Lecture Notes in Electrical Engineering 85

Juan Andrade Cetto
Joaquim Filipe
Jean-Louis Ferrier
Editors

Informatics in Control Automation and Robotics

Revised and Selected Papers from the International Conference on Informatics in Control Automation and Robotics 2009

Springer

Lecture Notes in Electrical Engineering

Volume 85

Juan Andrade Cetto, Joaquim Filipe ·
Jean-Louis Ferrier (Eds.)

Informatics in Control Automation and Robotics

Revised and Selected Papers from the International Conference on Informatics in Control Automation and Robotics 2009

Springer

Dr. Juan Andrade Cetto
Institut de Robòtica i Informàtica Industrial
Consejo Superior de Investigaciones Científicas / Universidad
Politècnica de Catalunya
Llorens i Artigas, 4-6
Edifici U
08028 Barcelona
Spain
E-mail: cetto@iri.upc.edu

Prof. Joaquim Filipe
Polytechnic Institute of Setúbal / INSTICC
2910-595 Setubal
Portugal
E-mail: jfilipe@insticc.org

Prof. Jean-Louis Ferrier
Institut des Sciences et Techniques de l'Ingénieur d'Angers (ISTIA)
Labo. d'Ingénierie des Systémes Automatisés (LISA)
62 avenue Notre Dame du Lac
49000 Angers
France
E-mail: ferrier@istia.univ-angers.fr

ISBN 978-3-642-19729-1 e-ISBN 978-3-642-19730-7

DOI 10.1007/978-3-642-19730-7

Library of Congress Control Number: 2011923552

Typeset & Coverdesign: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

The present book includes a set of selected papers from the Sixth International Conference on Informatics in Control Automation and Robotics (ICINCO 2009), held in –Milan, Italy, from 2 to 5 July 2009. The conference was organized in three simultaneous tracks: Intelligent Control Systems and Optimization, Robotics and Automation, and Systems Modeling, Signal Processing and Control. The book is based on the same structure.

ICINCO 2009 received 365 paper submissions, from 55 different countries in all continents. From these, after a blind review process, only 34 were accepted as full papers, of which 21 were selected for inclusion in this book, based on the classifications provided by the Program Committee. The selected papers reflect the interdisciplinary nature of the conference. The diversity of topics is an important feature of this conference, enabling an overall perception of several important scientific and technological trends. These high quality standards will be maintained and reinforced at ICINCO 2010, to be held in Funchal, Madeira - Portugal, and in future editions of this conference.

Furthermore, ICINCO 2009 included 5 plenary keynote lectures given by Daniel S. Yeung (University of Technology, China), Maria P. Fanti (Polytechnic of Bari, Italy), Janan Zaytoon (University of Reims Champagne Ardennes, France), Alessandro Giua (Universita' di Cagliari, Italy) and Peter S. Sapaty (Institute of Mathematical Machines and Systems, National Academy of Sciences Ukraine). We would like to express our appreciation to all of them and in particular to those who took the time to contribute with a paper to this book.

On behalf of the conference organizing committee, we would like to thank all participants. First of all to the authors, whose quality work is the essence of the conference and to the members of the Program Committee, who helped us with their expertise and diligence in reviewing the papers. As we all know, producing a conference requires the effort of many individuals. We wish to thank also all the members of our organizing committee, whose work and commitment were invaluable.

July 2010

Juan Andrade-Cetto
Jean-Louis Ferrier
Joaquim Filipe

Conference Committee

Conference Chair

Joaquim Filipe	Polytechnic Institute of Setúbal/INSTICC, Portugal

Program Co-chairs

Juan Andrade Cetto	Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Spain
Jean-Louis Ferrier	University of Angers, France

Organizing Committee

Sérgio Brissos	INSTICC, Portugal
Marina Carvalho	INSTICC, Portugal
Helder Coelhas	INSTICC, Portugal
Vera Coelho	INSTICC, Portugal
Andreia Costa	INSTICC, Portugal
Bruno Encarnação	INSTICC, Portugal
Bárbara Lima	INSTICC, Portugal
Raquel Martins	INSTICC, Portugal
Elton Mendes	INSTICC, Portugal
Carla Mota	INSTICC, Portugal
Vitor Pedrosa	INSTICC, Portugal
José Varela	INSTICC, Portugal
Pedro Varela	INSTICC, Portugal

Program Committee

Arvin Agah, USA
Alessandro Chiuso, Italy
Hyo-Sung Ahn, Korea, Republic of
Frank Allgower, Germany
Francesco Amigoni, Italy
Plamen Angelov, UK
Helder Araújo, Portugal
Marco Antonio Arteaga, Mexico
Vijanth Sagayan Asirvadam, Malaysia
T. Asokan, India
Robert Babuska, The Netherlands
Ruth Bars, Hungary
Adil Baykasoglu, Turkey
Maren Bennewitz, Germany

Karsten Berns, Germany
Arijit Bhattacharya, Ireland
Sergio Bittanti, Italy
Stjepan Bogdan, Croatia
Jean-louis Boimond, France
Djamel Bouchaffra, USA
Bernard Brogliato, France
Edmund Burke, UK
Clifford Burrows,UK
Dídac Busquets, Spain
Giuseppe Carbone, Italy
J. L. Martins de Carvalho, Portugal
Alessandro Casavola, Italy
Riccardo Cassinis, Italy
Chien Chern Cheah, Singapore
Tongwen Chen, Canada
Wen-Hua Chen, UK
Graziano Chesi, China
Carlos Coello Coello, Mexico
Yechiel Crispin, USA
Michael A. Demetriou, USA
Guilherme DeSouza, USA
Jorge Dias, Portugal
Gamini Dissanayake, Australia
Denis Dochain, Belgium
Tony Dodd, UK
Alexandre Dolgui, France
Prabu Dorairaj, India
Marco Dorigo, Belgium
Venky Dubey, UK
Petr Ekel, Brazil
Andries Engelbrecht, South Africa
Sebastian Engell, Germany
Simon G. Fabri, Malta
Sergej Fatikow, Germany
Jean-marc Faure, France
Paolo Fiorini, Italy
Juan J. Flores, Mexico
Georg Frey, Germany
Manel Frigola, Spain
John Qiang Gan, UK
Nicholas Gans, USA
Leonardo Garrido, Mexico
Andrea Garulli, Italy
Lazea Gheorghe, Romania
Paulo Gil, Portugal
Alessandro Giua, Italy
Luis Gomes, Portugal
Dongbing Gu, UK
Guoxiang Gu, USA
Jason Gu, Canada
Wail Gueaieb, Canada
José J. Guerrero, Spain
Thomas Gustafsson, Sweden
Maki K. Habib, Japan
Hani Hagras, UK
Wolfgang Halang, Germany
Riad Hammoud, USA
Uwe D. Hanebeck, Germany
Robert Harrison, UK
Dominik Henrich, Germany
Francisco Herrera, Spain
Victor Hinostroza, Mexico
Wladyslaw Homenda, Poland
Guoqiang Hu, USA
Marc Van Hulle, Belgium
Fumiya Iida, USA
Atsushi Imiya, Japan
Hisao Ishibuchi, Japan
Thira Jearsiripongkul, Thailand
Dimitrios Karras, Greece
Dusko Katic, Serbia
Graham Kendall, UK
Tamas Keviczky, The Netherlands
Jonghwa Kim, Germany
Won-jong Kim, USA
Waree Kongprawechnon, Thailand
Israel Koren, USA
George L. Kovács, Hungary
H. K. Lam, UK
Kemal Leblebicio, Turkey
Graham Leedham, Australia
Kauko Leiviskä, Finland
Kang Li, UK
Tsai-Yen Li, Taiwan
Yangmin Li, China
Huei-Yung Lin, Taiwan
Zongli Lin, USA
Jing-Sin Liu, Taiwan
Jose Tenreiro Machado, Portugal
Frederic Maire, Australia
Om Malik, Canada

Jacek Mandziuk, Poland
Hervé Marchand, France
Gerard Mckee, UK
Seán McLoone, Ireland
Carlo Menon, Canada
Sanya Mitaim, Thailand
Pieter Mosterman, USA
Rafael Muñoz-salinas, Spain
Kenneth Muske, USA
Andreas Nearchou, Greece
Luciana P. Nedel, Brazil
Sergiu Nedevschi, Romania
Anton Nijholt, The Netherlands
Hendrik Nijmeijer, The Netherlands
Juan A. Nolazco-Flores, Mexico
Urbano Nunes, Portugal
José Valente de Oliveira, Portugal
Romeo Ortega, France
Manuel Ortigueira, Portugal
Selahattin Ozcelik, USA
Christos Panayiotou, Cyprus
Stefano Panzieri, Italy
Igor Paromtchik, France
D. T. Pham, UK
Marie-Noëlle Pons, France
Jerzy Respondek, Poland
A. Fernando Ribeiro, Portugal
Robert Richardson, UK
Rodney Roberts, USA
Juha Röning, Finland
António Ruano, Portugal
Fariba Sadri, UK
Carlos Sagüés, Spain
Mehmet Sahinkaya, UK
Antonio Sala, Spain
Abdel-badeeh Salem, Egypt
Mitsuji Sampei, Japan
Medha Sarkar, USA
Jurek Sasiadek, Canada
Daniel Sbarbaro, Chile
Carla Seatzu, Italy
João Sequeira, Portugal
Michael Short, UK
Silvio Simani, Italy
Dan Simon, USA
Adam Slowik, Poland
Michael Small, Hong Kong
Burkhard Stadlmann, Austria
Tarasiewicz Stanislaw, Canada
Karl Stol, New Zealand
Olaf Stursberg, Germany
Chun-Yi Su, Canada
Cornel Sultan, USA
Ryszard Tadeusiewicz, Poland
Choon Yik Tang, USA
Daniel Thalmann, Switzerland
N. G. Tsagarakis, Italy
Antonios Tsourdos, UK
Nikos Tsourveloudis, Greece
Anthony Tzes, Greece
Dariusz Ucinski, Poland
Nicolas Kemper Valverde, Mexico
Eloisa Vargiu, Italy
Laurent Vercouter, France
Bernardo Wagner, Germany
Axel Walthelm, Germany
Dianhui Wang, Australia
Qing-Guo Wang, Singapore
Zidong Wang, UK
James Whidborne, UK
Dirk Wollherr, Germany
Marek Zaremba, Canada
Janan Zaytoon, France
Qin Zhang, USA

Auxiliary Reviewers

Andrea Baccara, Italy
Rui Cortesao, Portugal
Matteo de Felice, Italy
Andrea Gasparri, Italy
Zhi Han, USA
Vitor Jorge, Brazil
Enrico di Lello, Italy
Andrea Monastero, Italy
Federico di Palma, Italy
Maura Pasquotti, Italy

Katalin Popovici, France
Monica Reggiani, Italy
Maurizio di Rocco, Italy
P. Lopes dos Santos Santos, Portugal
Thomas Tometzki, Germany
Ali Emre Turgut, Belgium
Maja Varga, Croatia
Kai Wurm, Germany

Invited Speakers

Daniel S. Yeung	University of Technology, China
Maria Pia Fanti	Polytechnic of Bari, Italy
Janan Zaytoon	University of Reims Champagne Ardennes, France
Alessandro Giua	University of Cagliari, Italy
Peter S. Sapaty	Institute of Mathematical Machines and Systems, National Academy of Sciences, Ukraine

Contents

Part II: Robotics and Automation

Part III: Signal Processing, Systems Modeling and Control

Invited Papers

ICT Application on the Management of Intelligent Transportation Systems

Maria Pia Fanti

Department of Electrical and Electronic Engineering
Polytechnic of Bari, 70125, Bari, Italy
fanti@deemail.poliba.it

Abstract. Intelligent Transportation Systems (ITS) modelling, planning, and control are research streams that, in the last years, have received a significant attention by the researcher and practitioner communities due not only to their economic impact, but also to the complexity of decisional, organizational, and management problems. Indeed, the increasing complexity of these systems and the availability of the modern Information and Communication Technologies (ICT) for the interaction among the different decision makers and for the acquisition of information by the decision makers, require both the development of suitable models and the solution of new decision problems. The paper is aimed at showing the new attractive researches and projects in the fields of ITS operational control and management by using effectively and efficiently the latest key ICT solutions. In particular, the paper focuses on the ICT applications on the urban traffic control and the intermodal transportation network planning and management.

Keywords: Intelligent transportation systems, Urban traffic control, Intermodal transportation networks, Information and communication technologies.

1 Introduction

Intelligent Transportation Systems (ITS) refer to the application of Information and Communication Technologies (ICT) to transport infrastructure and / or to vehicles to improve the efficiency of transportation networks. In recent years, the European Union has sponsored several projects targeting advancements of different transportation systems. On the other hand, ITS topics are considered relevant and attractive research areas. In particular, research and projects on the ITS may be broadly divided into *infrastructure projects* and *vehicle-orientated applications*. Typical infrastructure projects and researches include the planning of freight and people transportation operations and the traffic management. Vehicle-orientated projects and researches include applications such as automated vehicle location and planning. This paper focuses on the infrastructure planning and management that apply ICT tools. The advent of ICT has a tremendous impact on the planning of freight and people transportation and on traffic management. Indeed, ICT increase the flow of available data, improve the timeliness and quality of information and offer the

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 3–13.
springerlink.com

possibility to control and coordinate operations and traffic in real-time. Significant research efforts are required to adequately model ITS, specify and characterize the various planning and management problems and to develop efficient solution methods based of the application of the ICT tools. The aim of the paper is presenting the last contributions in term of projects and research in the fields of the planning and management of ITS. Hence, the paper recalls in Section 2 the most important projects supported by the European Commission (EC) in the areas of ITS. Moreover, Sections 3 and 4 present the research advances in two crucial sectors of ITS: the control of Urban Traffic Networks (UTN) and the management of Intermodal Transportation Networks (ITN), respectively. In the two cases the paper enlightens the benefits of the ICT solution applications and the relative open problems. Finally, Sections 5 draws the conclusions.

2 European Projects in the Field of ITS

The potential of ITS applications has encouraged the EC to refer to them in its 2006 mid-term review of the transport white paper. In particular, the grant program for the Trans-European Network for Transport (TEN-T) [35] aims to establish interconnections, interoperability and continuity of services especially on long-distance routes and across borders. Its guidelines cover the infrastructure for traffic management, traveller information, emergency systems and electronic fee collection. Within the framework of the TEN-T program for the period 2007-2013 the EasyWay project [35] is selected. Such a project is driven by national road authorities and operators with partners including the automotive industry, telecom operators and public transport stakeholders. TEN-T program identifies the set of necessary ITS European services to deploy (traveller information, traffic management and freight and logistic services) and is an efficient platform that allows the European mobility stakeholders to achieve a coordinated and combined deployment of these pan-European services.

In the last years, the EC selected several projects that focus in turn on the planning and management of motorway, railway, and maritime traffic [16]. In particular, a basic project on ITS is CESAR I & II (Co-operative European System for Advanced Information Redistribution) that proposes an Internet communication platform that aims to integrate services and data for unaccompanied traffic and the rolling motorway traffic management. Moreover, in the field of railway system management, CroBIT (Cross Border Information Technology) is a new system that provides the railways with a tool to track consignments and integrates freight railways along a transport corridor providing total shipment visibility. A maritime navigation information structure in European waters is established by MarNIS (Maritime Navigation Information Systems) that is an integrated project aiming to develop tools that can be used to exchange maritime navigation information and to improve safety, security and efficiency of maritime traffic.

In addition, several projects focus specifically on efficient freight transportation. For instance, Freightwise aims to establish a framework for efficient co-modal freight transport on the Norwegian ARKTRANS system. One of the main objectives in Freightwise is establishing a framework for efficient co-modal freight transport and simplifying the interaction among stakeholders during planning, execution and completion of transport operations. Moreover, the project e-Freight is a continuation

of Freightwise to promote efficient and simplified solutions in support of cooperation, interoperability and consistency in the European transport system. E-Freight is to support the freight logistics action plan, which focuses on quality and efficiency for the movement of goods, as well as on ensuring that freight-related information travels easily among modes. Furthermore, in the 7th Framework Program (FP7), the SMARTFREIGHT project wants to make urban freight transport more efficient, environmentally friendly and safe by answering to challenges related to traffic management and the relative coordination. Indeed, freight distribution management in city centres is usually operated by several commercial companies and there is no coordination of these activities in a way that would benefit the city. The main aim of SMARTFREIGHT is therefore to specify, implement and evaluate ICT solutions that integrate urban traffic management systems with the management of freight and logistics in urban areas. In addition, EURIDICE (European Inter-Disciplinary Research on Intelligent Cargo for Efficient, Safe and Environment-friendly Logistics) is a project sponsored by the EC under the FP7 seeking to develop an advanced European logistics system around the concept of 'intelligent cargo'. The goal is networking cargo objects like packages, vehicles and containers to provide information services whenever required along the transport chain. The project aims to build an information service platform centered on the individual cargo item and its interaction with the surrounding environment and the user.

Moreover, a set of European projects focuses on the people transportation management. For instance the CESARE project (Common Electronic Fee Collection System for a Road Tolling European Service) started more than ten years ago and its development was structured into four project phases that are mainly focused on studying the technical aspects of the European interoperability. In particular, the CESARE IVth and final phase is specifying, designing, developing, promoting and implementing a common interoperability of electronic road toll systems in the European community, in order to function in a coordinated way. Furthermore, in the project called eMOTION public authorities, transport and telecommunication operators, IT suppliers and transport consultancies specify and assess multi-modal, on-trip traffic and travel information services for the European travellers. In particular, the range of services comprises real time traffic information for road traffic and public transport, dynamic (and multimodal) routing services and special services like reservation of parking space, booking of personalized public transport services as well as tourist information or hotel reservation.

Summing up, the EC recognizes that innovation and ITS help to make transport more sustainable, which means efficient, clean, safe and seamless. In other word, new infrastructure cannot solve all transport problems of congestion and emissions and the integration of existing technologies can create new services. Consequently, for the next years the EC prioritizes projects involving application of ICT on transportation systems, in close collaboration with national governments, to ensure effective European coordination.

3 Intelligent Transportation in Urban Traffic Control

Traffic congestion of urban roads undermines mobility in major cities. Traditionally, the congestion problem on surface streets was dealt by adding more lanes and new

links to the existing Urban Traffic Networks (UTN). Since such a solution can no longer be considered for limited availability of space in urban centres, greater emphasis is nowadays placed on traffic management through the implementation and operation of ICT. In particular, traffic signal control on surface street networks plays a central role in traffic management. Despite the large research efforts on the topic, the problem of urban intersection congestion remains an open issue [10],[11],[23],[27]. Most of the currently implemented traffic control systems may be grouped into two principal classes [28]: i) fixed time strategies, that are derived off-line by use of optimization codes based on historical traffic data; ii) vehicle actuated strategies, that perform an on-line optimization and synchronization of the signal timing plans and make use of real time measurements. While the fixed time strategies do not use information on the actual traffic situation, the second actuated control class can be viewed as a traffic-responsive control policy employing signal timing plans that respond automatically to traffic conditions. In a real time control strategy, detectors located on the intersection approaches monitor traffic conditions and feed information on the actual system state to the real time controller, which selects the duration of the phases in the signal timing plan in order to optimize an objective function. Although the corresponding optimal control problem may readily be formulated, its real time solution and realization in a control loop has to face several difficulties such as the size and the combinatorial nature of the optimization problem, the measurements of traffic conditions and the presence of unpredictable disturbances. The first and most notable vehicle actuated techniques are the British SCOOT [20] and the Australian SCATS [24], that decides an incremental change of splits, offsets and cycle times based on real time measurements. However, although SCOOT and SCATS exhibit centralized hardware architecture, the strategies are functionally decentralized with regard to splits setting. A formulation of the traffic signal network optimization strategy is presented in [23] and [30]. However, the resulting procedures lead to complex mixed integer linear programming problems that are computationally intensive and the formulation for real networks requires heuristics for solutions. Indeed, the most available signal control strategies are not suitable for saturated traffic conditions because they fail to consider the downstream traffic conditions in the real-time decision making at individual junctions. Due to this reason, there is a lack of efficient and systematic co-ordinated control strategies applicable to large scale network. In order to solve such a problem, some contributions are proposed based on the so-called store- and-forward modelling approach [19] that describes the network traffic flow process so as to circumvent the inclusion of binary variables. This choice permits the use of efficient optimization and control methods. In particular, we recall the design approach of Diakaki *et al.* [9] that propose a traffic responsive co-ordinated urban control strategy formulized as an optimal control problem based on a store-and-forward modelling approach. More precisely, the urban traffic control problem is formulated as a linear-quadratic (LQ) optimal control problem leading to a highly efficient and extremely simple co-ordinated strategy applicable to large-scale networks. Moreover, in [1] the authors compare the methodology based on the linear multivariable feedback regulator derived through the formulation of the LQ optimal control problem with two novel methodologies based on the store-and-forward modelling paradigm. The authors highlight the advantages and the shortcomings of the different strategies.

Based on the same model, a recent paper [8] presents a prototype of computational platform to control an urban traffic network. The platform automatically implements a distributed Model Predictive Control (MPC) from the mathematical specification of the problem and runs a distributed algorithm. The platform prototype is applied in the green-time control of a real-world network and the distributed algorithm is compared with the LQ approach application from [1]. The simulation results show that the MPC-based approach outperform the LQ-based approach in term of traffic performance metrics. Moreover, the advantage of the distributed model predictive control approach is the easiness of expansion and reconfiguration of the control structure.

Despite the efficiency of the proposed control strategies, such a modelling approach can not directly consider the effects of offset for consecutive junctions and the time-variance of the turning rates and the saturation flows. An improvement on urban traffic actuated control strategies is provided in [13] where the green splits for a fixed cycle time are determined in real time, in order to minimize the number of vehicles in queue in the considered signalized area. To this aim, the paper pursues simplicity in the modelling and in the optimization procedure by presenting a macroscopic model to describe the urban traffic network. Describing the system by a discrete time model with the sampling time equal to the cycle, the timing plan is obtained on the basis of the real traffic knowledge and the traffic measurements in a prefixed set of cycles. The traffic urban control strategy is performed by solving a mathematical programming problem that minimizes the number of vehicles in the considered urban area. The minimization of the objective function is subject to linear constraints derived from the intersection topology, the fixed cycle duration and the minimum and maximum duration of the phases commonly adopted in practice.

Despite the large literature in the field of the urban traffic control, the future research has to give new contributions in facing the *apparently insurmountable difficulties* [28] in the real time solution and realization of the control loop governing an urban intersection by traffic lights. Future work will be devoted to provide methodologies enjoying the following properties: i) guaranteeing a real time network wide signal control in large scale urban traffic; ii) exhibiting an efficient control behaviour in different traffic scenarios, such as saturated traffic conditions; iii) developing additional algorithms for considering different real components such as bus-priorities and pedestrians; iv) optimizing the effects of offset for consecutive junctions. Finally, in order to validate the proposed methodologies, field applications to extended network urban parts are necessary.

4 Intelligent Transportation in Intermodal Transportation Networks

Intermodal Transportation Networks (ITN) are transportation systems integrating different transportation services, designed to move goods from origin to destination in a timely manner and using multiple modes of transportation [4]. The ITN management and planning are currently relevant subjects of research because ITN allow more efficient, cost effective and sustainable transportation than the traditional transportation systems [6]. However, ITN decision making is a very complex process,

due to the dynamical and large scale nature of these systems, the hierarchical structure of decisions, the multiplicity of actors involved, as well as the randomness of various inputs and operations. A systematic way to capture all decisions in the management of ITN typically proposed in the related literature is based on a three-level hierarchy: strategic, tactical and operational ones. Strategic level planning involves ITN design and considers time horizons of a few years, requiring approximate and aggregate data. Tactical level planning basically refers to the optimization of the flow of goods and services through a given logistics network. Operational level management is a short-range planning, involving transportation scheduling of all transporters on an hour-to-hour basis, subject to the changing market conditions as well as to unforeseen transportation requests and accidents.

With the development of ICT, operative and tactical issues can be dealt in a different way than in the past, taking advantage of the effective impact of these innovative technologies on ITN decision making. Indeed, ICT solutions can increase the data flow and the information quality while allowing real-time data exchange in intelligent transportation systems and traffic networks [6], [13]. As discussed by Giannopoulos [20], numerous novel applications of ICT to the transportation field are currently in various stages of development, but in the information transfer area the new systems seem to be unimodal. In the application of ICT solutions to multimodal chains, an important and largely unexplored research field is the assessment of the impact of the new ICT tools on management and control of ITS at the operational level before their implementation, by a cost-benefit analysis [38]. In particular, an effective ITN model at the operational level should focus on evaluating performance indices describing activities, resources (cost, utilization and inventory), output (throughput, lead time) and flexibility (lead-time, lead time variability) by integrating information flows.

