perform more complex and difficult works [4].

With the development of high-speed Internet networks, it has formed a network-based robot teleoperation with a combination of the advantages of network technology and robot technology [5]. We used just this technology to build the Telescope v1 system.

Along with the development of high-speed Internet networks, the cloud services have become popular. Cloud technology has brought a whole new concept of software programming, operation, and maintenance. Cloud providers guarantee a worldwide 24/7 availability, offer autoscaling of deployed services in the case that demand exceeds limits of service and the last but not least on-demand computational and storage sources [6, 7]. On the other hand, it requires to such applications be deployed somewhere on the Internet, more precisely in cloud providers data centers. The cloud computing has become popular also in research. An evidence of that is the emergence of a new approach, named cloud robotics, which James Kuffner defined in 2010 in the paper [8].

This new approach inspired us to the development of a cloud-based system for robot teleoperation. Moreover, we enhanced typical teleoperation systems with the possibility of operation different robotic systems and also with a possibility of acquiring relevant information not only from robots but also from the environment, where the robot acts. As was mentioned; this environment should be an intelligent space. By intelligent space, we mean a room equipped with many cameras and sensors, which enable the space to perceive and understand what is happening in them [9]. Such space also can provide additional information for the operator and also directly for robots.

The paper structure is as follows. Chapter "Trajectory Generation Using RNN with Context Information for Mobile Robots" briefly describes Telescope v1 system, as an ancestor of cloud-based Telescope v2 system. Then, the attention is focusing on Telescope v2 description, which consists of robot teleoperation module and intelligent space devices control module. We compare old and new system and highlight the advantages of each solution. In Chap. "Accurate Localization in Urban Environments Using Fault Detection of GPS and Multi-sensor Fusion", we focus on experiments and their evaluations.

2 Server-Based Teleoperation Tool—Telescope V1

Telescope v1 was developed as a server-based modular system, which allows operating connected devices through the Internet all over the world. Simply, it allows teleoperation of connected devices. In our understanding, we consider a device as any electronic appliance that has an ability to communicate over the network and can be accessed programmatically (using API).

We designed the system for various independent operators, who can teleoperate their connected devices at the same time. We made a special effort to implement Telescope v1 to be platform-independent and so solve a problem how to connect and operate devices with heterogeneous environments. By heterogeneous environment, we mean devices with different interfaces for programmers (API), programmable with various inconsistent programming languages and feature with diverse user's interfaces. We solved that using a wrapper program (similar to a device driver), which translates Telescope's commands into the commands understandable for the appropriate device. Such wrappers were implemented for each type of connected device.

Figure 2 is shown as an architecture of Telescope v1 system. It consists of two integrated modules—Teleoperation module and Telediagnostics module, which provides a full set of information about the state of the operated device and its

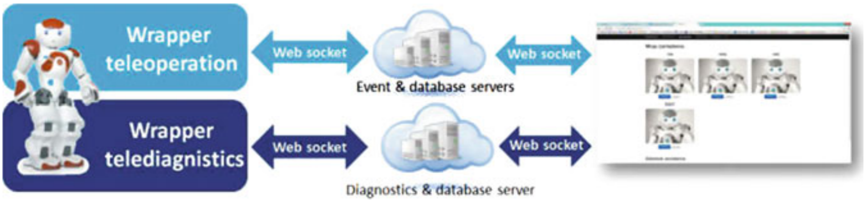


Fig. 2 The basic architecture of the telescope v1 system

possible faults or errors for the operator. It is necessary for successful completion of the task remotely.

The essential part of the system is Event server. This server interconnects messages coming from the appropriate operator and device(s), which the operator teleoperates. Web server hosts Telescope and Telediagnostics Web portals and provides controls and overview of connected devices for operators [10]. Each device can connect and communicate with Event server only in the case that there is installed wrapper program. Simultaneously, we implemented Telediagnostics server, which provides a service for acquiring and storing data of healthy state, faults, and errors of connected devices. By comparing up-to-date data and stored historical trends, we can identify arisen problem of the device. For all ongoing communication, the Web socket [10] technology is used.