In the related literature, simulation has represented an effective and useful instrument to analyze transport logistics and evaluate the impact of the proposed solutions [5], [34] [37]. However, ITN and their decision making processes exhibit a high degree of concurrency and are characterized by resource sharing and conflicts. Hence, ITN can be successfully modelled as Discrete Event Systems (DES), whose dynamics depends on the interaction of discrete events, such as demands, departures and arrivals of means of transportation at terminals and acquisitions and releases of resources by vehicles. DES models are widely used to describe decision making and operational processes in logistics systems. In particular, Ramstedt and Woxenius [32] analyze the literature about the simulation of the decision-making process within a transportation chain and Gambardella et al. [18] simulate a resource allocation problem in an intermodal container terminal. In a subsequent work, Rizzoli et al. [33] present a discrete-event simulation model of the flow of intermodal terminal units among and within inland intermodal terminals, serving a user area via a road network and interconnected by rail corridors. Moreover, Legato and Mazza [22] propose a queuing network model of the logistic activities related to the basic processes of vessels in a container terminal. Also in [29] a discrete event simulation model is used to analyze in an intermodal container environment the impact of new road and railway networks on the logistics system. Among the available DES models, Petri Nets (PN) may be selected as a graphical and mathematical technique, suitable to describe

concurrency and synchronization [31]. Indeed, processes in intermodal terminals are addressed in [17] with stochastic PN and in [7], [12], [14] with Timed Petri Nets (TPN).

Since the cited models are designed to describe a particular ITN, it is evident in the related literature the lack of a systematic and general methodology to describe in detail the multiplicity of elements that can influence the ITN dynamics and the corresponding information and decision making structure. An effort to propose a methodology able to thoroughly and systematically model a generic ITN is performed by Dotoli et al. [14] that use a modular modeling approach in a TPN framework to describe and simulate the ITN behavior. Moreover, in [14] the application of ICT solutions that allow sharing information among stakeholders in an intermodal system is analyzed and some performance indices are singled out to evaluate activities, resources (utilization) and output (throughput, lead time). A case study is modelled and simulated considering an ITN constituted by a port and a truck terminal of an Italian town including the road-ship transhipment process. The simulation results show that ICT have a huge potential for efficient real time management and operation of ITN, as well as an effective impact on the infrastructures, reducing both the utilization of the system resources as well as the cost performance indices.

In addition, paper [2] is a contribution to propose not only a systematic modelling approach devoted to describe a generic ITN, but also a management technique based on a closed loop mode. To this aim, papers [2] develops an Integrated System (IS) to be used by decision makers that have to take operational as well as tactical decisions in large and complex ITN and may rely on information based services. The proposed management and planning approach is based on the specification of two main modules that are the core of the IS: an ITN reference model and a simulation module. In particular, the reference model describes systematically and in detail a generic ITN, providing the system states to the simulation module that foresees the evolution of the ITN. In order to obtain a generic model describing a non-specific ITN, the paper proposes a metamodelling technique that applies to models. Moreover, the metamodelling approach provides an accurate description of the constructs and rules needed to obtain semantic models that encapsulate all the concepts necessary to describe the structure and the behaviour of a particular system. More in detail, the considered metamodelling approach is a top down procedure based on the Unified Modelling Language (UML) [25]: a graphic and textual modelling language intended to understand and describe systems from various viewpoints. In addition, the simulation module is specified by identifying an instance of the metamodel and by easily translating such UML formalism in the simulation software. Hence, the simulator reproduces and foresees the ITN behaviour and provides the Decision Making System (DMS) with suitable performance indices in order to evaluate the proposed decision strategies.

The presented IS can work in two alternative ways, respectively devoted to the off-line planning (tactical level) and the on-line management (operational level) of the ITN. Figure 1 shows the IS structure for the tactical level decisions: the IS bases on the reference model analysis and simulation the detection of the system anomalies and bottlenecks. Successively, the IS proposes, tests and evaluates some solutions on the basis of the estimation of suitable performance indices.

Fig. 1. The IS structure for the operational level application

Fig. 2. The IS structure at the tactical level application

Figure 2 depicts the IS structure for operational decisions. In this case, the IS works in real time and the reference model is updated on the basis of data exchanged by the stakeholders and the information obtained by using modern ICT tools. Moreover, the simulation module reproduces and foresees the behaviour of the ITN, starting from the model state knowledge and the detected real events. Hence, the reference model supplies the decision making system with the knowledge base necessary for decisions in real time and for optimizing suitable performance indices. In order to show the effectiveness of the presented management approach, the paper [3] specifies and analyzes at the tactical level for a real ITN composed of the port of Trieste (Italy) and the inland terminal located in Gorizia (Italy). The case study is analyzed in the framework of the EURIDICE Integrated Project, sponsored by the EC under the PF7 [26] and the performances obtained by the simulation studies show how integrating ICT into the system leads to a more efficient system management and drastically reduces the system lead times.

The papers [2] and [3] are contributions devoted to propose a systematic modelling approach to describe a generic ITN as well as to specify decision making strategies. Future research will address the open problems of the detailed description of all the

ITN nodes such as airports and railway systems, and the specification and design of the Decision and Management System modules.

5 Conclusions

The paper presents some new attractive researches and projects in the field of ITS planning, control and management. In particular, the key solutions of using effectively and efficiently the latest developments of ICT for ITS management are pointed out. To this aim, the paper recalls the most important European Projects that are selected to make freight and people transportation more sustainable, efficient and safe by exploiting novel ICT solutions. Moreover, two crucial fields of the ITS management and control are considered: the control of Urban Traffic networks and the management of Intermodal Transportation Networks. In the two cases are emphasized the results and the advances of the ICT applications and the challenges of future researches.

References

1. Aboudolas, K., Papageorgiou, M., Kosmatopoulos, E.: Store-and-forward based methods for signal control problem in large-scale congested urban road networks. Transportation research, Part C 17, 163–174 (2009)
2. Boschian, V., Fanti, M.P., Iacobellis, G., Ukovich, W.: A Metamodeling Technique for Managing Intermodal Transportation Networks. In: Proc. of IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey (2010)
3. Boschian, V., Fanti, M.P., Iacobellis, G., Ukovich, W.: Using Information and Communication Technologies in Intermodal Freight Transportation: a Case Study. In: Proc. of IEEE International Conference on Systems, Man and Cybernetics, San Antonio, Texas, US (2009)
4. Caris, C., Macharis, G., Janssens, K.: Planning problems in intermodal freight transport: Accomplishments and prospects. Transportation Planning and Technology 31(3), 277–302 (2008)
5. Coronado, A.E., Coronado, E., Lalwani, C.S.: Wireless Vehicular Networks to Support Road Haulage and Port Operations in a Multimodal Logistics Environment. In: Proceedings of the IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics, July 22-24, pp. 62–67 (2009)
6. Crainic, T.G., Kim, K.H.: Intermodal transportation. In: Barnhart, C., Laporte, G. (eds.) Transportation, Handbooks in Operations Research and Management Science, vol. 14, pp. 467–537. North-Holland, Amsterdam (2007)
7. Degano, C., Di Febbraro, A.: Modelling automated material handling in intermodal terminals. In: Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1023–1028 (2001)
8. de Souza, F., Peccin, V.B., Camponogara, E.: Distributed Model Predictive Control Applied to Urban traffic Networks: Implementation, Experimentation, and Analysis. In: 6th IEEE Conference on Automation Science and Engineering, pp. 399–405 (2010)
9. Diakaki, C., Papageorgiou, M., Aboudolas, K.: A multivariable regulator approach to traffic-responsive network-wide signal control. Control Engineering Practice 10, 183–195 (2002)

10. Di Febbraro, A., Giglio, D., Sacco, N.: On applying Petri nets to determine optimal offsets for coordinated traffic light timings. In: Proc. 5th IEEE Int. Conf. on Intelligent Transportation Systems, Singapore, pp. 773–778 (2002)
11. Di Febbraro, A., Giglio, D., Sacco, N.: Urban traffic control structure based on hybrid Petri nets. IEEE Trans. on Intelligent Transportation Systems 5, 224–237 (2004)
12. Di Febbraro, A., Porta, G., Sacco, N.: A Petri net modelling approach of intermodal terminals based on Metrocargo system. In: Proc. Intelligent Transportation Systems Conf., pp. 1442–1447 (2006)
13. Dotoli, M., Fanti, M.P., Meloni, C.: A Signal Timing Plan Formulation for Urban Traffic Control. Control Engineering Practice 14(11), 1297–1311 (2006)
14. Dotoli, M., Fanti, M.P., Mangini, A.M., Stecco, G., Ukovich, W.: The Impact of ICT on Intermodal Transportation Systems: a Modelling Approach by Petri Nets. Control Engineering Practice 18, 893–903 (2010)
15. EasyWay project, http://www.easyway-its.eu/1/
16. European Commission, http://ec.europa.eu/transport/its/road/research_and_studies_en.htm
17. Fischer, M., Kemper, P.: Modeling and analysis of a freight terminal with stochastic Petri nets. In: Proc. 9th IFAC Symposium Control in Transportation Systems (2000)
18. Gambardella, L.M., Rizzoli, A.E., Zaffalon, M.: Simulation and planning of an intermodal container terminal. Simulation 71, 107–116 (1998)
19. Gazis, D.C.: Optimum control of a system of oversatured intersections. Operation Research 152, 302–320 (1964)
20. Giannopoulos, G.A.: The application of information and communication technologies in transport. European Journal of Operational Research 152, 302–320 (2004)
21. Hunt, P.B., Robertson, D.L., Beterton, R.D., Royle, M.C.: The SCOOT on-line traffic signal optimization technique. Traffic Engineering and Control 23, 190–199 (1982)
22. Legato, P., Mazza, R.M.: Berth planning and resources optimization at a container terminal via discrete event simulation. European Journal of Operational Research 133, 537–547 (2001)
23. Lo, H.K.: A cell-based traffic control formulation: strategies and benefits of dynamic timing plans. Transportation Science 35, 148–164 (2001)
24. Lowrie, P.R.: SCATS: the Sydney co-ordinated adaptive traffic system-principles, methodology, algorithms. In: Proceedings of the IEEE International Conference on Read Traffic Signalling, London, UK, pp. 67–70 (1982)
25. OMG: UML Resource Page (2009), http://www.uml.org/
26. Paganelli, P., Charalampos, V., Cornelisse, E., Damentka, A., Forcolin, M., Jermol, M., Styczynski, R., Szczurek, W., Vedovato, D.: Euridice Project White paper, http://www.euridice-project.eu/euridice_rep_new/files/PublicDocs/pub/White%20Papers/EURIDICE%20White%20Paper%20-%20Nov%2009.pdf
27. Papageorgiou, M.: Automatic control methods in traffic and transportation. In: Toint, P., Labbe, M., Tanczos, K., Laporte, G. (eds.) Operations Research and Decision Aid Methodologies in Traffic and Transportation Management, pp. 46–83. Springer, Heidelberg (1999)
28. Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., Wang, Y.: Review of road traffic control strategies. Proceedings of the IEEE 91, 2043–2067 (2003)
29. Parola, F., Sciomachen, A.: Intermodal container flows in a port system metwor: Analysis of possible growths via simulation models. International Journal of Production Economics 97, 75–88 (2005)

30. Patel, M., Ranganathan, N.: IDUTC: an intelligent decision-making system for urban traffic control applications. IEEE Trans. on Vehicular Technology 50, 816–829 (2001)
31. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice Hall, Englewood Cliffs (1981)
32. Ramstedt, L., Woxenius, J.: Modelling approaches to operational decision-making in freight transport chains. In: Proc. 18th NOFOMA Conf. (2006)
33. Rizzoli, A.E., Fornara, N., Gambardella, L.M.: A simulation tool for combined rail/road transport in intermodal terminals. Mathematics and Computers in Simulation 59, 57–71 (2002)
34. Shabayek, A.A., Yeung, W.W.: A simulation model for the Kwai Chung container terminals in Hong Kong. European Journal of Operational Research 140, 1–11 (2002)
35. Trans. European Transport Network, `http://tentea.ec.europa.eu/`
36. Wey, W.-M.: Model formulation and solution algorithm of traffic signal control in an urban network. Computers, Environment and Urban Systems 24, 355–377 (2000)
37. Yun, W.Y., Choi, Y.S.: A simulation model for container-terminal operation analysis using an object-oriented approach. International Journal of Production Economics 59, 221–230 (1999)
38. Zografos, K.G., Regan, A.: Current Challenges for Intermodal Freight Transport and Logistics in Europe and the US. Journal of the Transportation Research Board 1873, 70–78 (2004)

Abstractions of Hybrid Systems for Verification

Hervé Guéguen[1] and Janan Zaytoon[2]

[1] SUPELEC / IETR, Rennes, France
[2] Université de Reims Champagne Ardenne / CRESTIC, Reims, France
herve.gueguen@supelec.fr, janan.zaytoon@univ-reims.fr

Abstract. Safety verification for hybrid systems is a very important issue but also a very complex one. The abstraction is a mean to simplify it by shifting from global considerations to local ones. The aim of this paper is to present an overview of techniques of abstraction of hybrid systems that are commonly used for safety analysis. Reachability approaches related to abstraction for verification are briefly presented.

1 Introduction

Formal verification of properties is a very important area of analysis of hybrid dynamical systems. It is, indeed, essential to use methods and tools to guaranty that the global behaviour of the system is correct and consistent with the specifications. This is especially true for safety properties that insure that the system is not dangerous for itself or its environment as long as the assumptions on which the model is built are fulfilled [1].

Safety verification is a complex task and one approach to deal with it consists in defining a new model that is simpler to analyse in such a way that the conclusion on the simple model is guarantied to be valid for the initial model. As behaviour of systems is concerned and safety guaranty is the aim, this simplification is based on the concept of abstraction. Of course a part of the work load that is no longer required for the global analysis has to be spent on the abstraction process, but the considerations are then local and therefore easier to deal with.

The aim of this paper is to give an overview of the concept of abstraction for safety analysis and the related reachability approaches. In Sect. 2 the formal model of hybrid automaton is presented and the concept of abstraction is specified. Then two types of abstractions are considered. In the first one, presented in Sect. 3, the hybrid nature of the system is retained but its continuous dynamics is simplified by linear differential inclusions. The second one (Sect. 4) considers a purely discrete event system to abstract the system. As it will be seen in these sections, reachability is a central issue in the abstraction process and therefore some issues on reachability are finally presented in Sect. 5. A more detailed presentation of reachability may be found in [2].

2 Hybrid Systems and Hybrid Automata

2.1 Hybrid Automata

Hybrid systems are characterized by sequential and continuous dynamics that interact and, therefore, require specific modeling formalism to represent these phenomena. Such a formalism can be based on hybrid automata inspired from [3].

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 15–28.
springerlink.com

A hybrid automaton, without synchronising events is specified by the tuple $< L, X, U, Inv, F, E, Guard, Jump >$ where:

- L is the set of locations or discrete states, X is the continuous state space and U is the space of continuous inputs,
- Inv is a function that maps each location to a region of $X \times U$: $Inv(l_i)$ or, what will be considered as equivalent here, is a set of predicates on state and input variables that characterizes this region,
- F associates to each location l_i in L, the continuous dynamics $F(l_i, \mathbf{x}, \mathbf{u})$,
- E is a set of discrete transitions where each transition e is associated with a pair of locations (l, l')
- $guard$ is a function that associates to each transition a region in $X \times U$ or a set of predicates on state and input variables
- $Jump$ is a function that associates to each transition a map from the state and input spaces to the state space.

The behavior of such an automaton is specified by the set of admissible trajectories from an initial state. Each element, τ, of the set $\Theta\left((l_i, \mathbf{x}_i), \mathbf{u}(t)\right)$ of trajectories from a given state $(l_i, \mathbf{x}_i)$ subject to the input signal $\mathbf{u}(t)$, is a finite or infinite ordered set:

$$\tau = \{(l_{\tau_0}, 0, \Phi_0), (l_{\tau_1}, t_1, \Phi_1), \cdots, (l_{\tau_k}, t_k, \Phi_k), \cdots\}$$

such as

- $l_{\tau_0} = l_i$ and $\Phi_0(0) = \mathbf{x}_i$,
- for all indices k, $\forall t \in [t_k, t_{k+1}[$: the continuous state is $\mathbf{x}(t) = \Phi_k(t)$, the continuous dynamics is specified by the activity of the location $\dot{\Phi}_k(t) \in F(l_{\tau_k}, \mathbf{x}(t), \mathbf{u(t)})$ and the continuous state remains within the invariant of the location $(\Phi_k(t), \mathbf{u}(t)) \in Inv(l_{\tau_k})$
- for all indices k, there exists a transition $e = (l, l')$ in E such as: $l = l_{\tau_k}$, $l' = l_{\tau_{k+1}}$, the state before the firing is within the guard $(\Phi_k(t_{k+1}), \mathbf{u}(t_{k+1})) \in guard(e)$, and the initial state in the next location is the result of applying the jump function to the state before the firing $\Phi_{k+1}(t_{k+1}) = Jump(e)\ ((\Phi_k(t_{k+1}), \mathbf{u}\ (t_{k+1})))$.

For a given trajectory, each term $(l_{\tau_k}, t_k, \Phi_k)$ is then associated with a time interval $[t_k, t_{k+1}]$ during which the location does not change and the continuous state evolves according to the continuous dynamics associated with this active location l_{τ_k}, thus defining a continuous transition. Each time t_k is a discrete transition firing time when the location changes. A trajectory is then a sequence of discrete and continuous transitions.

For a trajectory τ of a hybrid automaton, the state at time t will be denoted $\tau(t)$ and is defined by $\tau(t) = (l_{\tau_k}, \Phi_k(t))$ where k corresponds to $t \in [t_k, t_{k+1}[$.

These definitions of a hybrid automaton and of the set of its trajectories from one hybrid state, leads to the following definitions of the successor sets of a point of the hybrid state $(l_i, \mathbf{x}_i)$.

The hybrid successor set of $(l_i, \mathbf{x}_i)$ is the set of states that are reachable by a trajectory starting in $(l_i, \mathbf{x}_i)$.

$$Succ_H((l_i, \mathbf{x}_i)) = \{(l_k, \mathbf{x}) | \exists \mathbf{u}(t)\ s.t.\ \exists \tau \in \Theta((l_i, \mathbf{x}_i), \mathbf{u}(t)),\ \exists t,\ \tau(t) = (l_k, \mathbf{x})\}$$

The discrete successor set of the point $(l_i, \mathbf{x}_i)$ is the set of points reachable by the firing of a discrete transition:

$$Succ_D((l_i, \mathbf{x}_i)) = \{(l_k, \mathbf{x}) | \exists e = (l_i, l_k) \in E,\ \exists \mathbf{u} \in U\ s.t. (\mathbf{x}_i, \mathbf{u}) \in Inv(l_i) \\ \wedge ((\mathbf{x}_i, \mathbf{u}) \in guard(e)) \wedge (\mathbf{x} = Jump(e)((\mathbf{x}_i, \mathbf{u}))\}$$

The continuous successor set of the point $(l_i, \mathbf{x}_i)$ is the set of points reachable by the firing of a continuous transition:

$$Succ_C((l_i, \mathbf{x}_i)) = \{(l_i, \mathbf{x}) | \exists \mathbf{u}(t)\ s.t.\ \exists \tau \in \Theta((l_i, \mathbf{x}_i), \mathbf{u}(t)), \exists t \in [0, t_1],\ \mathbf{x} = \Phi_0(t)\}$$

Symmetrically from these definition of successor sets it is possible to define the predecessor sets of $(l_i, \mathbf{x}_i)$ as the states from which it can be reached[1]:

$$Pred_\bullet((l_i, \mathbf{x}_i)) = \{(l, \mathbf{x}) | (l_i, \mathbf{x}_i) \in Succ_\bullet((l, \mathbf{x}))\}$$

These predecessor sets can be explicitely specified and for example the discrete predecessor set is given by:

$$Pred_D((l_i, \mathbf{x}_i)) = \{(l_k, \mathbf{x}) | \exists e = (l_k, l_i) \in E,\ \exists \mathbf{u} \in U\ s.t. (\mathbf{x}, \mathbf{u}) \in Inv(l_k) \\ \wedge ((\mathbf{x}, \mathbf{u}) \in guard(e)) \wedge (\mathbf{x}_i = Jump(e)((\mathbf{x}, \mathbf{u}))\}$$

These definitions are easily extended to regions of the hybrid state space considering that the image of the region is the union of the images of its points. For example, the continuous and discrete successor set of a region R are given by:

$$Succ_C(R) = \bigcup_{(l_i, \mathbf{x}) \in R} Succ_C((l_i, \mathbf{x})), \quad Succ_D(R) = \bigcup_{(l_i, \mathbf{x}) \in R} Succ_D((l_i, \mathbf{x}))$$

2.2 Abstraction

Abstraction is a concept that characterises the relationships between two transition systems. It is based on the existence of a relation from the state space of one system to the state space of the other one, such as the image of each trajectory of the first system is a trajectory of the second one. This notion may easily be related to reachable spaces of the two transitions systems, and expressed by (1), where π denotes the map that characterizes the abstraction, $Succ_1$ the reachable space according to the first transition system, $Succ_2$ the reachable space according to the second transition system, and χ a state of the first system:

$$\pi(Succ_1(\chi)) \subset Succ_2(\pi(\chi)) \tag{1}$$

This is usefull when considering safety properties of systems that can be expressed as the unreachability of specific regions of the state space from an initial region. Then the computation of the reachable space of the abstract system may be sufficient to prove the property for the first system provided that the mapping preserves the expression of the property.

[1] In this expression the . symbol denotes H, C, or D.

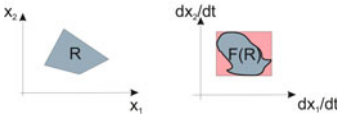


Fig. 1. Differential inclusion mapping

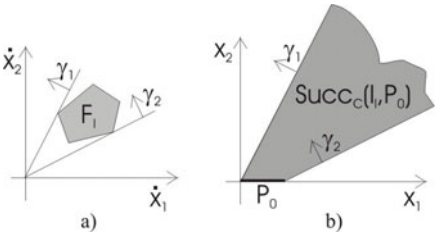


Fig. 2. Reachability for linear inclusion

3 Linear Differential Inclusion Abstraction

A first level of abstraction that may be useful to consider when dealing with verification and reachability is the abstraction of the continuous dynamics.

3.1 General Principle

The general principle of this approach [4] is illustrated on Fig. 1. As each location l of the hybrid system is associated with an invariant region, it is possible to compute the image of this region by the flow function $F(l_i, \mathbf{x}, \mathbf{u})$ associated with the location. This image (or one of its over-approximations) can then be considered as specifying the continuous dynamics of the abstracted location by differential inclusion.

So if we consider a hybrid automaton $AH =< L, X, U, Inv, F, E, G, J >$, the automaton defined by the relations below is an abstraction where the abstraction mapping π is the identity function.

- $L^* = L, X^* = X, U^* = U, Inv^*(l_i) = Inv(l_i)$
- $F^*(l, \mathbf{x}, \mathbf{u}) = F_l \supset F(l, Inv(l))$
- $E^* = E, G^* = G, J^* = J$

If the set $F_l \supset F(l, Inv(l))$ is specified by affine constraints $\mathbf{w}^T\mathbf{x} \leq b$, it is possible to easily compute the continuous reachable space by considering the most restricting constraints $\mathbf{w}^T\mathbf{x} \leq 0$ that will define with the initial region the borders of the reachable space. For example on Fig. 2, the linear inclusion (Fig. 2.a) allows computing the most restricting constraints specified by γ_1 and γ_2 that are used to compute the reachable space from P_0 (Fig. 2.b).

Generally the abstraction that is obtained by this method is too coarse and the computed reachable space does not make it possible to conclude for safety verification. The solution is then to refine the abstraction by considering subsets of the continuous regions. From the specification of the original automaton AH and an abstracted automaton $\widetilde{AH}$ obtained with the previous method or after some refinement steps, a refinement step with respect to a location $\tilde{l}_j$ of the abstracted automata $\widetilde{AH}$ allows one to compute the next abstraction $\widehat{AH}$ and is specified by the algorithm 1.

Algorithm 1. Hybrid abstraction refinement

Given: $\widetilde{AH}$ a hybrid abstraction of AH the hybrid system ,$\tilde{\pi}$ the associated abstraction mapping, $\tilde{l}_j$ one location of $\widetilde{AH}$ such as $(\tilde{l}_j, \widetilde{Inv}(\tilde{l}_j)) = \tilde{\pi}(l, \widetilde{Inv}(\tilde{l}_j))$

Result: $\widehat{AH}$ the refined abstraction of AH and $\hat{\pi}$ the associated abstraction mapping.

1: **Init:** $\widehat{AH} \Leftarrow \widetilde{AH}, \hat{\pi} \Leftarrow \tilde{\pi}$
2: choose a set of regions D_k such as $\cup D_k = \widetilde{Inv}(\tilde{l}_j)$
3: delete the location $\tilde{l}_j$ and the related elements from $\widetilde{AH}$, define for all regions D_k, $(\hat{l}_k, D_k) = \hat{\pi}(l, D_k)$, add the new locations $\hat{l}_k$ to $\hat{L}$ and define for each new location $\widehat{Inv}(\hat{l}_k) = D_k$.
4: for each location $\hat{l}_k$ define the continous dynamics $\hat{F}(\hat{l}_k, \mathbf{x}, \mathbf{u}) = F_k \supset F(l, D_k)$
5: **for all** output transitions of $\tilde{l}_j$ in $\widetilde{AH}$, $e = (\tilde{l}_j, \tilde{l}_p)$ **do**
6: **if** $\widetilde{Guard}(e) \cap D_k \neq \emptyset$ **then**
7: add the transition $e_k = (\hat{l}_k, \hat{l}_p)$ to $\hat{E}$
8: define $\widehat{Guard}(e_k) = \widetilde{Guard}(e) \cap D_k$ and $\widehat{Jump}(e_k) = \widetilde{Jump}(e)$
9: **end if**
10: **end for**
11: **for all** pairs $(\hat{l}_k, \hat{l}_n)$ of states that were created at the substep 3 above **do**
12: **if** a continuous trajectory goes from D_k to D_n according to the initial dynamics $F(l, \mathbf{x}, (u))$ **then**
13: add the transition $e = (\hat{l}_k, \hat{l}_n)$ to $\hat{E}$
14: Define $\widehat{Jump}(e)$ as the identity and $\widehat{Guard}(e)$ as the border of D_k
15: **end if**
16: **end for**

This refinement procedure may be iterated and may lead to more precise abstraction but there is no guaranty that it will eventually make it possible to conclude.

3.2 Affine Systems

An important point in this abstraction procedure is the choice of the continuous domains that are used to define the abstraction mapping because it may induce difficulties to compute the image by the continuous dynamics F and the set of transitions between the new locations. However it is possible for classes of systems to use some properties of the dynamics to guide the choice of the domains. For example for affine systems, that is systems such as the continuous dynamics in each location are specified by $\dot{\mathbf{x}} = \mathbf{A}.\mathbf{x} + \mathbf{b}$, it is possible (see [5] for more details) to consider families of hyperplanes defined by equation $H = \{x | \mathbf{q}^T\mathbf{x} = k\}$ where there exists a vector γ such as $\mathbf{q} = \mathbf{A}^T\gamma$ and $k = -\gamma^T b$, because the differential inclusion is then defined by this vector γ. Moreover if the left eigenvectors of the matrix A are used to generate the family of hyperplanes, all trajectories cross the hyperplane in the same direction leading to a very simple transition structure for the abstraction. This is exemplified on Fig. 3 in the case where the vectors q are defined as linear combinations of complex left eigenvectors. This approach may be extended to systems where the dynamics are defined by $\dot{\mathbf{x}} = \mathbf{A}.\mathbf{x} + \mathbf{Bu}$ and where U the space of continuous inputs is a polytope [6].

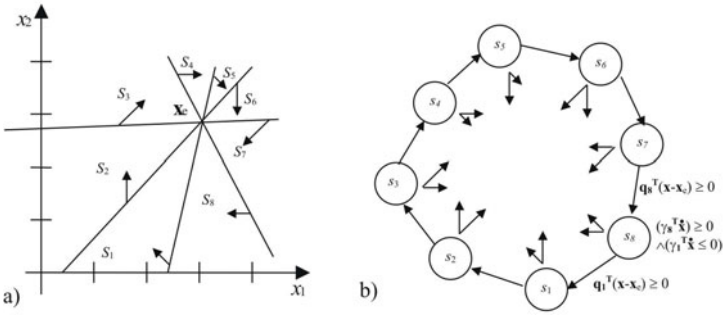


Fig. 3. Linear abstraction for affine systems

4 Discrete Event Abstraction

The second type of abstraction of hybrid systems that we will consider is discrete event abstraction. While considering a hybrid system modelled by a hybrid automaton as specified by Sect. 2, the idea is then to build a discrete event system and a map between the two state spaces such as the relation (1) is satisfied. The concept of such discrete abstractions was introduced for theoritical study of the decidability of reachability computations for hybrid systems (see e.g. the paper by [7]) but also to perform verification [8,9].

4.1 General Principles

Below we will consider the problem of computing whether the forbidden region R_f of the state space is reachable from an initial region R_i for a hybrid automaton H using a discrete event abstraction. For simplicity reasons we will also assume that each of the regions R_f and R_i is associated with only one location that is $R_f = (l_f, D_f)$ and $R_i = (l_i, D_i)$. The question is then to determine whether:

$$R_f \cap Succ_H(R_i) = \emptyset \tag{2}$$

The first step is to determine the set of states Q of the abstract system. As the aim is to prove that (2) holds it is natural to first create two states q_i and q_f and to consider that $q_i = \pi((l_i, D_i))$ and $q_f = \pi((l_f, D_f))$. The problem to solve will then change and the new problem will be to prove for the discrete system, that (3) holds provided that the discrete system is an abstraction of the hybrid one.

$$q_f \notin Succ(q_i) \tag{3}$$

Several choices are possible for the other discrete states. The simplest one is to create a discrete state for each location such as $q_k = \pi((l_k, Inv_k))$ where Inv_k is the projection of the invariant on the continuous state space, i.e. $Inv_k = \{\mathbf{x} | \exists \mathbf{u} \in U \ s.t. \ (\mathbf{x}, \mathbf{u}) \in Inv(l_k)\}$. This choice is however often too rough and it is often better to create a discrete state for each transition of the hybrid automaton with $q_e = \pi(l_k, G_e)$ where l_k is the source of the transition e and G_e is the projection of the guard on the

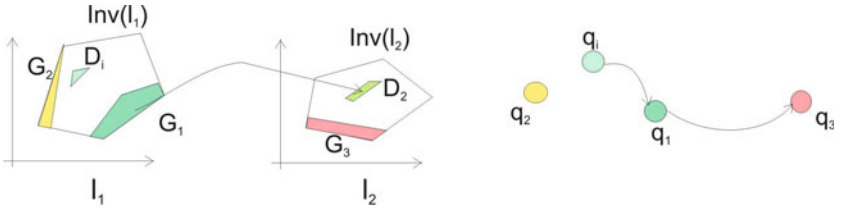


Fig. 4. Discrete mapping for abstraction on guards

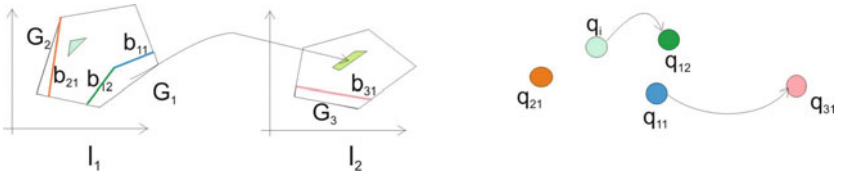


Fig. 5. Discrete mapping for abstraction on borders

state space[2] For example, on Fig. 4 each of the regions G_1, G_2, G_3 (representing the projection of the guards of some transitions) and D_i(representing the initial region) is associated with a discrete state, whereas D_2 (representing the image of G_1 by the jump function) is not. The transition from q_1 to q_3 stems from the continuous reachability of G_3 from D_2. Another possible choice is to consider the borders of this set G_e for each transition in order to get sets of lower dimension. For example, on Fig. 5 borders b_{jk} of the previous regions are associated with discrete states q_{jk} and the transition from q_i to q_{12} (but not to q_{11}) expresses the fact that all trajectories from D_i attain G_1 through the border b_{12}, whereas the transition from q_{11} to q_{31} (but not from q_{12} to q_{31}) expresses the fact that G_3 is reachable from b_{11} but not from b_{12}.

When the set Q of states is defined, the second step consists in computing the transitions of the discrete model. The constraint to be satisfied is then:

$$(l_k, \mathbf{x}_k) \in Succ_H\left((l_n, \mathbf{x}_n)\right) \Rightarrow \pi\left((l_k, \mathbf{x}_k)\right) \in Succ\left(\pi\left((l_n, \mathbf{x}_n)\right)\right) \tag{4}$$

Relative to the principle used to define the discrete states, local considerations may be used to satisfy this constraint. For the examples given above it is possible to show that the constraint is satisfied when using the following procedure:

- for $q_i = \pi\left((l_i, D_i)\right)$ add a transition (q_i, q) for all $q = \pi\left((l_i, D)\right) \in Q$ such as $(l_i, D) \cap Succ_C\left(l_i, D_i\right)) \neq \emptyset$,
- for all discrete states $q_k \in Q - \{q_i, q_f\}$ add a transition (q_k, q) for all $q = \pi\left((l, D)\right) \in Q$ such as $(l, D) \cap Succ_C\left(Succ_D\left(Succ_C\left(l_k, D_k\right)\right)\right) \neq \emptyset$.

When the discrete states are defined by the guards (case of Fig. 4) an alternative that is based on only one continuous successor set is given below. Of course this new procedure leads to another discrete event system that also satisfies constraint (4).

[2] i.e. $G_e = \{\mathbf{x} | \exists \mathbf{u} \in U \; s.t. \; (\mathbf{x}, \mathbf{u}) \in Inv(l_k) \cap Guard(e)\}$.

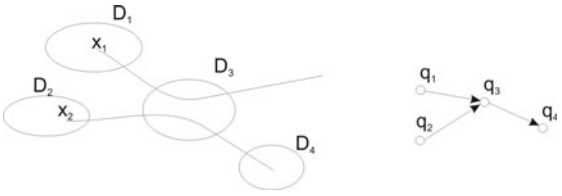


Fig. 6. Spurious transitions

- for $q_k = \pi((l_k, D_k))$ add a transition (q_k, q) for all $q = \pi((l_k, D)) \in Q$ such as $(l_k, D) \cap Succ_C(l_k, D_k)) \neq \emptyset$,
- for all discrete states $q_k \in Q - \{q_f\}$ add a transition (q_k, q) for all $q = \pi((l, D)) \in Q$ such as $(l, D) \cap Succ_C(Succ_D(l_k, D_k))) \neq \emptyset$.

During the abstraction procedure that leads to the discrete event system, the only consideration is to garanty that the satisfaction of (3) will prove the satisfaction of (2). However the non satisfaction of (3) does not necessary imply that (2) is not satisfied. This is due to the over-approximation of the reachable space introduced by the abstraction. This is illustrated in Fig. 6 where the abstraction process will lead to conclude that D_4 is reachable from D_1.

If the abstraction is too rough to conclude about the satisfaction of (2) from (3) it is possible to refine it. The idea of refinement is to consider a discrete transition and to make a partition of the continuous domain of the region mapped to the source location in order to introduce two new discrete states, one from which the transition can be fired and the other one from which it can not [10]. The detailed process depends on the process used to build the abstraction. For example if the second procedure presented above is used, it is possible to refine the abstraction with respect to a transition (q_k, q_p), where $q_k = \pi((l_k, D_k))$ and $q_p = \pi((l_p, D_p))$, by using the procedure specified by algorithm 2.

At the end of this refinement step, it is possible to reach (l_p, D_p) from all states of (l_k, D^1) and the new discrete transition does not introduce over-approximation. The difficulty is obviously to choose the transition that has to be considered for refinement. It is then possible to begin with transitions leading to the closest regions to the forbidden area [11], or to use the result of the verification and especially the counter-example trajectory given by the checking tool, to guide the refinement [12]. Another difficulty is that a refinement according to one transition is in general not sufficient to conclude that (3) is satisfied. So this procedure has to be iterated. If the procedure stops because there are no more changes in the discrete system for all transitions, it is eventually possible to conclude that if (3) is not satisfied then neither is (2). However there is no proof that this convergence will always happen except for specific systems, such as time automata [7].

Algorithm 2. Discrete abstraction refinement

Given: $< Q, T >$ a discrete abstraction of AH the hybrid system , π its abstraction mapping, (q_k, q_p) a transition of T such as $q_k = \pi((l_k, D_k))$ and $q_p = \pi((l_p, D_p))$

Result: $< \tilde{Q}, \tilde{T} >$ the discrete abstraction of AH refined with respect to (q_k, q_p) and $\tilde{\pi}$ its abstraction mapping

1: **Init:** $\tilde{Q} \Leftarrow Q, \tilde{T} \Leftarrow T, \tilde{\pi} \Leftarrow \pi$
2: if $l_k = l_p$ compute $D^1 = D_k \cap Pred_C(l_p, D_p)_{|l_k}$[3] and $D^2 = D_k - D^1$
3: if $l_k \neq l_p$ compute $D^1 = D_k \cap Pred_D\left(Pred_C(l_p, D_p)\right)_{|l_k}$ and $D^2 = D - D^1$
4: **if** $D^1 \neq \emptyset$ and $D^2 \neq \emptyset$ **then**
5: delete q_k from $\tilde{Q}$ and $q_k = \pi((l_k, D_k))$ from the defintion of $\tilde{\pi}$. Define $q_k^1 = \tilde{\pi}\left((l_k, D^1)\right)$ and $q_k^2 = \tilde{\pi}\left((l_k, D^2)\right)$ and add them to $\tilde{Q}$
6: Delete (q_k, q_p) from the set of transitions $\tilde{T}$ and add (q_k^1, q_p) to $\tilde{T}$
7: for all transitions $(q_n, q_k) \in T$, delete (q_n, q_k) from $\tilde{T}$ and compute whether (q_n, q_k^1) and (q_n, q_k^2) belong to $\tilde{T}$
8: **end if**

4.2 Remarks

This approach that builds a discrete event model of a continuous or hybrid system may be found in various propositions [13, 14, 11, 15, 16, 17] according to the assumptions that are made about the system (guards, invariants, continuous dynamics, ...). The exact specification of the procedures depends on these assumptions and on the choices that are made to specify the mapping. In all cases, computation of successor and predecessor sets (and especially the continuous ones) is a core problem of the approach.

5 Reachability for Abstractions

As it can be seen in the previous sections an important point to build an abstraction is to be able to state what are the reachable regions. Two types of answers may be searched for. When the problem is to decide whether there is a discrete transition between two locations in Sect. 3 or two discrete states in Sect. 4, it is possible to use methods that give a yes or no answer, thus characterizing the reachable space without computing it. However, it is of course possible to compute the reachable space. When it is necessary to refine the abstraction in Sect. 4, it becomes mandatory to get an estimation of the predecessor space and then to compute it. It can also be noticed that in both approaches these reachability considerations are related to one location and are then mainly concerned by continuous reachability.

5.1 Characterizing the Reachable Space

Families of approaches that characterize the continuous reachable space may be used to answer the question whether it is possible to reach a given area from another one. A first solution is to find borders that separate the two domains and can't be crossed by the continuous trajectories as exemplified on Fig. 7.

For some specific classes of continuous dynamics it is possible to find generic constraints. For example [18], for linear dynamics without inputs $\dot{\mathbf{x}} = \mathbf{A}.\mathbf{x}$, the derivative of the linear term $p = \mathbf{c}^T\mathbf{x}$ where $\mathbf{c}$ is a left eigenvector of $\mathbf{A}$ associated to the the eigenvalue λ, is $\dot{p} = \lambda p$, and hence $p = e^{\lambda t}\mathbf{c}^T\mathbf{x}_0$ for any continuous trajectory. So according

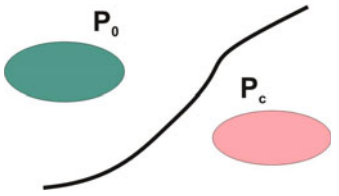


Fig. 7. Uncrossable Borders

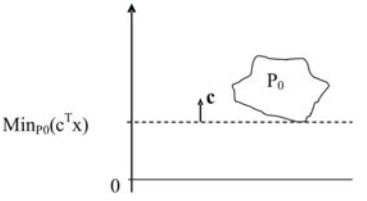


Fig. 8. Qualitative invariant

to the sign of λ, the term p is always increasing or decreasing and its minimum or maximum value defines an uncrossable border. As exemplified on Fig. 8, if λ is positive the region $\{\mathbf{x}|\mathbf{c}^T\mathbf{x} < Min_{P_0}(\mathbf{c}^T\mathbf{x})\}$ is unreachable. Such considerations may also be used for affine systems with rational eigenvalues [19].

Another solution that can be used to characterize the reachable space is the concept of barrier certificate [20]. The idea is then to determine a function B that verifies the three conditions in (5) as illustrated on Fig. 9. Then the set where this function is equal to 0 is an uncrossable border that guarantee that the region P_c is unreachable from P_0. With some specific assumptions on the dynamics and the regions it is possible to search for this function B by optimization techniques.

$$\begin{aligned} \forall \mathbf{x} \in P_C \qquad & B(\mathbf{x}) > 0 \\ \forall \mathbf{x} \in P_0 \qquad & B(\mathbf{x}) < 0 \\ \forall \mathbf{x} \in X,\ \forall \mathbf{u} \in U \qquad & B(\mathbf{x}) = 0 \Rightarrow \frac{\partial B(\mathbf{x})}{\partial x} f(\mathbf{x}, \mathbf{u}) \leq 0 \end{aligned} \tag{5}$$

A last appraoch consists in searching for characteristics of the reachable space in specific subspaces, such as the eigensubspaces for linear systems, and then to prove that the global set of constraints is inconsistent. This approach will be illustrated in the case of linear systems (specified by ($\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$) and temporal constraints [21] but can also be used with spacial constraints [22].

Let us consider the problem of reachability of region P_C from region P_0 and an eigenvalue λ of $\mathbf{A}$ with an associated eigen subspace with dimension 1 (this case is illustrated for 2 dimensions on Fig. 10). It is easy to compute, for example with linear programming, the upper and lower bounds of the projections of these regions (P_0 and P_C) on the eigen subspace associated with λ, (z_0^l, z_0^u) and (z_C^l, z_C^u). In other respects, the projection of the trajectory from the point $\mathbf{x}_0$ is specified by $z(t) = z_0 e^{\lambda t}$ where z_0 is the projection of $\mathbf{x}_0$. It is then possible to compute the minimum and maximum time necessary to go from the projection of region P_0 to the one of P_C. For example the maximum time is given by $t_{max,\lambda} = \max(\frac{1}{\lambda}\log(\frac{z_C^l}{z_0^u}), \frac{1}{\lambda}\log(\frac{z_C^u}{z_0^l}))$.

If the maximum time is negative according to this equation, it means that P_C is unreachable from P_0 as its projection is unreachable in one eigen subspace. If this computation is performed with another eigen subspace, it is possible to consider the two time intervals. It can then be deduced from the emptiness of their intersection that P_C is not reachable from P_0.

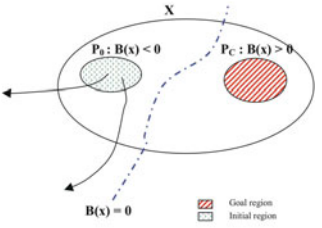


Fig. 9. Barrier certificate

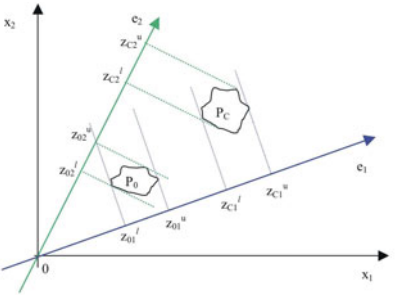


Fig. 10. Projections bounds

5.2 Reachable Space Computation

When it is necessary to refine a discrete abstraction it is mandatory to compute the predecessor set of a given region in order to be able to compute its intersection with other regions as explained in Sect. 4. The major difficulty is to compute the continuous predecessor set, or at least an over-approximation, in order to preserve safety conclusions.

For continuous systems specified by linear differential inclusions, the continuous predecessor set can be computed with geometric considerations and polytopes computations as in the case of the successor set in Sect. 3 (Fig. 2). For other classes of continuous dynamics it is possible to consider finite time predecessor sets that can be computed by time sampled computations. In general these approaches are devoted to successor sets computation but may be adapted to predecessor set for example by considering negative time steps.

Beside approaches based on interval integration of ordinary differential equations [23], most approaches are based on the same basic principle that consists in choosing a time step δ and computing the series of the over-approximations P_k of the reachable space between times $k\delta$ and $(k+1)\delta$, [8, 24, 25, 26]. The first steps of the method are illustrated on Fig. 11 when polyhedral regions are chosen. The first step (Fig. 11.a) consists in computing the image, X_1, of the initial region, X_0, after time δ. The second step (Fig. 11.b) then consists in searching for a polyhedron P_0 that includes the whole trajectory between the two times. These steps are then iterated to compute the reachable space at the next times (Fig. 11.c). In order to reduce the approximations, it is generally pertinent to use X_i as the basis of the next iteration. However, the second step that computes P_{i-1} from X_{i-1} and X_i is complex and time consuming, so it is also possible to adapt this procedure and to compute P_i without computing X_i directly from P_{i-1}. This is especially relevant when this does not induce further approximations as in the case of linear dynamics ($\dot{\mathbf{x}} = \mathbf{Ax}$). The evolution from one time step to the other one is then known, constant and given by $e^{A\delta}$, so, from the computation of P_0 it is easy to iteratively compute the series $P_i = e^{A\delta} P_{i-1}$.

This approach can also be used for systems specified by $\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{u}$ where $\mathbf{u}$ stands for a bounded uncertainty. At each time step the approximation of the reachable space is then given [24] by $P_i = e^{A\delta} P_{i-1} \oplus V$ where V is a region that depends on the uncertainty and the dynamics, and $\oplus$ is the Minkowski sum[4] of the sets.

[4] The Minkowski sum of 2 sets A and B, is defined by $A \oplus B = \{\, a + b \mid a \in A \ \wedge \ b \in B \,\}$.

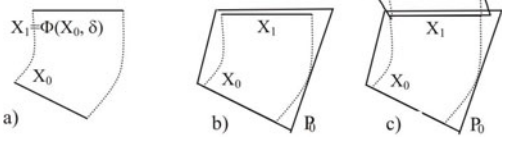


Fig. 11. Sampled approximation

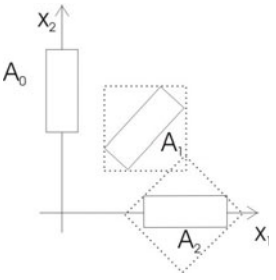


Fig. 12. Wrapping effect

This reachability computation requires computations on regions of the continuous state space and then the choice of the representation of these regions. If some works consider polynomial regions, (see for example [27]), classical sets that are considered in reachability computation are ellipsoids [28] and various types of polyhedrons such as hyperrectangles (especially with interval computation see e.g. [29]), zonotopes [30,24] or general polyhedrons.

Two main points are to consider in the choice of the region representation. The first one is that the set of regions must be closed for the operations that are involved in the computation. This is exemplified on Fig. 12 for the hyperrectangle representation and the computation of the image by the dynamics. A_1, representing the image of A_0 after one time step, is not a hyperrectangle, so it has to be overapproximated by the hyperrectangle in dash line. The result of the second step is the overapproximation of the image of this hyperrectangle whereas A_2, representing the real image of A_0 after two time steps, is given by the smaller hyperrectangle.

A second issue is the complexity of the representation and the possibilities to simplify it. For example, in the case of polyhedrons the number of constraints that are used to describe the region may increase at each step and it may be useful to be able to easily compute a tight overapproximation with less constraints in order to simplify the computation as examplified on Fig. 13.