The test bed for the Telescope v1 was a Nao robot, which is a complex robotic system, consists of various sensors and actuators. The Telescope v1 allows to operate almost all functions of Nao robot, including single-joints movements; stiffness settings; text to speech; and launching preprogrammed behaviors, such as sitting and walking. Moreover, the system allows adding, removing, and sharing devices with others users, what makes possible to operate one robot by various operators, or operate several robots by one operator.

3 Toward to Cloud-Based Teleoperation—Telescope V2

Here, we describe more deeply a cloud-based solution for teleoperation, named Telescope v2. This system evolved from Telescope v1 and enhanced its functionality by gathering relevant information from devices installed in the intelligent space. It allows to gain supporting information for robot teleoperation. Since the Telescope v2 consists of robot teleoperation module and a module for data acquiring from intelligent space, this section is divided into two subsections, which describe each module separately.

3.1 Robot Teleoperation Module

Telescope v2 is a platform for multiple-robot-type teleoperation, which is based on the Microsoft Azure cloud platform. This system has a big advantage according to his predecessor, which used only a server architecture. The advantage when using the cloud is you do not need to care about the physical machine. Also, the scalability can serve for the processing of many requests of the users. If many users want to connect to the system, we can simply increase the computational capabilities of the machine.

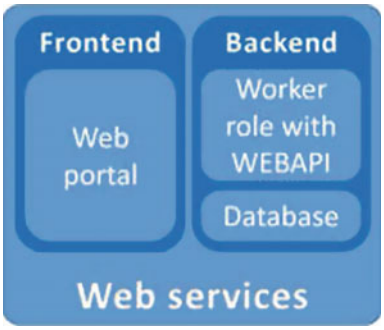


Fig. 3 The basic structure of the cloud-based telescope v2 system

The system consists of two main parts: One is for the interaction with the user and the second one is for the whole communication and calculation logic. This architecture is shown in Fig. 3.

The front end is a simple Web page that can be used by the user for teleoperation. The user can use the Web page for adding a new type of robot, or adding a movement for the robot. For this, frontend was the MVC.ASP.Net used [11] which can be simply explain as a combination of the C# programming language and HTML.

The worker role serves for starting and controlling the WEBAPI [10]. Because WEBAPI usually runs on the frontend size of the cloud service, we needed to find a way how to start it on the worker role. The result was the OWIN self-host library which serves exactly for our purpose. The WEBAPI is used for the communication between the robots and the service. It is a technology, which allows sending and receiving messages via http protocol. That means that you can use an Http client to call the specific function in our WEBAPI controller using a simple url. In this, system is the WEBAPI controller used for sending a request when a user clicks on the specific movement and for sending movements when a robot requests for a movement.

The last part of the system is the databases. At the beginning, only two of them are created. The first one is the table with the list of added robots, and the second one is the blob container (for storing raw data) for the files with the description of an added movement. When the user adds a new robot new table and blob container is created. This table is then used for storing the movements of the robots.

The movement is represented by a python script. This python script is specific for each robot. For instance, the movement for the NAO robot has code using naoqi and the robots based on the robotic operating system are using subprocess module for calling the rostopic commands.

When we need to connect a robot to a cloud service, we cannot do it directly. It is possible with using VPN, but we wanted to solve this without it. So we created a wrapper which is universal for all Linux-based robots. This wrapper is able to communicate with the WEBAPI service and can also save the time response between the robot and the cloud service.

The paring mechanism in this system is very simple. The user adds a unique string on the Web site and when he chooses the movement, the combination between the type of the robot with the unique number and the selected movement is saved to a dictionary. And when the wrapper sends a request to the cloud service, it also sends the type of the robot and the unique number of this robot. If the dictionary contains a movement for this concrete robot, it will send to it.

There are three possible scenarios how to use this cloud service. First one is that one user can control one robot. The second one is that multiple users can control one robot. At one time, the robot can do only one action, but we can change it in the future. The last way how the system can be used is that one user can control multiple robots which can move at the same time.

WEBAPI was also chosen because it is easy to create another interface which is using the cloud service. The application can simply use http calls to the WEBAPI via URL. So the developer can create an application which uses for instance some sensors for controlling the robots.