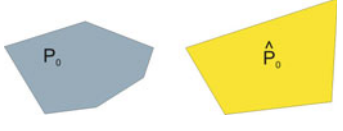


Fig. 13. Number of constraints reduction

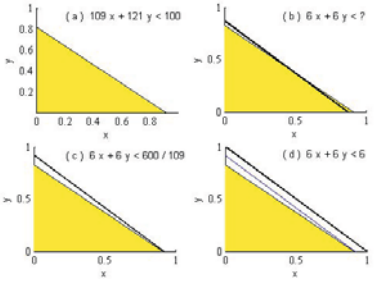


Fig. 14. Limiting the number of bits [32]

6 Conclusions

Safety verification of continuous time hybrid systems may be performed by means of abstraction, a formal notion that makes it possible to guarantee the relevance of the analysis conclusions and to transform a complex problem by mean of a number of simpler local problems. One important limitation of the abstraction is that it allows checking whether safety properties are satisfied but not always guarantee a solution if they are not. The refinement of the model when the analysis is not conclusive is an issue that requires some know-how and a crucial point is the formalization of this know-how and its integration in tools in order to aid control engineers during safety analysis.

References

1. Guéguen, H., Zaytoon, J.: On the formal verification of hybrid systems. Control Engineering Practice 12, 1253–1268 (2004)
2. Guéguen, H., Lefebvre, M., Zaytoon, J., Nasri, O.: Safety verification and reachability analysis for hybrid systems. Annual Reviews in Control 33, 25–36 (2009)
3. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T., Ho, P., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Theoretical Computer Science 138, 3–34 (1995)
4. Henzinger, T.A., Ho, P., Wong-Toi, H.: Algorithmic analysis of nonlinear hybrid systems. IEEE Trans. on Automatic Control 43, 540–554 (1998)
5. Lefebvre, M., Guéguen, H.: Hybrid abstractions of affine systems. NonLinear Analysis: Theory and Methods 65, 1150–1167 (2006)
6. Nasri, O., Lefebvre, M., Gueguen, H.: Abstraction based reachability computation for affine systems with bounded input. In: CDC Conference on Decision and Control, pp. 2609–2613. IEEE, Los Alamitos (2006)
7. Alur, R., Henzinger, T., Laferriere, G., Pappas, J.: Discrete abstractions of hybrid systems. Proceedings of IEEE 88, 971–984 (2000)
8. Chutinan, A., Krogh, B.: Computation techniques for hybrid systems verification. IEEE Trans. on Automatic Control 48, 64–75 (2003)
9. Chutinan, A., Krogh, B.: Verification of infinite-state dynamic systems using approximate quotient transition systems. IEEE Trans. on Automatic Control 46, 1401–1410 (2001)
10. Tabuada, P., Pappas, G., Lima, P.: Composing abstractions of hybrid systems. In: Tomlin, C.J., Greenstreet, M.R. (eds.) HSCC 2002. LNCS, vol. 2289, pp. 436–450. Springer, Heidelberg (2002)
11. Alur, R., Ivancic, F., Dang, T.: Progress on reachability analysis of hybrid systems using predicate abstraction. In: Maler, O., Pnueli, A. (eds.) HSCC 2003. LNCS, vol. 2623, pp. 4–19. Springer, Heidelberg (2003)
12. Fehnker, A., Clarke, E., Jha, S., Krogh, B.: Refining abstractions of hybrid systems using counterexample fragments. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 242–257. Springer, Heidelberg (2005)
13. Tiwari, A., Khanna, G.: Nonlinear systems: approximating reach sets. In: Alur, R., Pappas, G.J. (eds.) HSCC 2004. LNCS, vol. 2993, pp. 600–614. Springer, Heidelberg (2004)
14. Alur, R., Dang, T., Ivancic, F.: Reachability analysis of hybrid systems via predicate abstraction. In: Tomlin, C.J., Greenstreet, M.R. (eds.) HSCC 2002. LNCS, vol. 2289, pp. 35–48. Springer, Heidelberg (2002)

15. Ratschan, S., She, Z.: Safety verification of hybrid systems by constraint propagation based abstraction refinement. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 573–589. Springer, Heidelberg (2005)
16. Blouin, S., Guay, M., Rudie, K.: Discrete abstractions for two dimensional nearly integrable continuous systems. In: Engel, S., Guéguen, H., Zaytoon, J. (eds.) ADHS 2003: IFAC Conference on Analysis and Design of Hybrid Systems, Saint-Malo, France, pp. 343–348. Elsevier, Amsterdam (2003)
17. Kloetzer, M., Belta, C.: Reachability analysis of multi-affine systems. In: Hespanha, J.P., Tiwari, A. (eds.) HSCC 2006. LNCS, vol. 3927, pp. 348–362. Springer, Heidelberg (2006)
18. Tiwari, A.: Approximate reachability for linear systems. In: Maler, O., Pnueli, A. (eds.) HSCC 2003. LNCS, vol. 2623, pp. 514–525. Springer, Heidelberg (2003)
19. Rodriguez-Carbonell, E., Tiwari, A.: Generating polynomial invariance for hybrid systems. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 590–605. Springer, Heidelberg (2005)
20. Prajna, S., Jadbabaie, A., Pappas, G.: A framework for worst-cas and stochastic safety verification using barrier certificates. IEEE Transactions on Automatic control 52, 1415–1428 (2007)
21. Yazarel, H., Pappas, G.: Geometric programming relaxations for linear systems reachability. In: American Control Conference, pp. 553–559 (2004)
22. Yazarel, H., Prajna, S., Pappas, G.J.: Sos for safety. In: 43rd IEEE CDC, pp. 461–466 (2004)
23. Lohner, R.: Enclosing the solutions of ordinary initial and boundary value problems. In: Computer Arithmethic: Scientific Computation and Programming Languages. Wiley-Teubner Series in Computer Science, pp. 255–286 (1987)
24. Girard, A.: Reachability of uncertain linear systems using zonotopes. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 291–305. Springer, Heidelberg (2005)
25. Asarin, E., Dang, T., Frehse, G., Girard, A., Guernic, C.L., Maler, O.: Recent progress in continuous and hybrid reachability analysis. In: CACSD 2006, Munich, Germany (2006)
26. Hickey, T., Wittenberg, D.: Rigourous modelling of hybrid systems using interval arithmetic constraints. In: Alur, R., Pappas, G.J. (eds.) HSCC 2004. LNCS, vol. 2993, pp. 402–416. Springer, Heidelberg (2004)
27. Dang, T.: Approximate reachability computation for polynomial systems. In: Hespanha, J.P., Tiwari, A. (eds.) HSCC 2006. LNCS, vol. 3927, pp. 138–152. Springer, Heidelberg (2006)
28. Kurzhanski, A., Variya, P.: Ellipsoïdal techniques for reachability analysis. In: Lynch, N.A., Krogh, B.H. (eds.) HSCC 2000. LNCS, vol. 1790, pp. 202–214. Springer, Heidelberg (2000)
29. Nedialkov, N., Jackson, K., Corliss, G.: Validated solutions of initial value problems for ordinary differential equations. Applied Mathematics and Computation 105, 21–68 (1999)
30. Kühn, W.: Zonotope dynamics in numerical quality control. In: Hege, H.C., Polthier, K. (eds.) Mathematical Visualization, pp. 125–134. Springer, Heidelberg (1998)
31. Bagnara, R., Ricci, E., Zaffanella, E., Hill, P.: Possibly not closed convex polyhedra and the parma polyhedra library. In: Hermenegildo, M., Puebla, G. (eds.) SAS 2002. LNCS, vol. 2477, pp. 213–229. Springer, Heidelberg (2002)
32. Frehse, G.: Phaver: algorithmic verification of hybrid systems past hytech. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 258–273. Springer, Heidelberg (2005)

Meeting the World Challenges with Advanced System Organizations

Peter Simon Sapaty

Institute of Mathematical Machines and Systems, National Academy of Sciences
Glushkova Ave 42, 03187 Kiev, Ukraine
sapaty@immsp.kiev.ua

Abstract. We have been witnessing numerous world crises and disasters—from ecological to military to economic, with global dynamics likely to be increasing further. The paper highlights known holistic and gestalt principles mainly used for a single brain, extending them to any distributed systems which may need high integrity and performance in reaction to unpredictable situations. A higher organizational layer is proposed enabling any distributed resources to behave as an organism having global "consciousness" and pursuing global goals. This "over-operability" layer is established by implanting into key system points the same copy of a universal intelligent module, which can communicate with other such modules and interpret collectively global mission scenarios presented in a special Distributed Scenario Language. The behavioral scenarios can be injected from any module, and then self-replicate, self-modify, and self-spread throughout the system to be managed, tasking individual components, activating distributed resources, and establishing runtime infrastructures supporting system's integrity. Existing and prospective applications are outlined and discussed, confirming the paradigm's suitability for solving world problems.

Keywords: World crises, Atomism, Holism, Gestalt theory, System integrity, Waves, Distributed scenario language, Networked interpretation, Unmanned systems, Task level, Behavioral level, Crisis management, Robotized armies, Terrorism fight, First computers.

1 Introduction

To understand mental state of a handicapped person, problems of economy and ecology, or how to win on a battlefield, we must consider the system as a whole not just as a collection and interaction of parts. The situation may complicate dramatically if the system is dynamic and open, spreads over large territories, comprises unsafe or varying components, and cannot be observed in its entirety from a single point. Numerous world crises we have been witnessing at the beginning of this century, including the current economic one, may have emerged, first of all, due to our inability of seeing and managing complex systems as a whole.

To withstand the unwanted events and their consequences (ideally: predict and prevent them) we need effective worldwide integration of numerous efforts and often

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 29–46.
springerlink.com

dissimilar and scattered resources and systems. Just establishing advanced communications between parts of the distributed systems and providing the possibility of sharing local and global information from different points, often called *interoperability*, is becoming insufficient (even insecure and harmful) for solving urgent problems in dynamic environments, in real time and ahead of it.

We may need the whole distributed system to behave as an integral organism, with parts not so interoperating but rather complementing each other and representing altogether an integral whole pursuing global goals and having a sort of global awareness and consciousness. This whole should be essentially more than the sum of its parts, with the latter having sense, possibly even existence, in the context of this whole, rather than vice versa.

This paper develops further the over-operability principle researched in [1-4] and other works (with the term *overoperability* coined in [3]), which can establish intelligent dominant layer over distributed resources and systems and help to solve urgent world problems in a parallel, distributed, and dynamic way.

The rest of this paper compares the dominant atomistic approach in system design, implementation and management with holistic and gestalt principles, and describes a novel ideology and technology for integral solutions in distributed worlds, which can avoid many traditional management routines in solving global problems, with its numerous practical applications outlined and discussed.

2 Atomism, Holism, Gestalt

We used to exercise predominantly atomistic, parts-to-whole philosophy of the system design, comprehension and implementation, which extends even to the organization of management facilities themselves as a collection of interacting parts, or *agents* (this philosophy actually being the same as a century ago).

Originally a system or campaign idea and the functionality needed emerge in a very general form (in a single human mind or in a close collective of such minds). Then this general idea (shown symbolically in Fig. 1*a*) is partitioned into individual chunks, or "atoms", each detailed and studied further (Fig. 1*b*). This logical partitioning already causes swelling of the problem's complexity (as indicated in the figure).

The next step is materialization of the defined parts and their distribution in a physical or virtual space. To make these parts work or behave together within the original idea of Fig. 1*a*, we may need a good deal of their communication and synchronization, also sophisticated control infrastructures, as depicted in Fig. 1*c*. This overhead may be considerable, outweighing and shadowing the original project definition.

The main problem is that the initial idea (Fig. 1*a*) and even its second stage (Fig. 1*b*) usually remain in the minds of the creators only, and the *real system formalization, description, and implementation are starting from the already partitioned and interlinked stage* (Fig. 1*c*)*, resulting in this huge overhead*.

This parts-to-whole, agents-based approach also dominates in the controversial "society of mind" theory [5], which is trying to explain even human thinking from the atomistic positions.

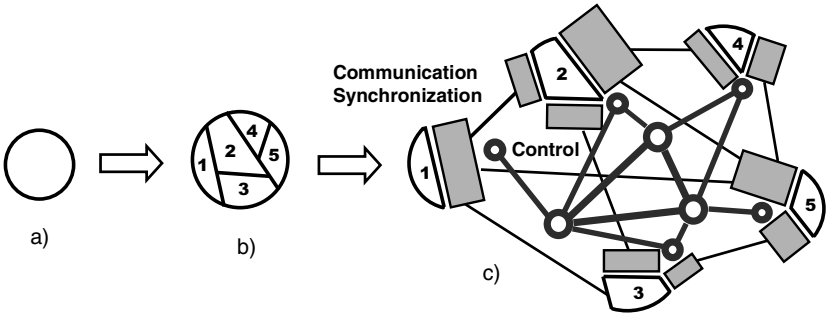


Fig. 1. Traditional atomistic approach in system design and management. (a) the original system idea; (b) breaking the idea into pieces; (c) system formalization, distribution, and implementation.

Holism [6] has quite an opposite vision of systems:

- Holism as an idea or philosophical concept is diametrically opposed to atomism.
- Where the atomist believes that any whole can be broken down or analyzed into its separate parts and the relationships between them, the holist maintains that the whole is primary and often greater than the sum of its parts.
- The atomist divides things up in order to know them better; the holist looks at things or systems in aggregate.

Gestalt theory [7, 8] is based on the holistic principles too:

- For the gestaltists, "Gestalten" are not the sums of aggregated contents erected subjectively upon primarily given pieces.
- Instead, we are dealing with wholes and whole–processes possessed of inner intrinsic laws.
- *Elements* are determined as parts by the intrinsic conditions of their wholes and are to be understood *as parts* relative to such wholes."

Although gestalt psychology and theory was a general approach, most of the work on gestalt was done in the area of perception. *In our research, we are trying to use the holistic and gestalt principles for the organization of distributed systems with highest possible integrity and performance* [9].

3 Waves, Fields, Scenarios

We describe here a novel organizational philosophy and model, based on the idea of spreading *interdependent parallel waves* (as shown in Fig. 2a), as an alternative to the dominant atomistic approach briefed above, staying under the influence of mentioned holistic and gestalt principles.

This allows us for an integral, parallel, and seamless navigation and coverage of virtual, physical or combined spaces where the solutions need to be found. Atomism emerges on the automatic implementation level only, which allows us to get high-level formal semantic definitions of systems and global operations in them, while omitting

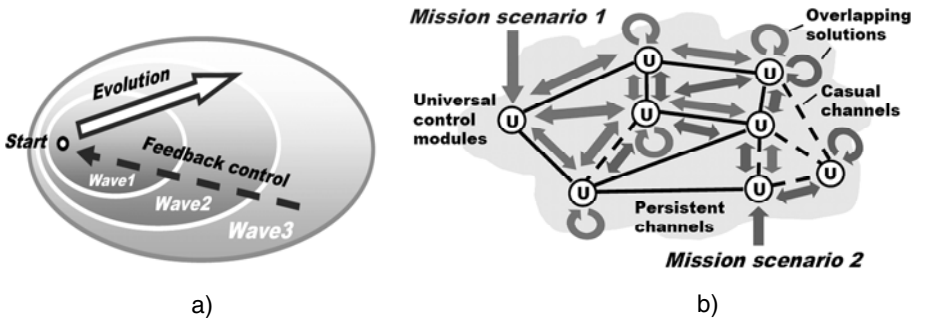


Fig. 2. The waves paradigm. (a) controlled grasping of distributed world with spatial waves; (b) self-spreading high-level mission scenarios in distributed networked worlds.

numerous organizational details (as in Fig. 1*c*) and concentrating on global goals and overall performance instead.

An automatic materialization of this approach is carried out by the network of universal intelligent modules (U), embedded into important system points, which collectively interpret integral mission scenarios expressed in the waves formalism, which can start from any U, subsequently covering the distributed system at runtime (see Fig. 2*b*).

The wavelike scenarios are usually very compact and can be created and modified on the fly. They can cooperate or compete with each other in the distributed networked space as overlapping fields of parallel solutions.

Spreading waves can create knowledge infrastructures arbitrarily distributed between system components (robots, sensors, humans). These, subsequently or simultaneously navigated by same or other waves, can effectively support distributed databases, command and control, situation awareness, and autonomous decisions.

This paradigm is much in line with the existing abundant evidence that certain aspects of cognition, morals, needs, object relations, motor skills, and language acquisition proceed in developmental stages. These stages appear to be fluid, flowing, overlapping waves [10], where:

- Each stage has a holistic pattern that blends all of its elements into a structured whole;
- These patterns unfold in a relational sequence, with each senior wave transcending but including its juniors.

Our approach is also consistent with the ideas of self-actualization and person-centered approach [11, 12], where the self is considered as an organized, consistent, conceptual gestalt exhibiting *active forward thrust* -- against tension reduction, equilibrium, or homeostasis [13]. In our case, instead of a single person we have the whole distributed system with high integrity and "active global thrust" behavior.

4 The Scenario Language

Distributed Scenario Language, or DSL (similar to is previous versions, WAVE and WPL including [2, 4]), reflects the waves model proposed, and allows us to directly

express semantics of problems to be solved in distributed worlds, also the needed global system behavior in a non-atomistic manner. DSL operates with:

- *Virtual World (VW)*, which is discrete and consists of nodes and links connecting these nodes.
- Continuous *Physical World (PW)*, any point in which may be accessed by physical coordinates (taking into account certain precision).
- *Virtual-Physical World (VPW)*, which is an extension of VW where nodes additionally associate with certain coordinates in PW.

It also has the following key features:

- A DSL scenario develops as a transition between sets of progress points (or *props*) in the form of parallel *waves*.
- Starting from a prop, an action may result in one or more props (the resultant set of props may include the starting prop too).
- Each prop has a resulting value (which can be multiple) and a resulting state, being one of the four: *thru* (full success, also allowing us to proceed further), *done* (success with planned termination), *fail* (regular failure, with local termination), and *abort* (emergency failure, terminating the whole distributed process, associated with other props too).
- Different actions may evolve independently or interdependently from the *same* prop, contributing to (and forming altogether) the resultant set of props.
- Actions may also *spatially succeed each other*, with new ones applied in parallel from all the props reached by preceding actions.
- Elementary operations can directly use local or remote values of props obtained from other actions (or even from the whole scenarios).
- Elementary operations can result either in open values that can be directly used as *operands* by other operations in an expression, or by the *next operations* in a sequence. They can also be directly assigned to *local or remote variables* (for the latter case, an access to these variables may invoke scenarios of any complexity).
- Any prop can associate with a *node* in VW or a *position* in PW, or *both* -- when dealing with VPW.
- Any number of props can be simultaneously linked with the same points of the worlds.
- Staying with world points (virtual, physical, or combined) it is possible to *directly access* and update local data in them.
- Moving in physical, virtual or combined worlds, with their possible modification or even creation from scratch, are as routine operations as, say, arithmetic or logical operations of traditional programming languages.
- DSL can also be used as a usual universal programming language (like C, Java, or FORTRAN).

DSL has recursive syntax, being on top level as follows (programs called *waves*, as parallel wavelike movement in distributed spaces dominates the language semantics).

wave → *phenomenon* | *rule* ({ *wave* , })
phenomenon → *constant* | *variable* | *special*
constant → *information* | *matter* | *combined*
variable → *heritable* | *frontal* | *environmental* | *nodal*
rule → *movement* | *creation* | *elimination* | *echoing* | *fusion* | *verification* | *assignment* | *advancing* | *branching* | *transference* | *timing* | *granting*

The basic construct, *rule*, can represent any definition, action or decision, being for example:

- elementary arithmetic, string or logic operation;
- hop in a physical, virtual, or combined space;
- hierarchical fusion and return of (remote) data;
- distributed control, both sequential and parallel;
- a variety of special contexts for navigation in space, influencing operations and decisions;
- type or sense of a value, or its chosen usage, guiding automatic interpretation.

There are different types of variables in DSL:

- *Heritable variables* – these are starting in a prop and serving all subsequent props, which can share them in both read & write operations.
- *Frontal variables* – are an individual and exclusive prop's property (not shared with other props), being transferred between the consecutive props, and replicated if from a single prop a number of props emerge.
- *Environmental variables* – are accessing different elements of physical and virtual words when navigating them, also a variety of parameters of the internal world of DSL interpreter.
- *Nodal variables* – allow us to attach an individual temporary property to VW and VPW nodes, being accessed and shared by props associated with these nodes.

These variables, especially when used together, allow us to create efficient spatial ("holographic") algorithms which work *in between* components of distributed systems rather than *in* them.

Elementary DSL programming examples are shown in Fig. 3.

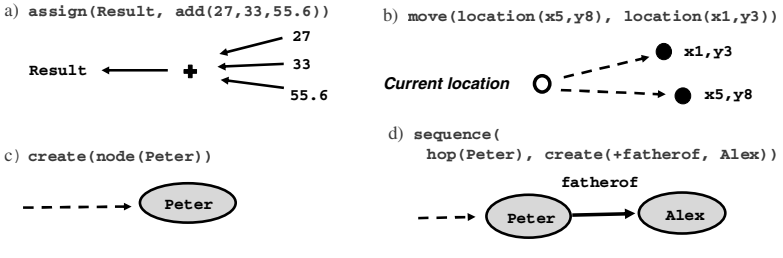


Fig. 3. Elementary examples in DSL. (a) assignment of a sum of values to a variable; (b) parallel hop into two physical locations; (c) creating of an isolated node in a virtual space; (d) extension of the network with a new link-node pair.

5 Composition of Waves

The language allows for an integral parallel navigation of distributed worlds in a controlled *depth and breadth mode*, with any combinations of the two. We will highlight here key possibilities of doing this by composition of DSL scenarios, or waves.

5.1 Single Wave Features

Single wave (let it be *W1*) development features are shown in Fig. 4. Starting from a prop, which may be associated with a point in the world, the related scenario evolves, grasps, and covers certain region in it, performing any operations needed in the distributed space.

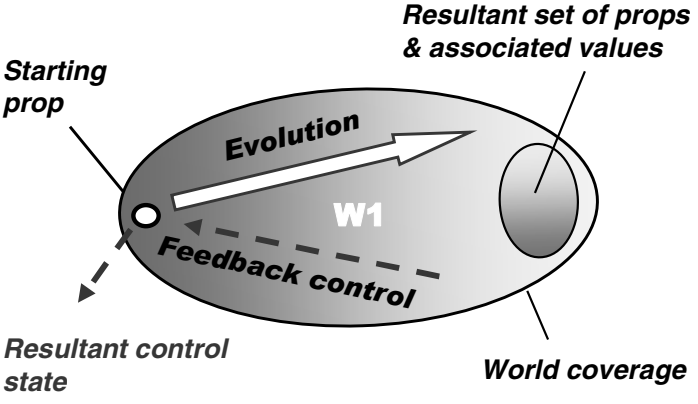


Fig. 4. Single wave features

The result of this spatial evolution may be multiple, and may lie in a (final) subregion of the region covered, being represented by a set of resultant props (each linked to world points) and associated with them values. After termination of the wave, its resultant control state (which, in a parallel feedback process, merges termination states throughout the region covered) is available in the starting prop, and may be taken into account for decisions at higher levels. Also, if requested from higher levels, the values associated with the resultant props (which may be remote) can be lifted, spatially raked, and returned to the starting prop for further processing.

5.2 Advancing in Space

The depth mode development of waves is shown in Fig. 5. For this type of composition, each subsequent wave is applied in parallel from all props in space reached by the previous wave, with the resultant set of props (and associated values) on the whole group being the one of the last applied wave (i.e. *W4* in the figure).

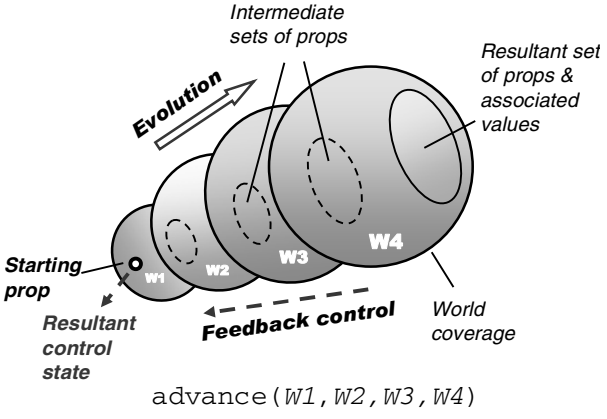


Fig. 5. Depth mode composition of waves

This spatial advancement of waves returns the resultant control state which is available at the starting prop, and the values of the resultant set of props can also be echoed to the starting prop if requested. Examples of other advancing rules:

- `advance synchronized` – the one where any new wave is applied only after all invocations of the previous wave have been terminated;
- `repeat` – where the same wave is applied repeatedly from all props reached by its previous invocation;
- `repeat synchronized` – where in the repeated invocation of a wave each new invocation starts only after full completion of the previous one.

5.3 Branching in Space

The branching breadth mode composition of waves is shown in Fig. 6, where all waves in the group are evolving from the same starting prop, and each wave, with its own resultant set of props and associated values, contributes to the final result.

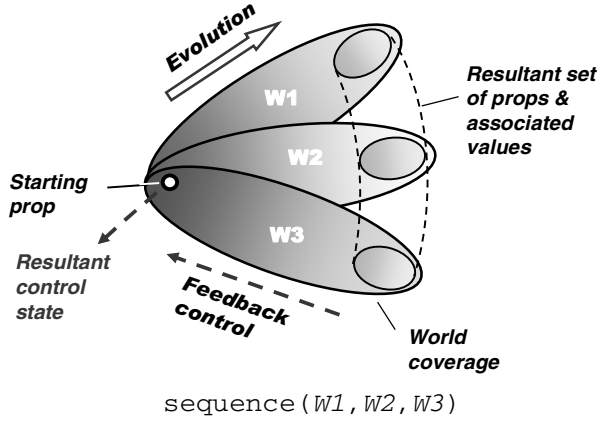


Fig. 6. Breadth mode composition of waves

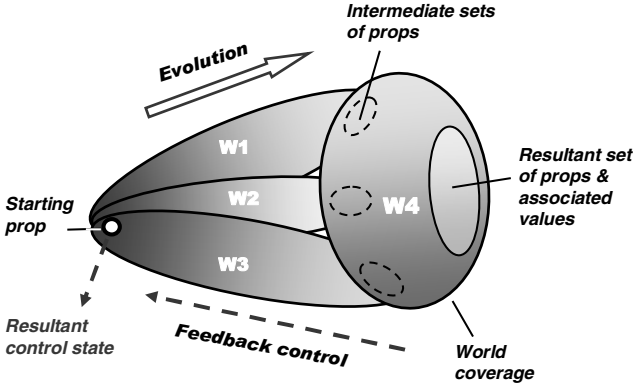
The merge of results from different waves depends on the branching rule used, with their repertoire (besides the `sequence` in Fig. 7) including:

`if, while, parallel, or, parallel or, and, parallel and, cycle, loop,` and `sling.`

More details on these and other rules can be found, say, from [2, 4].

5.4 Combined Branching-Advancing

Any combination of advancing and branching modes in a distributed space can be expressed and implemented in DSL (as shown in Fig. 7).



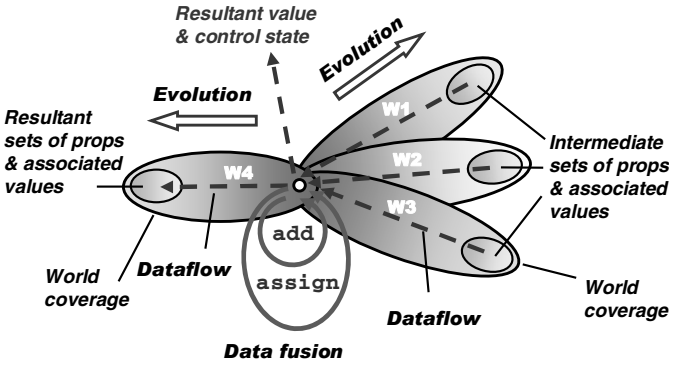
`advance(sequence(`*W1*`,`*W2*`,`*W3*`),`*W4*`)`

Fig. 7. Breadth–depth composition mode

These combinations, when embraced by the existing variety of composition rules, can provide any imaginable and even so far unimaginable spatial algorithms that can solve distributed problems in highly integral and compact ways, without explicit descending to the traditional atomistic level, the latter shifted to automatic implementation.

5.5 Operations on Remote Values

Due to fully recursive organization of DSL, it is possible to program in it arbitrary complex expressions directly operating not only on local but also arbitrarily remote values, where any programs (scenarios) can happen to be operands of any operations (expressed by rules). This gives an enormous expressive power and compactness to complex spatial scenarios evolving in distributed environments. An example of such compact expression of spatial operations on remote values and variables is shown in Fig. 8.




```
assign(W4,add(W1,W2,W3))
```

Fig. 8. Direct operations on remote values

6 Distributed Interpreter

The DSL interpreter (as from the previous language versions [1, 2, 4]) has been prototyped on different platforms. Its key features (see Fig. 9) are as follows:

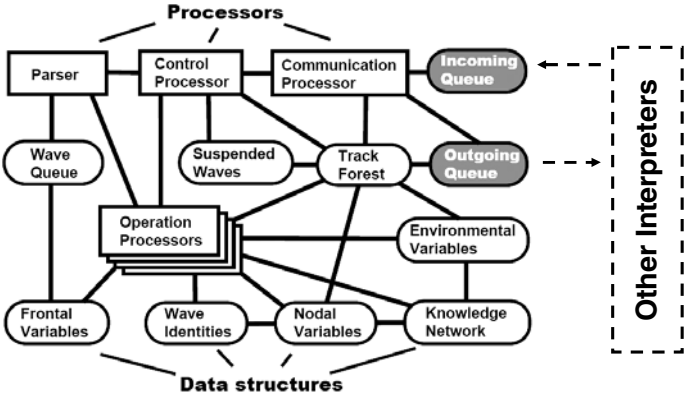


Fig. 9. Organization of DSL interpreter

- It consists of a number of specialized modules working in parallel and handling and sharing specific data structures supporting persistent virtual worlds and temporary hierarchical control mechanisms.
- The whole network of the interpreters can be mobile and open, changing at runtime the number of nodes and communication structure between them.
- The heart of the distributed interpreter is its *spatial track system*, with its parts kept in the Track Forest memory of local interpreters (Fig. 9); these being logically interlinked with such parts in other interpreter copies, forming altogether indivisible space coverage. This enables hierarchical command and control and remote data and code access, with high integrity of emerging parallel and distributed solutions, without any centralized resources.

- Copies of the interpreter can be concealed, as for acting in hostile systems, allowing us to impact them overwhelmingly (finding & eliminating unwanted infrastructures including).

The dynamically crated track trees (Fig. 10) spanning the systems in which DSL scenarios evolve (Fig. 10*a*), are used for supporting spatial variables and echoing and merging different types of control states and remote data, being self-optimized in the echo processes (Fig. 10*b*). They also route further waves to the positions in physical, virtual or combined spaces reached by the previous waves, uniting them with the frontal variables left there by preceding waves.

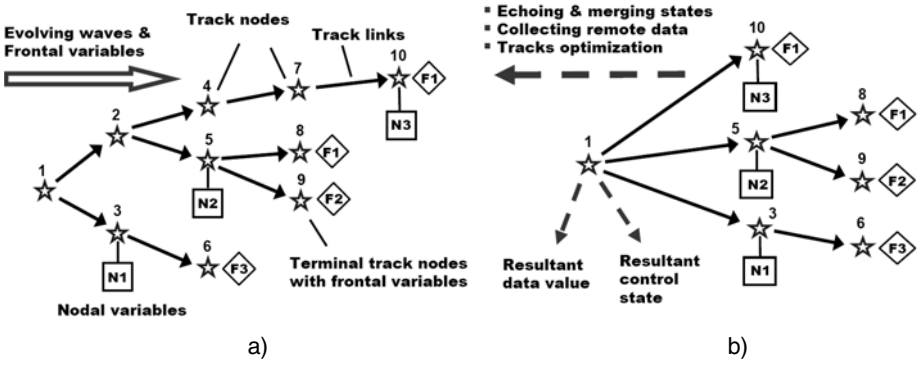


Fig. 10. Distributed track system. (a) forward operations; (b) backward operations.

7 Programming Examples

We will show here elementary examples of solution in DSL of some important problems on networks and graphs in a fully distributed way, where each node may reside in a separate computer.

7.1 Shortest Paths

The solution for finding the shortest path between two nodes (`a` and `e`) by navigating the network with parallel waves is shown in Fig. 11, and by the scenario that follows.

```
sequence(
 (hop(a); Ndist = 0; repeat(hop(alllinks);
  Fdist += LINK; or(Ndist == nil, Ndist > Fdist);
  Ndist = Fdist; Npred = BACK)),
 (hop(e); repeat(Fpath = NAME & Fpath; hop(Npred);
  USER = Fpath))
```

The result obtained in node `a` will be as `(a, b, d, e)`.

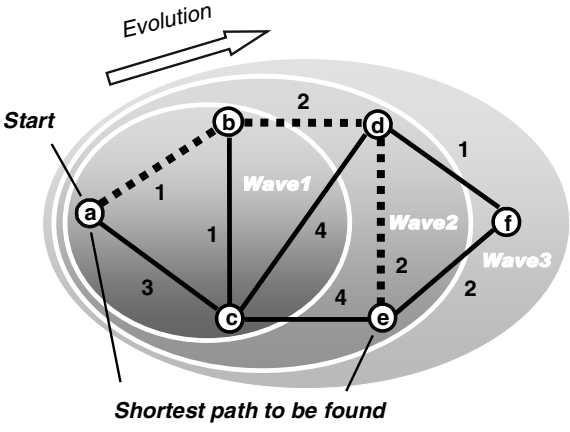


Fig. 11. Finding shortest path with waves

Many important problems of optimization and control may be expressed as finding shortest paths in a distributed solution space.

7.2 Topology Analysis

DSL allows us to directly analyze and process distributed topologies in a parallel and concise way. To find the weakest nodes in a network (like *articulation points*, see Fig. 12*a*) which, when removed, split it into disjoint parts, we only need the following program (resulting in node `d`).

```
hop(allnodes); IDENTITY = NAME; Mark = 1;
 and((random(hop(alllinks));
  repeat(grasp(Mark == nil; Mark = 1); hop(alllinks))),
 (hop(alllinks); Mark == nil), output(NAME))
```

Cliques (or fully connected sub-graphs of a graph, as in Figure 12*b*), on the contrary, may be considered as strongest parts of a system. They can be found in parallel by the following program, resulting for Figure 12*b* in: (`a`, `b`, `c`, `d`), (`c`, `d`, `e`), (`d`, `e`, `f`).

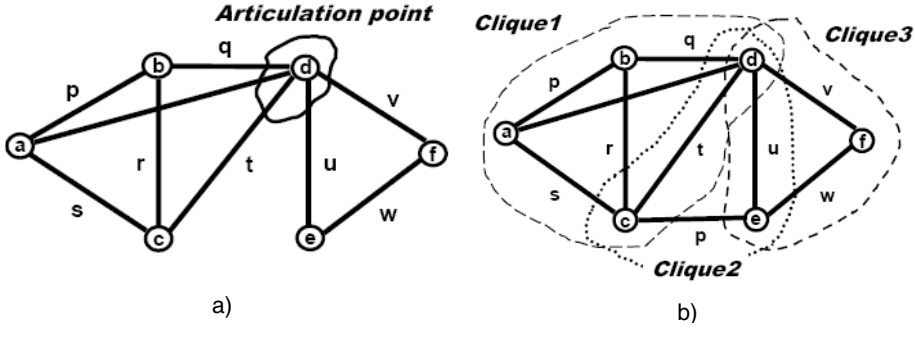


Fig. 12. Discovering distributed topological features. (a) articulation points; (b) cliques.

```
hop(allnodes); Fclique = CONTENT;
repeat(hop(alllinks); notbelong(CONTENT, Fclique);
 and(andparallel(hop(anylink, nodes(Fclique)); done),
  or((BACK > CONT; done), append(Fclique, CONT))));
output(Fclique)
```

The above programs work without any central resources, even if every node of the network is located in a different computer, with programs starting from any node [2]).

8 Collective Robotics

Installing DSL interpreter into mobile robots (ground, aerial, surface, or underwater, as in Fig. 13 for the first two) allows us to organize effective group solutions (incl. any swarming) of complex problems in distributed physical spaces in a concise way, shifting traditional management routines to automatic levels. Expressing explicit system behaviors at any levels can be easy too.

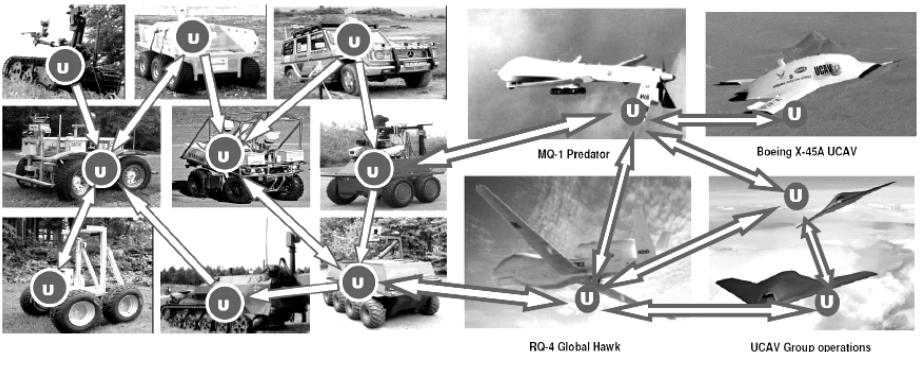


Fig. 13. Grouping unmanned ground and aerial vehicles

8.1 Task Level

Any groups of mobile robots (as in Fig. 13) can be tasked at a highest possible level, just telling what they should do together but without detailing how, and what are the duties of every unit, which may not be known in advance. An exemplary task here:

Go to physical locations of the disaster zone with coordinates (50.433, 30.633), (50.417, 30.490), and (50.467, 30.517). Evaluate damage in each location, find and transmit the maximum destruction value, together with exact coordinates of the corresponding location, to a management center.

The DSL program will be as follows:

```
transmit(max(
  move((50.433,30.633),(50.417,30.490),(50.467,30.517));
  append(evaluate(destruction), WHERE)))
```

Details of automatic implementation of this scenario by different numbers of mobile robots are discussed elsewhere [14].

8.2 Behavior Level

After embedding DSL interpreters into robotic vehicles, we can also provide any needed detailed collective behavior of them (at a lower than top task level, as before)- from *loose swarms* to a *strictly controlled integral unit* obeying external orders. Any mixture of different behaviors within the same scenario can be easily programmed too.

The following DSL scenario combines loose, random swarm movement in a distributed space with periodic finding/updating topologically central unit, and setting runtime hierarchical infrastructure between the units. The latter controls observation of distributed territory, collecting potential targets, distributing them between the vehicles, and then impacting potential targets by them individually. More on the implementation of this scenario can be found in [14, 15].

```
(hop(allnodes); Range = 500;
 Limits = (dx(0,8), dy(-2,5));
 repeat(Shift = random(Limits);
  if(empty(hop(Shift, Range), move(Shift)))),
(repeat(hop(Faver = average(hop(allnodes); WHERE);
  min(hop(allnodes); distance(Aver, WHERE)& ADDR):2));
  stay(hop(allnodes); remove(alllinks);
  Frange = 20; repeat(linkup(+infra, first, Frange));
  orpar(loop(nonempty(Fseen =
   repeat(free(detect(targets)), hoplinks(+infra));
   repeat(free(select_move_shoot(Fseen),
    hoplinks(+infra))), sleep(360)))
```

9 Other Applications

A variety of other potential applications of the proposed paradigm have been investigated and published [14-22]. We are mentioning here just two of them, currently in active research and assessment.

9.1 Robotized Armies

Distributed robotized systems are of rapidly growing importance in defense [23], where robotic swarming on asymmetric battlefields is becoming a major dimension of the new military doctrine for 21st century. But, as admitted in [23], swarming, along with its simple rules of individual behavior and fully distributed nature, agility, and ubiquity, may also result in unpredictability of behavior for both sides of the conflict.

The approach briefed in this paper, also investigated in previous publications on this paradigm, is very much in line with these modern trends. Moreover, we are offering a unified solution that can harness loosely coupled swarms, always guaranteeing their global-goal-driven behavior, where the watershed between loose swarming and strict hierarchical control may be situation dependent and changing over time.

These new doctrine trends will inevitably influence the role and sense of communications on battlefields, as with the planned drastic reduction of centralized C2 much more emphasis will be paid to intelligent tactical communications, where the scenario

mobility in networked systems, offered by the approach proposed, may constitute an effective solution, with the key points [17] as follows:

- Dramatic shift of global organization to intelligent tactical communications;
- Self-spreading and self-recovering mission scenarios and emergent command and control;
- Embedding intelligent protocol module into existing communication equipment;
- Situation-dependent watershed between global control and local communications.

In relation to the mentioned above, different (including new) types of command and control strategies for distributed robotized systems have been investigated in DSL [20].

9.2 Terrorism and Piracy Fight

It is no secret that the mightiest world armies with most powerful weapons and wonderful classical organizations are often losing to poorly armed terrorists and insurgents, also are powerless before the sea piracy, who are using very flexible, asymmetric tactics. A fresh piracy map is shown in Fig. 14, with piracy cases only for 2009 throughout the world, and possible information sources they are using, which form a sophisticated distributed network. The ideology and technology discussed in this paper can dynamically organize the whole world to withstand these activities, offering runtime (and ahead of it) spatial solutions, with high integrity and goal orientation—from global networks search to organizing smart unmanned swarms providing asymmetric responses to the asymmetric attacks. The approach can guarantee survivability of missions and fulfilling global and local objectives under any circumstances.

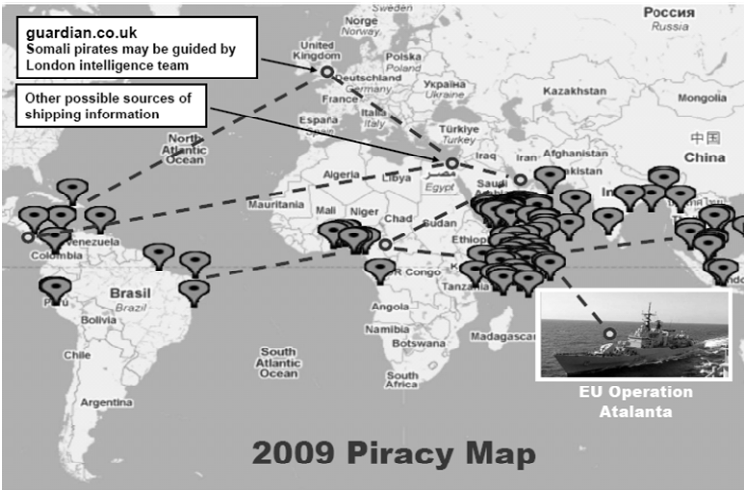


Fig. 14. Global piracy fight

There are a number of initiatives to utilize unmanned aircraft swarms to counter the threat of sea pirates, who also attack with swarming techniques [24]. The swarm will recon a wide region ahead of the chopper and converge to assault any emerging targets. The project seeks to make the swarm fully autonomous so that it can identify targets and choose which ones to attack. This would enable the vehicles and devices to act on their own, in an autonomous manner, based upon data sent from their own swarm or other swarms. An operational version of the network could help direct swarms of unmanned boats, helicopters or jet-powered aircraft.

All these initiatives can be effectively programmed and managed in DSL.

10 Comparison with the First Computers

The approach, offering an advanced system organization, can be compared with the invention of the first world computers [25] and first high-level programming languages [26]. In our case, this computer may not only operate with data stored in a localized memory, but can cover, grasp, and manage any distributed system, the whole world including, and can work not only with information but with physical matter or objects too.

If compared with the Turing computational model, instead of the head moving through tape in performing calculations, we have a recursive formula that unwraps, replicates, covers and matches the distributed world in parallel, scanning it forth and back, bringing operations and data directly to the points of their consumption, automatically setting distributed command and control infrastructures, and organizing local and global behaviors needed. The term "computer" first referred to the people who did scientific calculations by hand [27]. In the end, they were rewarded by a new electronic machine that took the place and the name of those who were, once, the computers.

We can draw the following symbolic parallel with this. Despite the overwhelming automation of human activity (in both information and matter processing) the world as a whole may still be considered as a human machine, as main decisions and global management still remain the human prerogative. With the approach offered, we can effectively automate this top-level human supervision, actually *converting the whole world into a universal programmable machine* spatially executing global scenarios in DSL or a similar language. Despite certain science fiction flavor of this comparison, we can find numerous applications for such a global approach, some mentioned above, where top level decision automation could withstand unwanted events and save lives, and where timely human reaction may not be possible, even in principle.

11 Conclusions

We have developed and tested a novel system approach, which can describe what the system should do and how to behave on a higher level, while delegating traditional management details (like partitioning into components, their distribution, interaction and synchronization) to the effective automatic layer.

A DSL scenario is not a usual program -- it is rather a recursive active spatial pattern dynamically matching structures of distributed worlds. It has a hierarchical organization, which is grasping, by means of spreading parallel waves, the whole of the system to be comprehended and impacted.

The DSL scenarios can also create, in a parallel and fully distributed way, active distributed worlds, which become persistent and operate independently. They may spatially intervene into operation of these and other worlds and systems, changing their structures and behaviors in the way required, also self-recover from indiscriminate failures and damages, as well as repair and recover the systems managed.

Prospective applications of this work can also be linked with economy, ecology and weather prediction—by using the whole networked world as a spatial supercomputer, self-optimizing its performance. Also, for terrorism and piracy fight, where the powerful parallel ability of analyzing distributed systems and finding strong and weak patterns in them (as shown in Section 7.2), as well as any structures, may be the key to global solutions.

Crises may emerge anywhere and anytime like, say, birds or swine flu or the current global economic disaster. We must be ready to react on them quickly and asymmetrically, withstanding and eradicating them -- in a "pandemic" way too, highly organized and intelligent, however.

We already have technical capabilities for this, as for example, the number of mobile phone owners in the world is approaching 3bn, and installing DSL interpreter in at least a fraction of them, can allow us to organize collective runtime (and ahead of it) response to any world events.

Acknowledgements. Special thanks to organizers of the international ICINCO conferences for their regular support of these ideas, and particularly to Joaquim Filipe, Marina Carvalho and Vitor Pedrosa.