3.2 Module for Acquiring Data from the Intelligent Space

While the teleoperation schemes of Telescope v1 and Telescope v2 can handle the user robot communication and control, it can collect data only acquired by the robot itself. In case of an indoor environment, the robot would have to rescan the given environment each time it makes an action. This approach is time and also resource-consuming. Hence, we have used an approach gathering preprocessed data from intelligent space with the ability of storing the relevant data directly in the cloud environment.

The proposed solution relies on the ability of Microsoft Azure cloud storage to store multiple types of data, which can be accessed efficiently in real-time. Furthermore, the Microsoft Azure cloud is used as the main platform for the user device communication, ensures the commands routing, keeps track of processes, and to some extent ensures data gathering.

The system consists of three main parts (see Fig. 4). The first part serves as the frontend for the interaction with the users, the second part is the main cloud–client communication module, and the third part is the client application in the desired local environment, where the devices are placed.

For the frontend, the ASP.NET MVC 5 [11] was used, which offers the user control capabilities in the form of graphical interface, where all the clients and devices are shown. This frontend was placed on Azure Web role and is interconnected with the backend by WEBAPI technology [12], which offers making simple method-like calls from one Role to another in real-time.

The second part is the backend which is represented by Azure worker role, which holds the WEBAPI controller for frontend to backend communication and the SignalR Hub, which further communicates with the client application. The SignalR technology [13] is based on Web socket communication where the

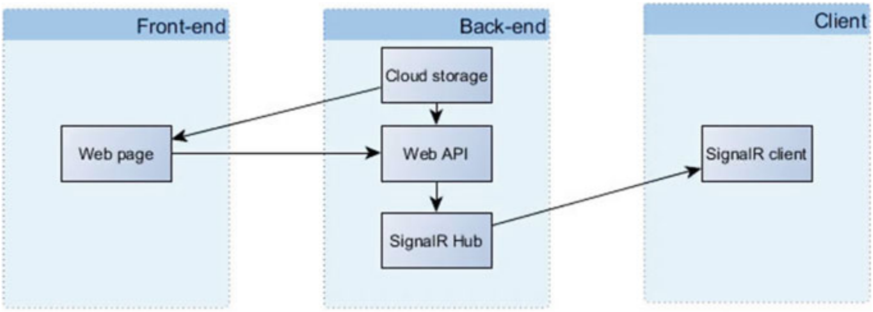


Fig. 4 The basic architecture of the teleoperation in intelligent space

server-side logic (Hub) is responsible for connection registration and in/out connection management, whereas the client-side logic is only able to receive incoming connection and call methods from the Hub. For more complex mechanism for returning messages from the client, a number of modifications had to be made. These modifications mainly influenced the Hub, where several methods were created for the client, which can this way notify the Hub about any relevant information or even send data back.

Together, five buffers were created: Client buffer, Device buffer, Process buffer, Data buffer and Device-related data buffer. The Client and the Device buffers are in the form of simple dictionary, which hold the data about the connected client applications and about the devices connected to them. The Process buffer is the most crucial part of the system, while it holds the information about all the running processes, i.e., running data gathering. It ensures that only one type of action is performed on a single device, while there can be multiple accesses to the given client. The last two buffers were created especially for data upload, which can by mainly used for sending data to other roles for further processing. The reason behind why there are two buffers is to ensure simplicity and flexibility within the Hub, while multiple types of devices could be feeding data to the Hub and this buffer stores them in a joined queue and then distributes them to their appropriate containers within the Device-related data buffer, where the exact types can be stored.

The third and the last part is the client application, which runs on Windows machine to which the devices in the environment are connected. It is a simple console application where the SignalR client is running. The client is able to communicate with devices and forward the user input from the cloud to the device the way it would be controlled locally using device controllers. For each type of devices, there is a unique controller native for the communication with the devices firmware. Moreover, this client is also responsible for its registration and also the registration of the devices connected to it by uploading information about itself and the devices to the Azure cloud table, which serves as a simple database. In some cases, it is important to upload only raw data to the cloud with no preprocessing needed, and in this case, the client application is also able to upload the gathered data directly to the Azure cloud storage.

For being able to control, the devices from the cloud at least one client with a controllable device have to be connected to it and running. The user can then easily select the desired client with the device and perform action with this device. The input from the user is then forwarded to the process buffer, where it is noted that a certain process started after the device has started performing the given action and only after then the user is notified by the system about the action success. Usage of the Web socket technology and the request sizes ensures that this whole action is only matter of milliseconds.