References

1. Sapaty, P.S.: A Distributed Processing System. European Patent No. 0389655. European Patent Office (1993)
2. Sapaty, P.S.: Mobile Processing in Distributed and Open Environments. John Wiley & Sons, New York (1999)
3. Sapaty, P.S.: Over-Operability in Distributed Simulation and Control. The MSIAC's M&S Journal Online 4(2) (2002) (Winter Issue)
4. Sapaty, P.S.: Ruling Distributed Dynamic Worlds. John Wiley & Sons, New York (2005)
5. Minsky, M.: The Society of Mind. Simon and Schuster, New York (1988)
6. Smuts, J.C.: Holism and Evolution. Kessinger Publishing, LLC (2007)
7. Koffka, K.: Beiträge zur Psychologie der Gestalt. von F. Kenkel. Zeits. f. Psychol. 67 (1913)
8. Wertheimer, M.: Gestalt Theory. Erlangen, Berlin (1924)
9. Sapaty, P.: Gestalt-Based Ideology and Technology for Spatial Control of Distributed Dynamic Systems. In: Proc. International Gestalt Theory Congress, 16th Scientific Convention of the GTA. University of Osnabrück, Germany (2009)

10. Wilber, K.: Ken Wilber Online: Waves, Streams, States, and Self–A Summary of My Psychological Model (Or, Outline of An Integral Psychology). Shambhala Publications (2009)
11. Rogers, C.R.: Carl Rogers on Personal Power: Inner Strength and Its Revolutionary Impact. Trans-Atlantic Publications (1978)
12. Kriz, J.: Self-Actualization: Person-Centred Approach and Systems Theory. PCCS Books (2008)
13. Freud, S.: General Psychological Theory. Touchstone (1997)
14. Sapaty, P.S.: Providing Spatial Integrity for Distributed Unmanned Systems. In: Proc. 6th International Conference in Control, Automation and Robotics ICINCO 2009, Milan, Italy (2009)
15. Sapaty, P.: Remote Control of Open Groups of Remote Sensors. In: Proc. SPIE Europe Security + Defence, Berlin, Germany (2009)
16. Sapaty, P.: Distributed Capability for Battlespace Dominance. In: Electronic Warfare 2009 Conference & Exhibition, Novotel London West Hotel & Conference Center, London (2009)
17. Sapaty, P.: High-Level Communication Protocol for Dynamically Networked Battlefields. In: Proc. International Conference Tactical Communications 2009 (Situational Awareness & Operational Effectiveness in the Last Tactical Mile), One Whitehall Place, Whitehall Suite & Reception, London, UK (2009)
18. Sapaty, P.: Distributed Technology for Global Dominance. In: Suresh, R. (ed.) Proc. of SPIE. Defense Transformation and Net-Centric Systems, vol. 6981 (2008)
19. Sapaty, P.: Grasping the Whole by Spatial Intelligence: A Higher Level for Distributed Avionics. In: Proc. International Conference Military Avionics 2008, Cafe Royal, London (2008)
20. Sapaty, P., Morozov, A., Finkelstein, R., Sugisaka, M., Lambert, D.: A New Concept of Flexible Organization for Distributed Robotized Systems. Artificial Life and Robotics 12(1-2) (March 2008)
21. Sapaty, P.: Crisis Management with Distributed Processing Technology. International Transactions on Systems Science and Applications 1(1) (2006)
22. Sapaty, P., Sugisaka, M., Finkelstein, R., Delgado-Frias, J., Mirenkov, N.: Advanced IT Support of Crisis Relief Missions. Journal of Emergency Management 4(4) (July/August 2006)
23. Singer, P.W.: Wired for War: The Robotics Revolution and Conflict in the 21st Century. Penguin (2009)
24. Axe, D.: U.S. Military UAVs Experiment against Pirate Sea Attack Swarms. AUVSI's Unmanned Systems eBrief (26) (2009)
25. Rojas, R.: Konrad Zuse's Legacy: The Architecture of the Z1 and Z3. IEEE Annals of the History of Computing 19(2) (1997)
26. Zuse, K.: Uber den Plankalk, als Mittel zur Formulierung Shematisch Kombinativer Aufgaben. Archiv Mathematik, Band I (1948)
27. Grier, D.A.: When Computers Were Human. Princeton University Press, Princeton (2005)

Part I

Intelligent Control Systems and Optimization

An Improved Mean-shift Tracking Algorithm Based on Adaptive Multiple Feature Fusion

Hongpeng Yin[1], Yi Chai[2], Simon X. Yang[3], and David K.Y. Chiu[3]

[1] College of Automation, Chongqing University, Chongqing City, 400038, China
yinhongpeng@gmail.com
[2] State Key Laboratory of Power Transmission Equipment & System Security and New Technology, College of Automation, Chongqing University, Chongqing City, 400038, China
chaiyi@cqu.edu.cn
[3] College of Physical & Engineering Science, University of Guelph
Guelph, Ontario, N1G 2W1, Canada
syang@uoguelph.ca, dchiu@cis.uoguelph.ca

Abstract. In this paper an improved Mean-shift tracking algorithm based on adaptive multiple feature fusion is presented. A two-class variance ratio is employed to measure the discriminate between object and background. The multiple features are Fused by linear weighting to realise Mean-shift tracking using the discrimination. Furthermore, an adaptive model updating mechanism based on the likelihood of the features between successive frames is addressed to alleviate the mode drifts. Based on biology vision theory ,colour, edge and texture cue are employed to implement the scheme. Experiments on several video sequences show the effectiveness of the proposed method.

1 Introduction

Visual tracking is a common task in computer vision and play key roles in many scientific and engineering fields. Various applications ranging from video surveillance, human computer interaction, traffic monitoring to video analysis and understanding, all require the ability to track objects in a complex scene. Many powerful algorithms for target tracking have yielded two decades of vision research.

Mean shift algorithm is a nonparametric kernel density estimator based on the colour kernel density estimation, which has recently gained more attention due to its low computational complexity and robustness. Fukunaga and Hostetler first proposed the mean shift algorithm [1] for clustering data in 1975, and Cheng [2] introduced it to the image-processing community a decade ago. Mean shift algorithm to image segmentation and tracking [7,8,9,11,12] is proposed by Commanicui et al. Later, many variants of the mean shift algorithms were proposed for different applications [10,11,12,13,14,15].

Frame difference and adaptive background subtraction combined with simple data association techniques can effectively track in real-time for stationary cameras target tracking [3,4,5]. Optical flow methods using the pattern of apparent motion of objects, surfaces and edges in a visual scene caused by the relative motion between the camera and scene. These methods can achieve the target tracking in the stationary cameras scene and the mobile cameras scene [1,6]. Modern appearance-based methods using

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 49–62.
springerlink.com

the likelihood between the tracked target appearance describe model and the reference target appearance describe model can achieve the target tracking without prior knowledge of scene structure or camera motion. Modern appearance-based methods include the use of flexible template models [2,7] and kernel-based methods that track nonrigid objects used colour histograms [8,9,10]. Particle filter and Kalman filter are using to achieve more robust tracking of manoeuvring objects by introducing statistical models of object and camera motion [9,11,12,13].

The major difficulty in target tracking based on computer vision is the variation of the target appearance and its background. By using a stationary camera, the background in a long image sequence is dynamic. However, the performance of a tracker can be improved by using an adaptive tracking scheme and multiple features. The basic ideal is online adaptive selection of appropriate features for tracking. Recently, several adaptive tracking algorithms[14,15,16,17,18] were proposed. Collins et al. [14] proposed to online select discriminative tracking features from linear combination of RGB values. The two-class variance ratio is used to rank each feature by how well it separates the sample distributions of object and background. Top N features that have the greatest discrimination are selected to embed in a mean-shift tracking system. This approach,however only considers the RGB colour features. Actually, this approach is one feature-based tracking. Furthermore, it lacks an effective model update method to copy with the model drifts. Liang et al. [17] extended the work of Collins et al. by introducing adaptive feature selection and scale adaptation. A new feature selection method based on Bayes error rate is proposed. But how to deal with the model drifts is not addressed in this paper. He et al. [16] used a clustering method to segment the object tracking according to different colours, and generate a Gaussian model for each segment respectively to extract the colour feature. Then an appropriate model was selected by judging the discrimination of the features. The Gaussian model however, not always fit each segment in practice. Recently, Wang and Yagi [15] extended the standard mean-shift tracking algorithm to an adaptive tracker by selecting reliable features from RGB, HSV, normalised RG colour cues and shape-texture cues, according to their descriptive ability. But only two best discriminate features are used to represent the target. It dose not use fully all the features information it has computed and has a high time complexity.

A key issue addressed in this work is an online, adaptive multiple-feature fusion and template-update mechanism for target tracking. Based on the theory of biologically visual recognition system, the main visual information comes from the colour feature, edge feature and the texture feature [19,20,21]. In this paper, RGB colour features, Prewitt edge feature and local binary pattern (LBP) texture feature are employed to implement the scheme. Target tracking is considered as a local discrimination problem with two classes: foreground and background. Many works have point the features that best discriminate between object and background are also best for tracking performance [14,19,20]. In this paper the tracked target is represented by a fused feature. According to the discriminate between object and background measured by two-class variance ratio, the multiple features are combined by linear weighting to realise kernel-based tracking. As model drafts, better performance could be achieved by using a novel up-dating strategy that takes into account the similarity between the initial and current appearance of the target. Each feature's similarity is computed. The high similarity

features are given a big weight and the low similarity features are given a small weight. A good feature for tracking is a steady feature across the consecutive frames. The target update model is updated by re-weighting the multiple features based on the similarity between the initial and current appearance of the target. The proposed approach is shown as Figure 1.

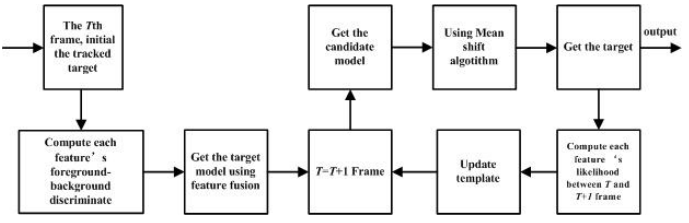


Fig. 1. The Flow chart of the purposed approach

The paper is organised as follows. Section 2 presents a brief introduction the feature subset selection. And Section 3 presents our proposed approach for target tracking. Computer simulation and results compared with related work are presented in Section 4.Concluding remarks are given in Section 5.

2 Feature Subset Selection

It is important to decide what kinds of features are used before constructing the feature fusion mechanism. In [14], the set of candidate features is composed of linear combination of RGB pixel. In [15], colour cue and shape-texture cues are employed to describe the model of the target. In this paper based on the theory of biologically visual recognition system [19,20,21], RGB colour features, Prewitt edge feature and local binary pattern (LBP) texture feature are employed to implement the scheme.

2.1 RGB Colour Feature

Colour information is an important visual feature. That is robust to the target rotary, non-rigid transformation and target shelter, widely used in the appearance model-based visual application. In this paper, colour distributions are represented by colour histograms, and RGB colour space as a very common colour space is used in this paper. The R, G, and B channels are quantised into 256 bins, respectively. The colour histogram, calculated using Epanechnikov kernel, is applied [8,9].

2.2 Prewitt Edge Feature

The edge information is the most fundamental characteristic of images. It is also included useful information for target tracking. There are many methods for edge detection, but most of them can be grouped into two categories: search-based and zero-crossing based. The search-based methods detect edges by first computing a measure

of edge strength, usually a first-order derivative expression, such as the gradient magnitude; and then searching for local directional maxima of the gradient magnitude using a computed estimate of the local orientation of the edge, usually the gradient direction [22,23]. The zero-crossing based methods search for zero crossings in a second-order derivative expression computed from the image, in order to find edges, usually the zero-crossings of the Laplacian or the zero-crossings of a non-linear differential expression [23].

In this paper, Prewitt operator is employed to get the edge feature. For its low computational complexity and high performance. Prewitt operator has two convolution kernels as shown in Figure 2. Images of each point are used for the convolution kernel, the first kernel usually corresponding to the largest vertical edge, and the second corresponding to the largest horizontal edge. The maximum values of each point convoluted with the two kernels are accepted. Convolution is admitted as output value; results of operations are edge images. The Prewitt operator can be defined as:

$$S_p = \sqrt{d_x^2 + d_y^2}, \tag{1}$$

$$\begin{aligned} d_x = &[f(x-1,y-1) + f(x,y-1) + f(x+1,y-1)] - \\ &[f(x-1,y+1) + f(x,y+1) + f(x+1,y+1)], \end{aligned} \tag{2}$$

$$\begin{aligned} d_y = &[f(x+1,y-1) + f(x+1,y) + f(x+1,y+1)] - \\ &[f(x-1,y-1) + f(x-1,y) + f(x-1,y+1)]. \end{aligned} \tag{3}$$

The histogram is used to represent the edge feature. Prewitt edge is also quantised into 256 bins Epanechnikov kernel like colour feature.

1	1	1
0	0	0
-1	-1	-1

1	0	-1
1	0	-1
1	0	-1

Fig. 2. The two convolution kernel of Prewitt

2.3 LBP Texture Feature

Local Binary Patterns (LBP) is basically a fine-scale descriptor that captures small texture details. It is also very resistant to lighting changes. LBP is a good choice for coding fine details of facial appearance and texture [24,25,26]. The Local Binary Patterns operator is introduced as a means of summarising local gray-level structure by Ojiala in 1996 [26]. The operator takes a local neighbourhood around each pixel, thresholds the pixels of the neighbourhood at the value of the central pixel, and uses the resulting

binary-valued image patch as a local image descriptor. It was originally defined for 3*3 neighbourhoods, given 8 bit codes based on the 8 pixels around the central one. Formally, the LBP operator takes the form

$$LBP(x_k, y_k) = \sum_{n=0}^{7} 2^n S(i_n - i_k), \tag{4}$$

$$S(u) = \begin{cases} 1, & x \geq th, \\ 0, & u < th, \end{cases} \tag{5}$$

where in this case n runs over the 8 neighbours of the central pixel k, i_k and i_n are the gray-level values at k and n, and $S(u)$ is 1 if $x \geq th$ and 0 otherwise. The LBP encoding process is illustrated in Figure 3.

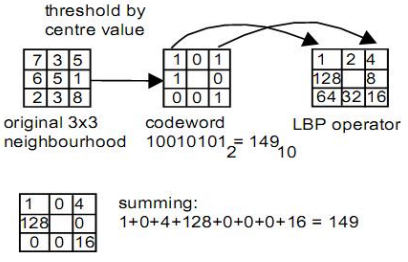


Fig. 3. Illustration of the basic LBP operator

In methods that turn LBPs into histograms, the number of bins can be reduced significantly by assigning all non-uniform patterns to a single bin, often without losing too much information. In this paper, it is quantised into 256 bins with Epanechnikov kernel.

3 Online, Adaptive Features Fusion Method for Tracking

There are two main components in this approach: the online adaptive features fusion based on discrimination criterion function, and the kernel-based tracking, which is used to track targets, based on the fused feature.

3.1 Features Fusion Method

In this paper, the target is represented by a rectangular set of pixels covering the target, while the background is represented by a larger surrounding ring of pixels. Given a feature f, let $H_{fg}(i)$ be a histogram of target and $H_{bg}(i)$ be a histogram for the background. The empirical discrete probability distribution $p(i)$ for the object and $q(i)$ for the background, can be calculated as $p(i) = H_{fg}(i)/n_{fg}$ and$q(i) = H_{bg}(i)/n_{bg}$, where n_{fg} is the pixel number of the target region and n_{bg} the pixel number of the background. The weight histograms represent the features only. It does not reflect the

descriptive ability of the features directly. A log-likelihood ratio image is employed to solve this problem [14,15]. The likelihood ratio nonlinear log likelihood ratio maps feature values associated with the target to positive values and those associated with the background to negative values. The likelihood ratio of a feature is given by

$$L(i) = max(-1, min(1, log(\frac{max(p(i), \epsilon)}{max(q(i), \epsilon)}))), \tag{6}$$

where ϵ is a very small number(set in 0.001 in this work), that prevents dividing by zero or taking the log of zero. Likelihood ratio images are the foundation for evaluating the discriminative ability of the features in the candidate features set. Figure 4 shows the likelihood ratio images of different features.

In the practice, the whole weighted images weighted by log likelihood are not needed to be calculated for the computational complexity. The corresponding variance is employed to measure the separately between target and background classes. Using the method in [14,15], based on the equality $var(x) = E[x^2] - (E[x])^2$, the variance of the log likelihood is computed as

$$var(L : p) = E[L(i)^2] - (E[L(i)])^2. \tag{7}$$

The discriminative ability of each feature is calculated by the variance ratio. The hypothesis in this paper is that the features that best discriminate between target and background are also best for tracking the target. So, as the target's features describe model, a highest weight is given to the best discriminate feature, and the less discriminate feature has a smaller contribution. Based on the discrimination criterion function , the target features describe model can be calculated as

$$p_t(i) = \sum_{k=1}^{n} \lambda_k p_k(i), \tag{8}$$

$$\lambda_k = \frac{var_k(L : p)}{\sum_{k=1}^{n} var_k(L : p)}, \tag{9}$$

where $p_k(i)$ is the feature $K's$ probability distribution model, $\lambda_k$is the weight and $\sum_{k=1}^{n} \lambda_k = 1$. Figure 5 shows the fusion of the five features image.

3.2 Kernel-Based Tracking

Mean shift is a nonparametric kernel density estimator, which, based on the colour kernel density estimation, has recently gained more attention due to its low computational complexity and robustness to appearance change, however, the basic mean shift tracking algorithm assumes that the target representation is discriminative enough against the background. This assumption is not always true, especially when tracking is carried out in a dynamic background [9,10]. An online, adaptive features fusion mechanism is embedded in the kernel-based mean shift algorithm for effective tracking. Due to the continuous nature of video, the distribution of target and background features in the current frame should remain similar to the previous frame and the fused feature model

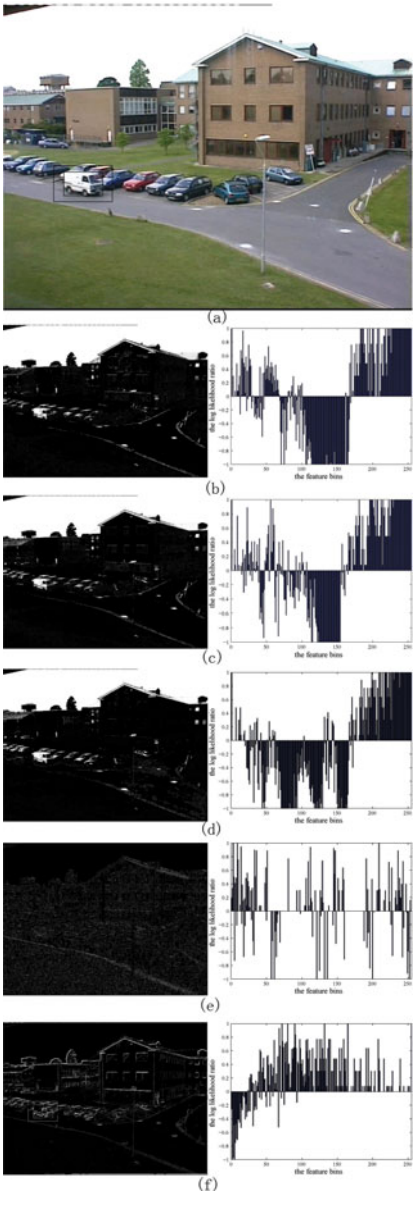


Fig. 4. Different feature image after likelihood ratio process. (a) The input frame, (b) feature R and likelihood ratio, (c) feature G and likelihood ratio, (d) feature B and likelihood ratio, (e) LBP texture feature and likelihood ratio, (f) Prewitt edge feature and likelihood ratio.

should still be valid. The initial position of the target is given by $y0$ which is determined in the previous frame. The target model is $P = p_{t,t=1\ldots m}$, $\sum_{t=1}^{m} p_t = 1$, and the candidate target model is $P(y_0) = p_t(y_0)_{t=1\ldots m}$, $\sum_{t=1}^{m} p_t = 1$, where p_t is the fused feature

Fig. 5. The image after fusion

model. The Epanechnikou profile [8, 9] is employed in this paper. The target's shift vector form y_0 in the current frame is computed as

$$y_1 = \frac{\sum_{i=1}^{n_h} X_i \omega_i g(\| \frac{y_0 - X_i}{h} \|^2)}{\sum_{i=1}^{n_h} \omega_i g(\| \frac{y_0 - X_i}{h} \|^2)}, \tag{10}$$

where $g(x) = -k'(x)$,$k(x)$ is Epanechnikou profile, h is bandwidth and $\omega_i$can compute as

$$\omega_i = \sum_{t=1}^{m} \sqrt{\frac{P}{P(y_0)}} \delta[b(x_i) - t]. \tag{11}$$

And the tracker assigns a new position of the target by using

$$y_1 = \frac{1}{2}(y_0 + y_1). \tag{12}$$

If$\|y_0 - y_1\|$, the iteration computation stops and y_1 is taken as the position of the target in the current fame. Otherwise let $y_0 = y_1$, then using Eq. (10) get the shift vector and do position assignment using Eq. (12). From Eq. (8) and Eq. (10), the pixels' weight is assigned by two parts. One is the kernel profile, which gives high weight to the pixel nearly to the centre. The other one is the discriminative ability of each feature. Higher weight is given to the higher discriminative ability feature.

3.3 Template Update Mechanism

It is necessary to update the target model, because the variation of the target appearance and its background. When the target appearance or the background changes, the fixed target model can not accurate describe the target, so it can not obtain the right position of target. But using an inaccurately tracking result to update the target model may lead to the wrong update of the target model. And with the error accumulate, it finally results in track failure.

In order to alleviate the mode drifts, an adaptive model update mechanism based on the likelihood of the features between successive frames is proposed in this paper.

During the initialisation stage, the target is obtained by a hand-draw rectangle and the target model is computed by the fusion method introduced in the previous subsection. The fused target model is used for tracking in the next frame and is also kept to use in subsequent model updates. Fellowing the method in [15], the updated target model M can be computed as

$$M = (1 - L_{ic})M_i + L_{ic}M_c, \tag{13}$$

$$L_{ic} = \sum_{u=1}^{m} \sqrt{M_i M_c}, \tag{14}$$

where L_{ic} is the likelihood between the initial model and current model measured by Bhattacharyya coefficient [8,9]; M_i is the initial model; M_c is the current target model computed as

$$M_c = (1 - L_{pc})M_p + L_{pc}M_a, \tag{15}$$

where M_p is the previous target model, M_a is the current target fused-feature introduced in section 3, and L_{pc} is computed the likelihood between the M_a and M_p. The proposed updating method considers temporal coherence by weighing the initial target model, previous target model and current candidate. It can be more robust for the target appearance and the background change.

4 Experiments

To illustrate the benefits of the proposed approach, experiments on various test video sequences using the proposed approach and other algorithms are conducted. The experiment was done using Pentium core 1.8G, Win XP, Matlab 7.0. The Epanechnikov profile was used for histogram computations. The RGB colour feature, LBP texture feature and Prewitt edge feature were taken as feature space and it was quantised into 256 bins. The public dataset with ground truth [27] is used to test the proposal. The tracking results are compared with the basic mean-shift [8], fore/background ratio, variance ratio, peak difference [9], and multiple feature [15] tracker. The initialisation of the tracking is given by a hand-draw rectangle. The same initialisations are given to all the trackers. In this experiment, sequences EgTest01, EgTest02, EgTest03, EgTest04 and EgTest05 in the database are chosen. This challenging tracking sequences is made by various factors, such as different viewpoints, illumination changes, reflectance variations of the targets, similar objects nearby, and partial occlusions.The Tracking results of sequences EgTest01, EgTest02, EgTest03, EgTest04 and EgTest05 as Fig.6, Fig.7, Fig.8, Fig.9 and Fig.10 respectively. While the red rectangle is the tracking window.

The tracking success rate is the most important criterion for target tracking, which is the number of successful tracked frames divided by the total number of frames. The bounding box that overlaps the ground truth can be considered as a successful track. To demonstrate the accuracy of tracking, the average overlap between bounding boxes (Avg overlap BB) and average overlap between bitmaps within the overlapping bounding box area (Avg overlap BM) are employed. Avg overlap BB is the percentage of the overlap between the tracking bounding box and the bounding box identified by ground

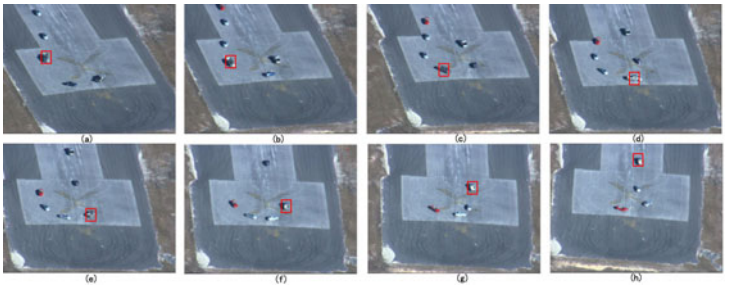


Fig. 6. The tracking result of sequences EgTest01 using proposed algorithm. (a) The initial frame, (b) The 50th frame tracking result, (c) The 100th frame tracking result, (d) The 150th frame tracking result, (e) The 200th frame tracking result, (f) The 250th frame tracking result, (g) The 300th frame tracking result, (h) The 350th frame tracking result.

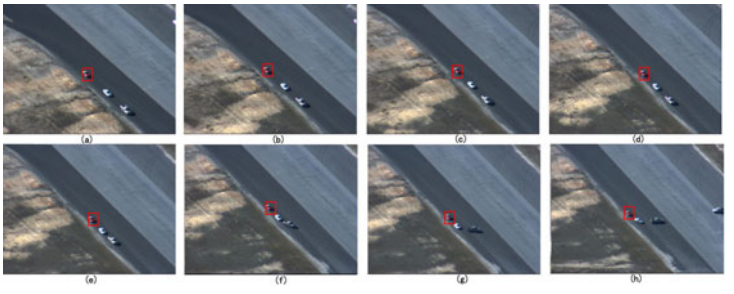


Fig. 7. The tracking result of sequences EgTest02 using proposed algorithm. (a) The initial frame, (b) The 50th frame tracking result, (c) The 100th frame tracking result, (d) The 200th frame tracking result, (e) The 300th frame tracking result, (f) The 400th frame tracking result, (g) The 500th frame tracking result, (h) The 600th frame tracking result.

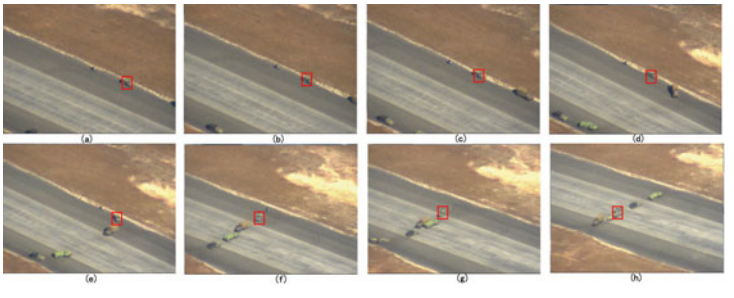


Fig. 8. The tracking result of sequences EgTest03 using proposed algorithm. (a) The initial frame, (b) The 50th frame tracking result, (c) The 100th frame tracking result, (d) The 200th frame tracking result, (e) The 300th frame tracking result, (f) The 400th frame tracking result, (g) The 500th frame tracking result, (h) The 600th frame tracking result.

truth files. Avg overlap BM is computed in the area of the intersection between the user bounding box and the ground truth bounding box. The comparison results are shown in Table 1.

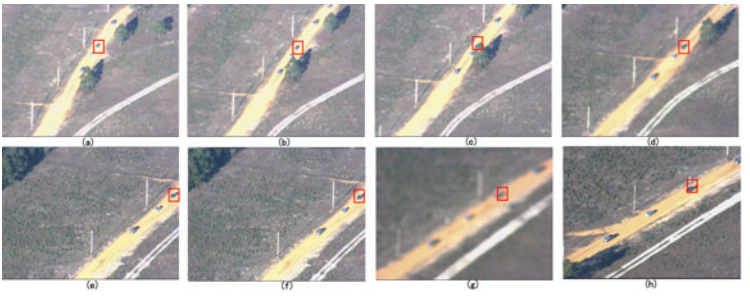


Fig. 9. The tracking result of sequences EgTest04 using proposed algorithm. (a) The initial frame, (b) The 50th frame tracking result, (c) The 100th frame tracking result, (d) The 200th frame tracking result, (e) The 300th frame tracking result, (f) The 400th frame tracking result, (g) The 500th frame tracking result, (h) The 600th frame tracking result.

Fig. 10. The tracking result of sequences EgTest05 using proposed algorithm. (a) The initial frame, (b) The 50th frame tracking result, (c) The 100th frame tracking result, (d) The 150th frame tracking result, (e) The 200th frame tracking result, (f) The 250th frame tracking result, (g) The 300th frame tracking result, (h) The 350th frame tracking result.

From the comparison results that show the successful tracking ratio, the proposed tracker gives the best results in five of the test sequences. The basic mean-shift tracker dose not have a good performance in EgTest01, EgTest02 and EgTest05, because the basic mean-shift tracker dose not use the multi-features information and lacks adaptive strategy.

Although an adaptive strategy is employed in Collins's approach, it dose not have good performance in EgTest01, EgTest02 and EgTest03. The peak difference algorithm has a better performance in the first sequences. The others sequences, however, do not demonstrate a good performance either. The multi-features methods that Integrate Colour and Shape-Texture features have a higher performance in all the sequences. Although the proposed approach has the best performance than the other trackers. But in EgTest03, EgTest04 and EgTest05 the successful tracking ratio is only 25.30%, 12.03% and 24.31%. The failed tracking examples are show as Figure 11.The main reason leading to tracking failure includes the similar feature distribution nearby as (a) in Fig.11, the lower discrimination between foreground and background as (b) in Fig.11 and long time occlusions as (c) in Fig.11.

Table 1. Tracking performance of different Algorithm. (a)EgTest01. (b)EgTest02. (c)EgTest03. (d)EgTest04. (e)EgTest05.

Criterion	MeanShift	FgBgRatio	VarianceRatio	PeakDiff	Multi-feature	The Proposed
Successful ratio	17.58%	100%	29.12%	100%	100%	100%
Avg overlap BB	65.50%	62.87%	76.87%	61.76%	61.62%	74.28%
Avg overlap BM	66.26%	49.15%	61.30%	57.76%	68.38%	76.56%

(a)

Criterion	MeanShift	FgBgRatio	VarianceRatio	PeakDiff	Multi-feature	The Proposed
Sucessful ratio	39.23%	39.23%	27.69%	30.77%	100%	100%
Avg overlap BB	91.09%	89.13%	85.19%	90.54%	93.32%	94.21%
Avg overlap BM	74.69%	66.98%	73.32%	65.91%	72.70%	73.53%

(b)

Criterion	MeanShift	FgBgRatio	VarianceRatio	PeakDiff	Multi-feature	The Proposed
Sucessful ratio	20.62%	17.90%	12.06%	12.06%	20.23%	25.30%
Avg overlap BB	86.96%	87.01%	93.74%	92.27%	88.66%	90.11%
Avg overlap BM	66.65%	54.04%	70.79%	67.20%	69.37%	71.23%

(c)

Criterion	MeanShift	FgBgRatio	VarianceRatio	PeakDiff	Multi-feature	The Proposed
Sucessful ratio	9.84%	8.74%	9.84%	3.83%	9.84%	12.03%
Avg overlap BB	66.78%	67.92%	66.03%	63.60%	69.52%	71.00%
Avg overlap BM	59.70%	52.75%	66.74%	66.42%	56.34%	68.43%

(d)

Criterion	MeanShift	FgBgRatio	VarianceRatio	PeakDiff	Multi-feature	The Proposed
Sucessful ratio	13.64%	13.64%	13.64%	13.64%	21.22%	24.31%
Avg overlap BB	94.58%	88.75%	86.46%	86.98%	72.65%	75.41%
Avg overlap BM	84.02%	71.12%	85.12%	69.90%	64.45%	70.03%

(e)

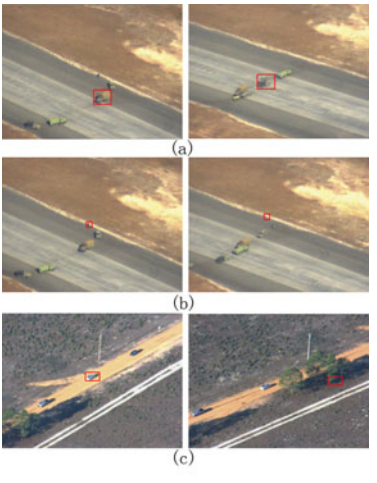


Fig. 11. The failure tracking examples of the proposed approach. (a) and (b) show the failure frame in EgTest03, (c) show the failure result in EgTest04.

For accuracy of tracking, the proposed tracking algorithm is not the best in some of the sequences. There is not obvious correlation between the tracking accuracy criterion and the tracking successful ratio.The proposed approach does not have the highest accuracy, because in most frame-sequences,the background and the target are not always separated accurately .

5 Conclusions

An online, adaptive multiple features fusion and template update mechanism for kernel-based target tracking is addressed in this paper. Based on the theory of biologically visual recognition system, RGB colour features, Prewitt edge feature and local binary pattern (LBP) texture feature are employed to implement the scheme. Experiment results show that the proposed approach is effective for target tracking. The comparison with other algorithms proved that the proposed approach has the best performance for target tracking.

Acknowledgements. This work is supported by China Scholarship Council. This work also supported by Chongqing University, ”211 Project” three creative talents training plan construction projects (S-09108).

References

1. Nir, T., Alfered, M.: Over-Parameterized Variational Optical Flow. International Journal of Computer Vision 76, 205–216 (2008)
2. Junqiu, W., Yasushi, Y.: Integrating Color and Shape-Texture Features for Adaptive Real-Time Object Tracking. IEEE Transactions on Image Processing 17, 235–240 (2008)
3. Collins, R., Lipton, A., Fujiyoshi, H., Kanade, T.: Algorithms for Cooperative Multi-Sensor Surveillance. Proceedings of the IEEE 89, 1456–1477 (2001)
4. Bar, Y., Fortmann, T.: Tracking and Data Association. Academic Press, London (1988)
5. Stauffer, C., Grimson, W.E.L.: Learning Patterns of Activity Using Real-Time Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 17, 814–820 (1995)
6. Barron, J.L., Fleet, D.J.: Performance of optical flow techniques. International Journal of Computer Vision 12, 43–77 (1994)
7. Mattews, I., Iashikawa, T., Baker, S.: The Template Update Problem. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 810–815 (2004)
8. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 603–619 (2002)
9. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 564–577 (2003)
10. Li, Z., Tang, Q., Sanq, N.: Improved meanshift algorithm for occlusion pedestrian tracking. Electronics Letters 44, 622–623 (2008)
11. Pan, J., Hu, B., Zhang, J.: Robust and Accurate Object Tracking Under Various Types of Occlusions. IEEE Transactions on Circuits and Systems for Video Technology 12, 223–236 (2008)
12. Chang, W.-Y., Chen, C.-S., Jian, Y.-D.: Visual Tracking in High Dimensional State Space by Appearance-Guided Particle Filtering. IEEE Transactions on Image Processing 17, 1154–1167 (2008)

13. Maggio, E., Smerladi, F., Cavallaro, A.: Adaptive Multifeature Tracking in a particle Filtering Framework. IEEE Transactions on Circuits and Systems for Video Technology 12, 1348–1359 (2007)
14. Collins, T., Liu, Y., Leordeanu, M.: Online Selection of Discriminative Tracking Features. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1631–1643 (2005)
15. Wang, J., Yagi, Y.: Integrating Color and Shape-Texture Features for Adaptive Real-Time Object Tracking. IEEE Transactions on Image Processing 17, 235–240 (2008)
16. He, W., Zhao, X., Zhang, L.: Online Feature Extraction and Selection for Object Tracking. In: Proceeding of IEEE Conference on Mechatronics and Automation, pp. 3497–3502 (2007)
17. Liang, D., Huang, Q.: Mean-Shift Blob Tracking with Adaptive Feature Selection and Scale Adaptation. In: IEEE Conference on Image Processing, vol. 3, pp. 369–372 (2007)
18. Yin, Z., Poriki, F., Collins, T.: Likelihood Map Fusion for Visual Object Tracking. In: IEEE Workshop on Applications of Computer Vision, pp. 1–7 (2008)
19. Thomas, S., Gabriel, K.: A Quantitative Theory of Immediate Visual Recognition. Programe of Brain Research 165, 33–56 (2007)
20. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A Biologically Inspired System for Action Recognition. In: IEEE 11th International Conference on Computer Vision, vol. 14, pp. 1–8 (2007)
21. Bar, M., Kassam, S.: Top-down facilitation of visual recognition. Proceedings of the National Academy of Sciences 103, 449–454 (2006)
22. Yong, Y., Croitoru, M.: Nonlinear multiscale wavelet diffusion for speckle suppression and edge enhancement in ultra sound images. IEEE Transactions on Medical Imaging 25, 297–311 (2006)
23. Sharifi, M., Fathy, M., Mahmoudi, T.: A Classifed and Comparative Study of Edge Detection Algorithms. In: International Conference on Information Technology:Coding and Computing, pp. 117–120 (2002)
24. Tan, X., Triggs, B.: Fusing Gabor and LBP Feature Sets for Kernel-Based Face Recognition. Analysis and Modeling of Faces and Gestures, 235–249 (2007)
25. Aroussi, E., Amine, A., Ghouzali, S.: Combining DCT and LBP Feature Sets For Efficient Face Recognition. In: ICTTA 2008, pp. 1–6 (2008)
26. Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recognition 29, 51–59 (1996)
27. Collins, T., Zhou, X.: An open source tracking testbed and evaluation website. Presented at the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (2005)

Modelling and Performance Enhancement of a Linear Actuation Mechanism Using Conducting Polymers

Elise T. Burriss[1], Gursel Alici[2], Geoffrey M. Spinks[1,2], and Scott McGovern[1]

[1] Intelligent Polymer Research Institute
[2] School of Mechanical, Materials and Mechatronic Engineering
University of Wollongong, 2522, NSW, Australia
(etb984,gursel,gspinks,scottmg}@uow.edu.au

Abstract. In this paper, we report on our investigation into modelling and performance enhancement of a linear actuation system based on cantilevered-type conducting polymer actuators, which can operate in air and aqueous media. We have employed the model to predict the linear displacement and force output of the actuation system, and to determine the optimum values of the system design parameters. The linear actuation system is a five bar parallel mechanism, articulated with two polymer actuators. Kinematic and force analyses of the mechanism including numerical results are presented, and its payload handling ability was experimentally evaluated. The experimental results prove that it is possible to generate an accurate linear movement and a corresponding rectilinear force from this mechanism. This mechanism can be employed as a motion and force transmission mechanism, which not only has a light weight, but also consumes a small electrical power.

Keywords: Electroactive Polymer Actuators, System Optimization, Linear Actuation, Kinematic Design.

1 Introduction

Conducting polymers are favourable as actuators due to their low actuation voltage, high force output relative to their weight, good strain properties, light weight, simple structure and silent motion. The common materials used for actuators are Ionic Polymer Metal Composite (IPMC), and electroactive polymers (EAPs) such as Polypyrrole (PPy) and Polyaniline (PANI) [1]. Disadvantages of polymer actuators include low speed of response, cyclic fatigue and non-linearity, which increases difficulty in modelling and controlling the actuator [2-3].

Bending actuators have been successfully used in applications where a nonlinear motion is required, such as a fin in an artificial fish [4]and robotic fingers [5]. However, the bending motion may limit the control of actuators along a linear path. Conducting polymers have been used as actuators in applications such as medical devices, toys, digital camera accessories and artificial muscles for robots as developed by Eamex [3]. Various PPy linear actuator designs have been investigated, including

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 63–78.
springerlink.com

were connected in parallel and immersed in a $LiClO_4$ aqueous solution, were able to produce a displacement of 60% of the actuator length in the longitudinal direction of the actuator. The use of linear actuators in air may provide an alternative when encapsulation of the electrolyte is not feasible. Yamakita et. Al [7] have developed an IPMC linear actuator for use in a biped walking robot. PPy actuators are used in applications where the oxidised or reduced position is maintained under a constant voltage, for which IPMC is not suitable as it will return to its original position. To overcome this limitation of IPMC, Thompson [1] adopted the design by Yamakita et al. using polypyrrole film as the active layers of a multi-layer actuator structure. Two methods of connecting the PPy film together to produce the linear design were employed by Thompson: (i) masking part of the film during PPy growth to create a flexible, electrically conductive hinge; (ii) electrically connecting the film with Cu tape, held in place with NdFeB magnets. This study implements the design investigated by Thompson using the second method of connecting the film together, as the method of masking allows the actuator to produce a clover-like shape under actuation, which impedes control and stability of the actuator. However, limitations arising through the use of Cu tape may include oxidation of the Cu, converting the tape to an insulator, thus limiting the actuation of the PPy components connected by copper tape. For optimisation of the PPy film in this study, the geometry of PPy bending actuators is limited to 50µm thickness and 4mm width as the charge distribution has been found to be no longer uniform at thicknesses greater than 50-60µm and curling is prominent at widths greater than 4mm, producing an increase in elastic stiffness of the actuator [2, 8]. These limitations are also expected to be evident in PPy linear actuators.

A significant amount of work to determine the performance of bending PPy actuators has been previously conducted [5, 8] However, the complexity and greater degree of variability in a linear actuator means that the performance of a bending actuator may not directly translate to linear actuators. The modelling of linear actuators is necessary to accurately predict the force, displacement and work outputs for control of the actuator in practical applications. Modelling and optimisation work on PPy bending actuators by Alici, Metz & Spinks [2], in which a mathematical model to calculate the expected bending behaviour of PPy bending actuators was developed, and suggests that as the length of the actuator decreases, force output increases. This study has investigated the effect of the length of PPy components in linear actuators to determine if length has a similar effect on outputs in linear actuators. Combinations of 2mm, 3mm, 4mm wide and 10mm, 15mm and 20mm long linear actuators, where the length refers to the length of each component, were experimentally evaluated for force and displacement outputs. It was found that an increase in the length of the PPy links in the linear actuators allowed a greater range of motion, whilst shorter and wider actuators lifted greater loads. The linear actuators were also compared to bending actuators of corresponding dimensions and it was found that the bending actuators had a greater range of motion and the linear actuators lifted heavier loads. As the proposed linear actuation mechanism converts the rotational work into the linear work, ideally they are equal to each other, it is expected that while the displacement is decreasing, the force is increasing, and vice versa.

2 Functioning Principle of Linear Actuation System

As the polymer actuators considered in this study are cantilevered from one end, they generate a rotary (bending) type motion. This is analogous to a single jointed crank motion. A single degree of freedom mechanism such as a four-bar mechanism does not allow control the output motion. With this in mind, a five-bar mechanism which needs two inputs (two bending type polymer actuators) is one of the multi crank mechanisms having practical importance, especially for following any arbitrary trajectory precisely [9]. This mechanism serves as a bending to-linear motion converter; converting bending angle into a linear movement, and bending moments into linear forces.

2.1 Kinematic Analysis

The topology of the mechanism is depicted in Figure 1, where it is assumed that the mechanism links are rigid for the sake of generating a pseudo-rigid-body-model for the size and operation optimization of this linear motion mechanism. The joints connected to the frame are the active ones. The others are passive. For the joint inputs θ_1 and θ_2 and the specified link lengths L_0, L_1, L_2, L_3, L_4 the analytical expressions for the coordinates of the output point **P** are obtained using the following algorithm [9];

$$\begin{aligned} \vec{L}_1 &= L_1 \cos\theta_1 \,\vec{i} + L_1 \sin\theta_1 \,\vec{j}, \\ \vec{L}_5 &= \left(L_0 + L_4 \cos\theta_2\right)\vec{i} + L_4 \sin\theta_2 \,\vec{j}, \end{aligned} \tag{1}$$

$$\left|AB\right| = \vec{R} = \left|O_1B\right| - \vec{L}_1 = C\,\vec{i} + D\,\vec{j}, \tag{2}$$

$$\begin{aligned} &C = \left|O_1B\right|_x - L_{1x}, \quad D = \left|O_1B\right|_y - L_{1y}, \\ &R = \sqrt{C^2 + D^2}, \end{aligned} \tag{3}$$

$$Q = \frac{L_2^2 + R^2 - L_3^2}{2R^2}, \tag{4}$$

The coordinates (x_P, y_P) of the output point P are;

$$\begin{aligned} x_P &= x_A + Q\,C - D\sqrt{\frac{L_2^2}{R^2} - Q^2}, \\ y_P &= y_A + Q\,D + C\sqrt{\frac{L_2^2}{R^2} - Q^2}, \end{aligned} \tag{5}$$

Depending on the link lengths, the expression in the square-root can become negative, which suggest that some kinematic design constraints are not satisfied. For a linear motion output, the output point can be constrained to move in a rectilinear fashion. This requires that $\theta_1 = \theta_2$ AND $L_1 = L_4$ AND $L_2 = L_3$. Such a mechanism

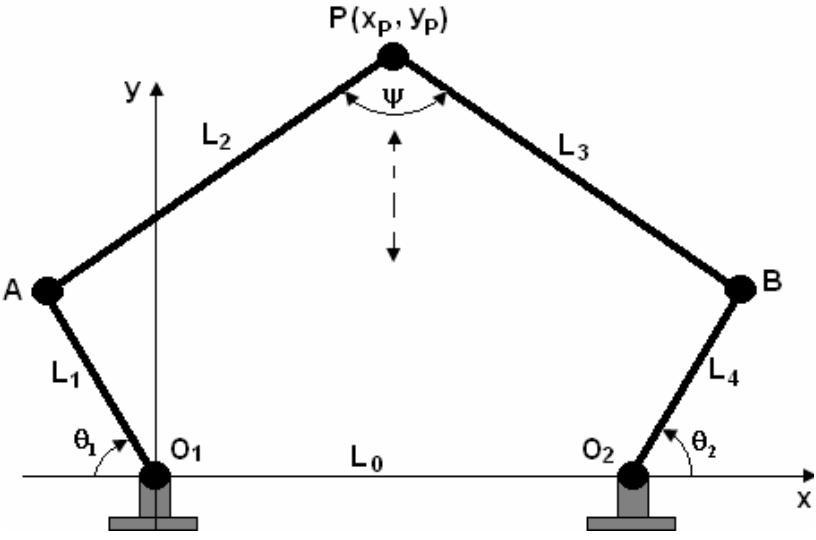


Fig. 1. Proposed linear actuation module based on a parallel five-bar mechanism

will convert all of its rotary work into the linear work - an efficient motion converter. The analytical expressions for the output point of such a linear motion module are obtained as

$$\begin{aligned} x_P &= -L_1 \cos\theta_1 + \frac{1}{2}\left[L_0 + L_1(\cos\theta_2 + \cos\theta_1)\right] \\ &\quad - L_1(\sin\theta_2 - \sin\theta_1)\sqrt{\frac{L_2^2}{R^2} - \frac{1}{4}} \\ y_P &= L_1 \sin\theta_1 + \frac{L_1}{2}(\sin\theta_2 - \sin\theta_1) \\ &\quad + \left[L_0 + L_1(\cos\theta_2 + \cos\theta_1)\right]\sqrt{\frac{L_2^2}{R^2} - \frac{1}{4}} \end{aligned} \tag{6}$$

where

$$\begin{aligned} R^2 &= L_0^2 + 2L_0L_1(\cos\theta_2 + \cos\theta_1) \\ &\quad + 2L_1^2[1 + \cos(\theta_1 + \theta_2)] \end{aligned} \tag{7}$$

Depending on the link length of the mechanism, Eq.6 simplifies to

i) For $L_1 = L_2$ and $\theta_1 = \theta_2 \Rightarrow x_P = \dfrac{L_0}{2}$ and

$$y_P = L_1 \sin\theta_1 + \frac{1}{2}\sqrt{4L_1^2 - L_0^2 - 4L_0L_1\cos\theta_1 - 4L_1^2\cos^2\theta_1}$$

ii) For $L_0 = 0$ and $\theta_1 = \theta_2 \Rightarrow x_P = 0$ and $y_P = L_1 \sin\theta_1 + \sqrt{L_2^2 - L_1^2 \cos^2\theta_1}$

iii) For $L_0 = 0$, $L_1 = L_2$ and $\theta_1 = \theta_2 \Rightarrow x_P = 0$ and $y_P = 2L_1 \sin\theta_1$

When choosing the link sizes, it is important for effective force transmission to minimize the variation of the transmission angle ψ from 90^0; the acceptable range is $90^0 \mp 40^0$ [10]. It is mathematically expressed as

$$\cos\psi = 1 - \frac{1}{2}\left(\frac{R}{L_2}\right)^2 \tag{8}$$

For $50^0 \le \psi \le 130^0$, $0.845 \le \left(\frac{R}{L_2}\right) \le 1.8126$. With reference to Eq.8, for a given L_0 and L_1, the range of L_2 satisfying the effective transmission angles can be calculated. It must be noted that L_0 should be greater than L_1 to prevent any physical interference among the mechanism links. A practical ratios of $1.5 \le \left(\frac{L_0}{L_1}\right) \le 3.0$ and $1.0 \le \left(\frac{L_2}{L_1}\right) \le 2.0$ should result in proportionate link lengths and transmission angles. For the data shown in Table 1, the transmission angle and the vertical movement of the linear actuation module are calculated and presented in Figure 2.

Table 1. Numerical values of the parameters for the exemplary results shown in Figure 2

Parameters
$L_0 = 40\,\text{mm}$
$L_1 = L_3 = 20\,\text{mm}$
$L_2 = L_4 = 22\,\text{mm}$
$\theta_1 = \theta_2 = 90^0$ to 115^0

The corresponding generalized relationship between rate of change $\dot{\Theta} = \begin{bmatrix}\dot{\theta}_1 & \dot{\theta}_2\end{bmatrix}^T$ of the input bending angles and the output velocity vector $\dot{X} = \begin{bmatrix}\dot{x}_P & \dot{y}_P\end{bmatrix}^T$ is given by [11].

$$\dot{X} = \begin{bmatrix}\frac{\partial x_P}{\partial(\theta_1, \theta_2)} & \frac{\partial y_P}{\partial(\theta_1, \theta_2)}\end{bmatrix}^T \dot{\Theta} = J\dot{\Theta} \tag{9}$$

where J is the mechanism Jacobian matrix.

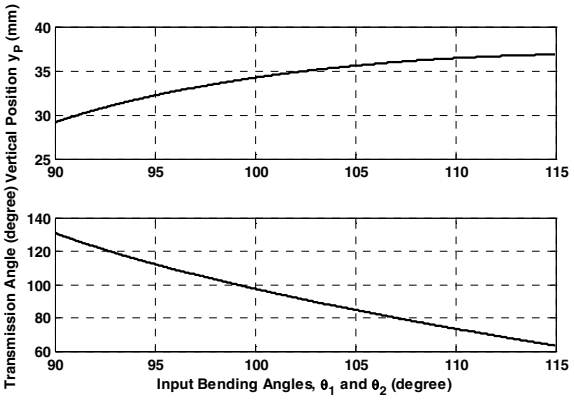


Fig. 2. Variation of the vertical position of the output point P and the transmission angle with the input bending angles. The net vertical distance is 7.6858 mm.