4 Experiments

We have done preliminary experiments focused on latency and response time measurements. Specifically, we measured reaction time in the teleoperation module and latency in the module for acquiring data from the intelligent space.

We focused on the experiments that show whether the used technologies (WEBAPI and SignalR) are a good choice for this purpose. In the first case, we tested the WEBAPI service that is the main component in the robot teleoperation module. The object of those experiments was to measure the time between the movement when the operator entered the request on the teleoperation Web site and the moment when the robot started to act. The times were saved in local log files in the robot, and then, they were sent to the cloud storage.

Till now, we measured a reaction time during the teleoperation of the Nao, Qbo, and Hanson robots. The operator and the robots were located in Košice, Slovakia, but the teleoperation was done through the cloud service. The results of the experiments are shown in Fig. 5.

The experiments show that the fastest connection was used with the Qbo robot. The most unstable connection was using the Nao robot. When we do not count the measurements that are untraditional, the average was between 86 and 259 ms. The most stable connection was on the Qbo robot. For now, we do not know whether the connection deviations are because of the WEBAPI technology or the device. More measurements need to be done to analyze this.

In the case of experiments of teleoperation in intelligent space, the measurements were based on the time that took to send the request to the cloud and receive the response. This measurement is the key to be able to tell whether this system is usable in real-time applications. The experiment was carried out as follows:

1. The SignalR client created a request, which consisted of simple binary data containing a simple string message notificating the SignalR Hub about the available information, i.e., the name of the client, client’s connection ID, and the connected devices information.

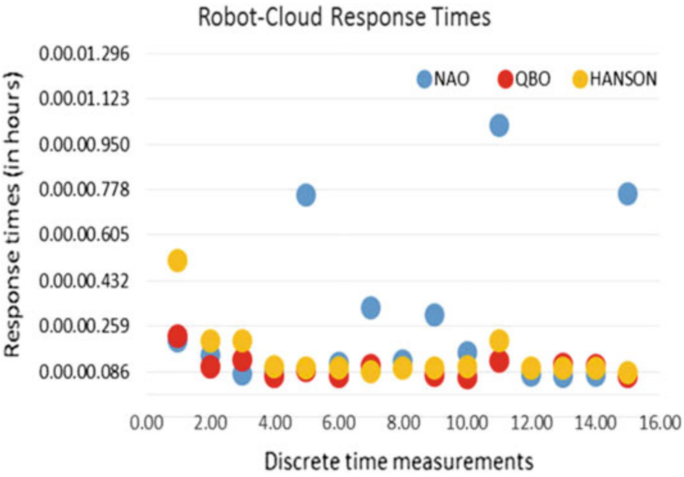


Fig. 5 The graph shows the results of robot–cloud response times from the latency measurements

2. The SingalR Hub receives the information, checks whether it is in a right format, and if there are no duplicate requests, it proceeds to respond back to the user.
3. The SignalR client receives the boolean information and if the connection was successful (true value of the received boolean variable), it stores the final time measurement.

During our experiments, there was no failure in sending data in the proper format and while there was also no conflict, we managed to get the time measurement of

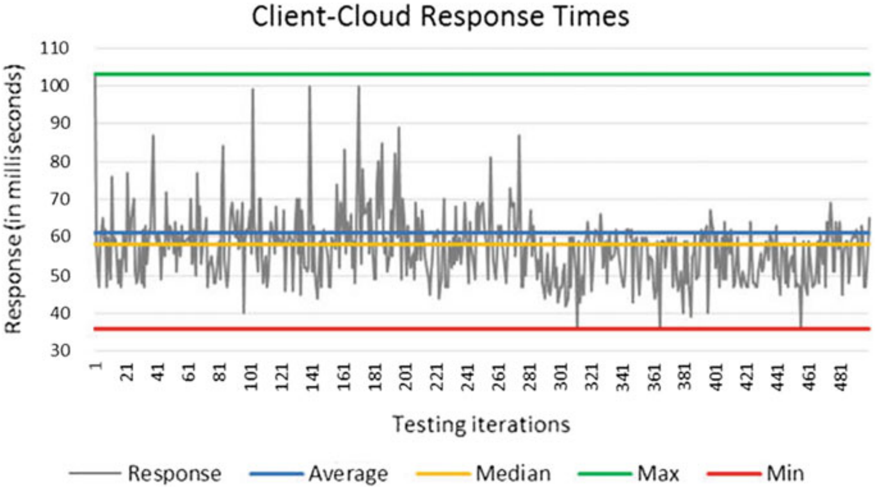


Fig. 6 The graph shows the results of client–cloud response times from the latency measurements

every request made by the SignalR client. The results of the experiment are shown in Fig. 6. We can see that our system managed to get the response from the cloud on average 63 ms, which is a very good result, if we consider that the basic latency measurements of our network were around 15 ms. The best latency measurement was 36 ms and the worst measurement was 103 ms. These data leave sufficient time window for further processing of the gathered data and also for data larger in size to be uploaded by this system.

5 Conclusion

Taking into consideration that all experiments were performed locally, where the operator and also the robot were in the same place, although the control was done via the cloud service. We cannot claim that the used methods are the right ones for teleoperation purpose. The results of our experiments were good enough, but we need to approve our hypothesis based on much more measurements. This way, we would like to ask volunteers to perform tests for us with our system, which is available on [14]. We need to collect teleoperation data generated during the robot teleoperation from different places all around the world, connected to the Internet using various bandwidths and by several Internet providers. Then, we will able to analyze acquired data, upgrade our system, and conclude the results. If someone is interested to participate on this, please contact us using the email address peter.takac.3@tuke.sk.

6 Acknowledgment

This publication is the result of the following projects implementation: University Science Park TECHNICOM for Innovation Applications Supported by Knowledge Technology, ITMS: 26220220182, partially supported by the National Research and Development Project Grant 1/0773/16 2016 – 2019 “Cloud Based Artificial Intelligence for Intelligent Robotics”, and also partially supported by Young researchers university grant FEI-2015-13 “System for robots teleoperation in the intelligent space.”

References

1. Inaba, M., et al.: A platform for robotic research based on the remote-brained robot approach. Int. J. Robot. Res. **19**(10), 933–954 (2000)
2. iRobot’s user paradigm view of autonomy levels. Robotics Summit, Virtual Conference and Expo, June 2011

3. Park, H., Kim, B., Kye L.: Measuring the machine intelligence quotient (MIQ) of human-machine cooperative systems. IEEE Trans. SMC/Part A Syst. Humans **31**(2) (2011)
4. Hu, H.H.H., Li, J.L.J., Xie, Z., Wang, X.Z., Bin, B.W., Liu, H.L.H., Hirzinger, G.: A robot arm/hand teleoperation system with telepresence and shared control. In: Song, W., Zhao, S., Jiang, F., Zhu (eds.) Proceedings of 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 24–28 (2005)
5. Zhu, K., Cao, L., Shi, Y. (2012). Teleoperation robot control system based on mindband sensor. In: 2012 Fifth International Symposium on Computational Intelligence and Design, pp. 299–302
6. NIST: The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. Nist Special Publication, vol. 145, p. 7 (2011)
7. Lorenčík, D., Cádrik, T., Mach, M., Sinčák, P.: Cloudová robotika; Vplyv cloudového computingu na budúcnosť robotiky. ATP J. **3**(1), 40–42 (2014)
8. Kuffner, J.J.: Cloud-enabled robots. In: IEEE-RAS International Conference on Humanoid Robotics (2010)
9. Lee, J.-H., Hashimoto, H.: Intelligent space—concept and contents. Adv. Robot. **16**(3), 265–280 (2002)
10. Miženko, L., Lorenčík, D., Ondo, J., Sinčák, P.: Telescope: System Overview. In: Sobota, B., Balász, S. (eds.) Developments in virtual reality laboratory for factory of the future, pp. 100–105. Bay Zoltán Nonprofit Ltd, Miskolc (2014)
11. ASP.NET MVC. http://www.asp.net/mvc
12. ASP.NET WebAPI. http://www.asp.net/web-api
13. ASP.NET SignalR. http://www.asp.net/signalr
14. Telescope v2 website. http://wizardofoz.cloudapp.net/