2.2 Force Analysis

Assuming that two active links O_1A and O_2B made of electroactive polymers generate bending moments M_1 and M_2, which act on the actuation module in the opposite directions. Assume that the output point P can apply a planar force vector of $F_p = F_x \vec{i} + F_y \vec{j}$ to the environment to realise a functional task. Please recall that the mechanism converts the work in the bending coordinates Θ into the work in linear coordinates X. These two works are ideally equal to each other.

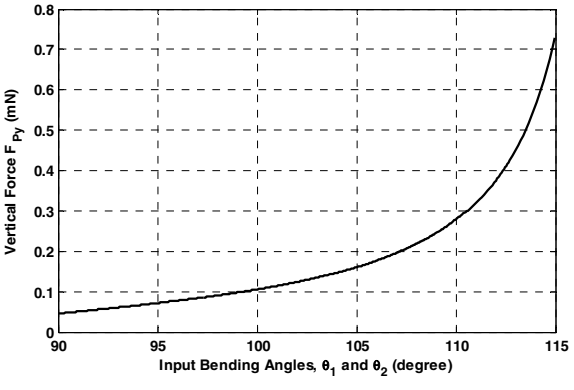


Fig. 3. Variation of the vertical force component of the output point P with the input bending angles. The net vertical force is 0.6837 mN. The horizontal component of the force is zero.

Using the duality between the generalized relationships for motion and force transfer between the actuation and output spaces, the following force relationship is obtained [11];

$$M = J^T F_P \quad \Rightarrow \quad F_P = J^{-T} M \tag{10}$$

where $F_P = \begin{bmatrix} F_x & F_y \end{bmatrix}^T$ and $M = \begin{bmatrix} M_1 & M_2 \end{bmatrix}^T$ and $(.)^T$ denotes transposition. For the data in Table 1 and $M = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$ Nmm, the force output at point P is calculated using Eq.10 and is presented in Figure 3. Because the mechanism link lengths are chosen such that it can generate a vertical movement, the mechanism will not create any horizontal force component. These kinematic and force analyses suggest that it is possible to optimize the topology of the mechanism for an efficient motion and force transmission.

3 Conducting Polymer Actuators: Synthesis and Actuation Principle

The PPy film was manufactured by combining a solution of 0.1M pyrrole monomer with 0.1M lithium triflouromethanesulfonimide (Li^+TFSI^-) in propylene carbonate (PC) with 1 % H_2O, degassed with nitrogen gas and stirred for 15 minutes. This solution was used to grow the PPy onto a gold sputter coated, 0.45µm Millipore Immobilon-P porous poly(vinylidene fluoride) (PVDF) film by electrodeposition over a period of 12 hours at -33°C with a current density of 0.1mA.m^{-2}. After polymerization, the film was washed in acetone and then resoaked in a 0.1M LiTFSI electrolyte and then stored away from oxygen. Actuators were then cut to desired size using a scalpel and were limited to size by the area of the 40mm x 55mm film. The resulting film consists of two PPy outer layers and two gold layers separated by a PVDF core is shown in Figure 4.

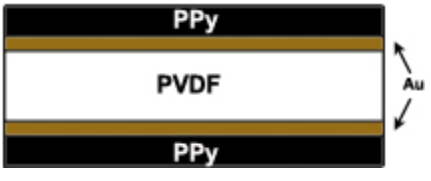


Fig. 4. General structure of trilayer polymer actuator

After doping with electrolyte, both the PPy layers of a bending actuator become partially oxidised (Figure 5a) and upon application of a positive voltage, one of the polymer layers is highly oxidized and the other is reduced. In an attempt to neutralise the charge imbalance, ions transfer from the electrolyte into the polymer layers. In LiTFSI doped actuators, $TFSI^-$ anions move into interstitial spaces in the polymer backbone of the oxidized layer, causing it to expand. Simultaneously, the opposing polymer film is reduced and contracts by removing $TFSI^-$ ions, overall creating a bending motion (Figure 5b.) By applying a square wave voltage, the actuator is allowed to return it to its original position and continue to bend through the neutral position so that the actuator may bend alternately in both directions.

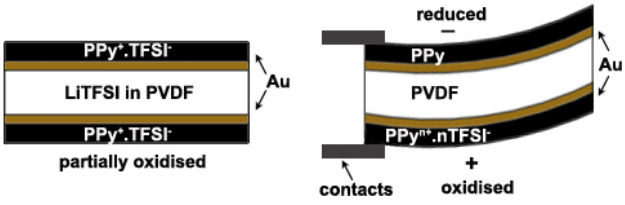


Fig. 5. Schematic of PPy actuator doped in LiTFSI (a) partially oxidised (b) under an applied voltage

To produce the linear actuators, two pieces of PPy film cut to equal dimensions were connected together in parallel at their tips with copper tape and two Ø 3mm × 0.5mm neodymium iron boron (NdFeB) magnets one either side of each tip (Figures 6-7.) Once placed between the electrodes, the PPy configuration effectively produces four PPy components, with the length referred to throughout this paper as the length depicted in Figure 7b.

Under an applied voltage, it was observed that the PPy linear actuator produced the shape as shown in Figure 8. In this case, the outer PPy layer of the upper components is oxidised and the inside layer is reduced, and under no constraints, almost uniform curvature may be experienced. The lower components may be activated; however the possible oxidation of the Cu tape suggests that the lower components remain either inactivated. This linear design restricts the bending motion of the actuators due to the presence of the magnets used to clamp the copper tape to the actuators (Figure 8). For the shorter lengths, a greater percentage of the actuator is affected by the clamping induced by the magnets and as a result, the actuator has a more elongated appearance (Figure 9). Due to this elongation, it doesn't move as far in the vertical direction as it potentially could. To overcome this limitation, another method of electrically connecting the upper and lower actuator components that does not restrict the motion is necessary. For this to be possible, the electrical connection must be flexible and able to bond to the actuator without the assistance of magnets. Although the magnets restrict motion, they do provide some stability which is important in attempting to control the actuator.

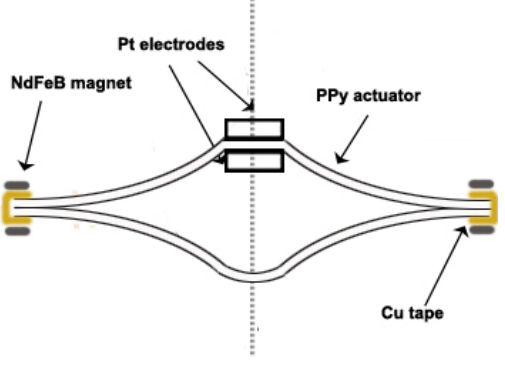


Fig. 6. Configuration of the linear PPy actuator with Cu tape and NdFeB magnets

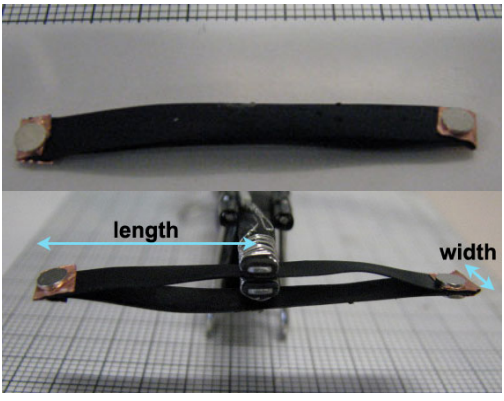


Fig. 7. Linear PPy actuator in neutral position

4 Performance Characterization Results

Linear actuators shown in Figures 6-7 with link lengths of 10mm, 15mm, 20mm and widths of 2mm, 3mm, 4mm were manufactured and tested under a series of applied loads. Loads were added to the bottom tip of the actuators and the overall displacement of the base tip was measured visually using a grid paper, as illustrated in Figure 10a. The initial load used was the non-magnetic wire hook and a 0.266g NdFeB magnet glued to its base (Figure 10b), with incremental loads applied by adding magnets of either 0.266g or 0.54g to the base of the wire hook. Voltage was maintained at +/-1V to avoid over-oxidation and reduce the variation in results that may arise due to early onset of fatigue. The frequency of the square voltage inputs was held constant for all experiments at 0.5Hz to allow sufficient response time. Typical voltage, current and charge data recorded by the datalogging system used is presented in Figure 11.

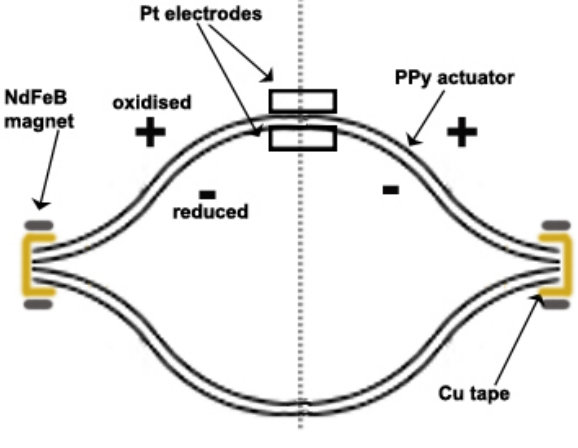


Fig. 8. Shape of actuator under actuation

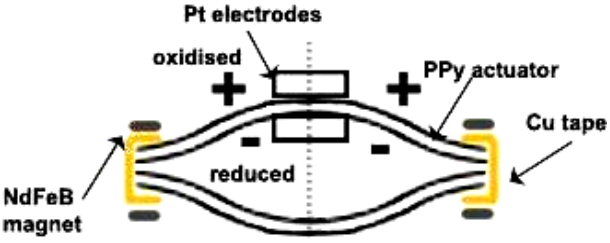
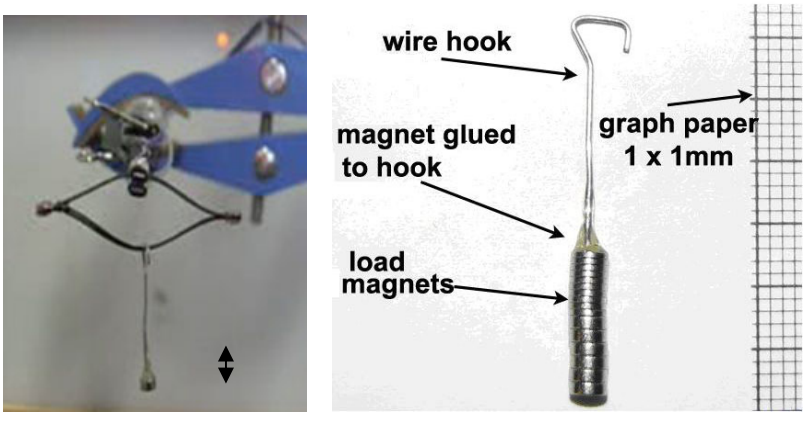


Fig. 9. Shape of shorter actuators under actuation



(a) (b)

Fig. 10. (a) PPy linear actuator with a load. (b) Wire hook; used to separate magnet loads from magnets on the actuator.

Figures 12(a)-(i) provide an overview of the base tip position in extended and contracted positions under given applied loads, with the electrode contacts as the reference point. In all actuators, as the load increases the position of the base tip below the reference point also increases. Generally, as the width decreases the position below the electrode increases and is more evident in the 15mm long actuators. As width increases, the actuator becomes stiffer which allows a smaller deflection below the reference point under a load than an actuator of a small width.

Displacement increases and then decreases for increasing load as shown in Figure 12(d)-(i) for the 15mm and 10mm long actuators. This behaviour is because under no load, the actuators are restricted in their movement by the magnets clamping the Cu tape in place. The clamping effect has a greater impact on shorter actuators because, as a percentage, a greater proportion of the actuator is restricted than longer actuators. Under applied loads, the shorter actuators are able to move because the load forces the actuator base tip to be displaced to lower than the original position, and under an applied voltage the actuator attempts to return to the original position and produces an upward motion. For this reason, the shorter actuators may only be suitable for applications requiring a pulling load, and if shorter actuators are required for pushing, then another method of connecting the actuators that provides less restriction on motion is required.

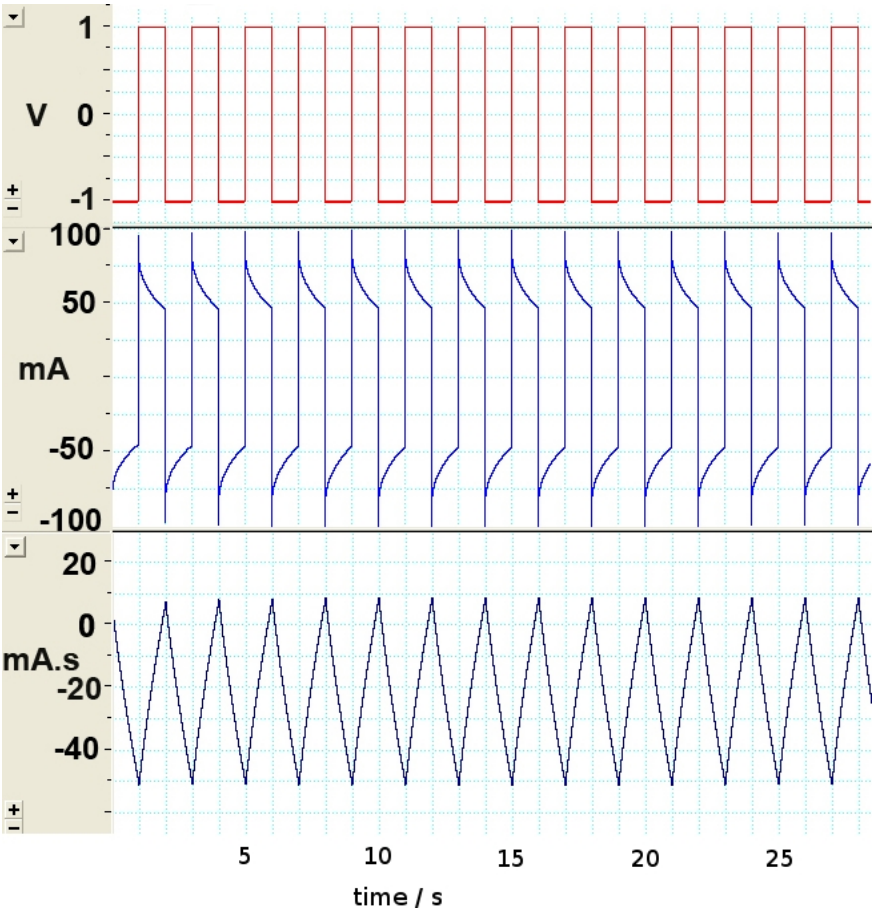


Fig. 11. Example voltage, current and charge input data for 20mm x 2mm linear PPy actuator at 0.5Hz

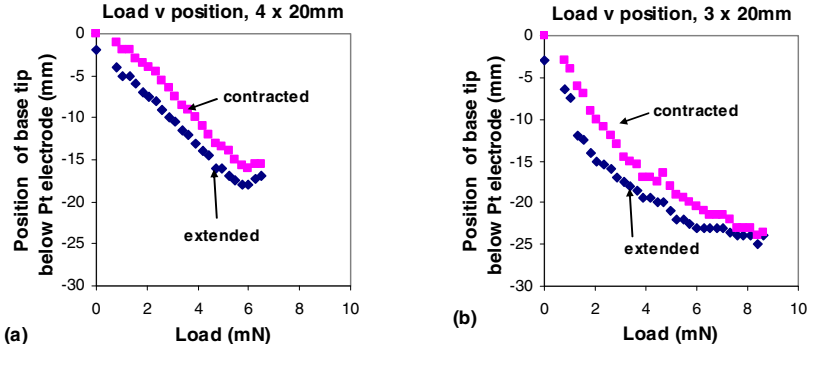
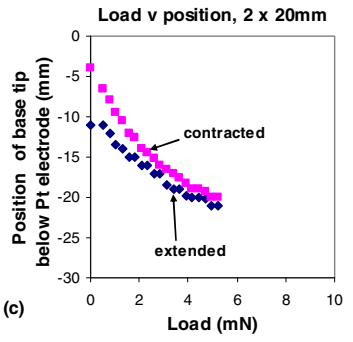
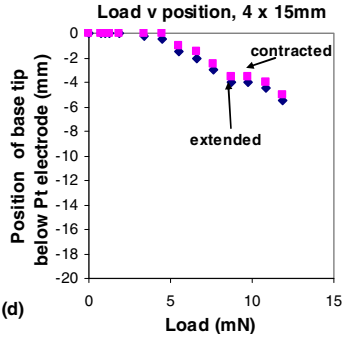
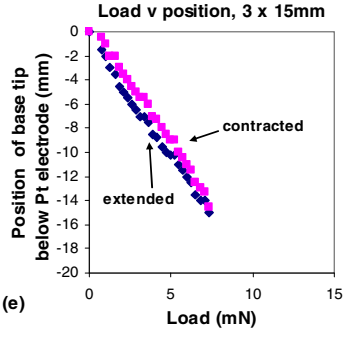
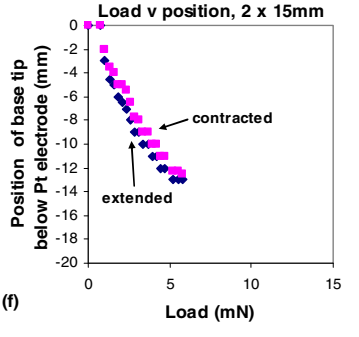


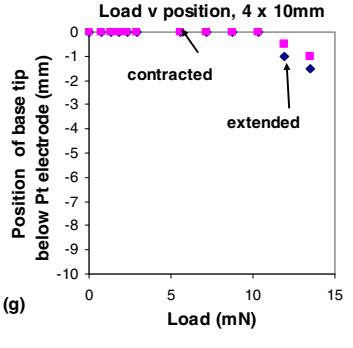
Fig. 12. Position of base tip of linear actuators under applied load

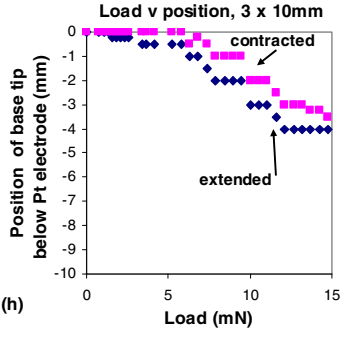












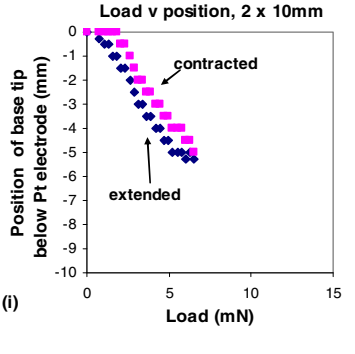


Fig. 12. (*continued*)

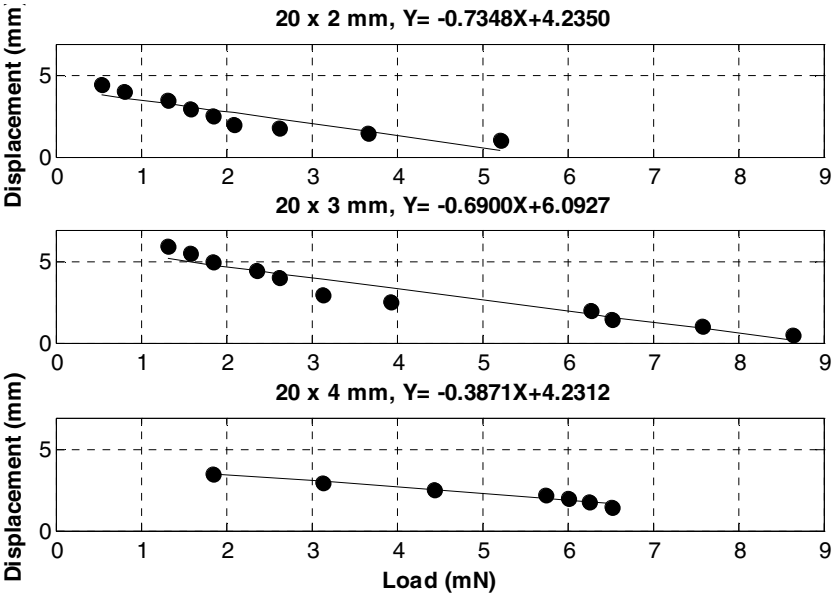


Fig. 13. Variation of the tip displacement of the linear actuation module with the payload

A comparison of load and displacement output between the linear and cantilever actuators of corresponding length has revealed that the cantilever actuators provide a greater range of motion in the vertical direction, while the linear actuator lifted heavier loads. The linear actuators are effectively 4 times the volume of their corresponding bending actuator. However, they produced much less displacement due to the restriction in motion provided by the magnet connections. The cantilever actuators have the ability to move above and below the electrodes whereas the linear actuators are limited to motion below the electrodes only. Although much less displacement is achieved, the benefit of a linear actuator is that motion in one direction is more controlled than the motion of a bending actuator. For displacement output, the linear actuators with 20mm PPy links produced the greatest displacement than actuators with shorter PPy links over a range of loads. Figure 13 depicts the displacement in the vertical direction for linear actuators of varied width with 20mm length. As width of the PPy links increases, a greater displacement output for a given load may be expected. However, as width increases, the PPy film has a tendency to curl across the width, particularly in actuators for 4mm wide or greater [4]. Curling increases stiffness and thus decreases the potential displacement output, hence the 4mm wide actuator produced a slightly smaller displacement than actuator with 3mm wide PPy links.

The calculated work output for actuators of varying length and width is presented in Fig. 14 and compared by the width for each actuator length. For the 20mm long

actuators, the 3mm and 4mm wide actuators produced a similar work output, and were significantly higher than the 2mm wide actuator. As width increases, the actuators should experience more charge and therefore should produce greater work. However, the 4mm wide actuator may have experienced curling, in which it would have contributed some of its energy to curling across the width of the actuator. The curling effect may also impede motion by increasing the stiffness of the actuator, therefore reducing work output by restricting motion.

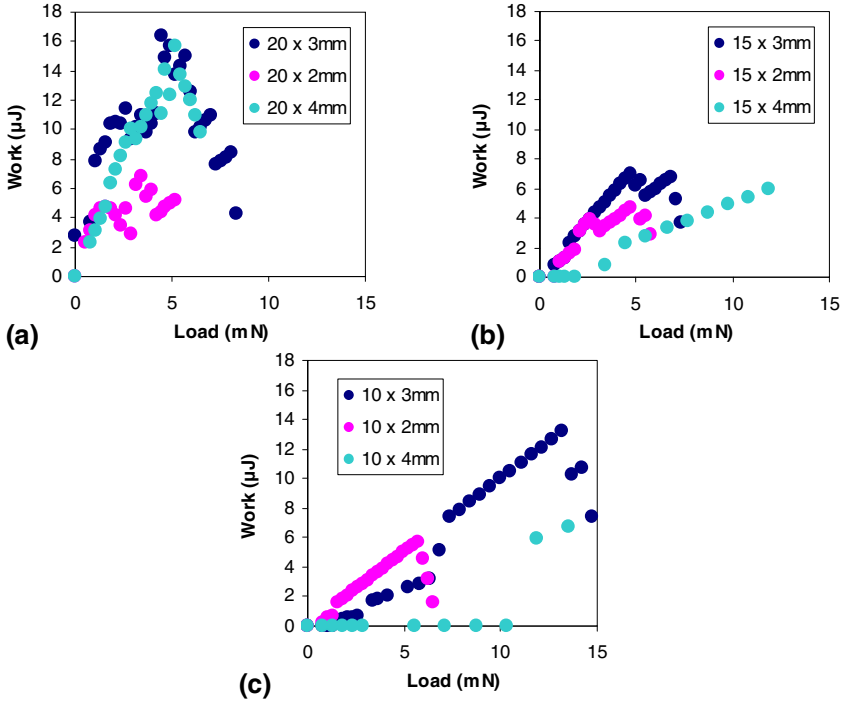


Fig. 14. Work versus load for PPy linear actuator: (a) 20mm long components (b) 15mm long components (c) 10mm long components

Work versus load plots should depict a parabola shape, which is evident in the 20mm long actuators (Fig.14a) The parabola shape is not present in the work versus load plots for the 15mm and 10mm long actuators (Fig. 14b-c) because a significant proportion of work being done was being opposed by clamping of the magnets at the hinges. Therefore, the magnets impeded the work output for shorter actuators, and the development of an alternate method of joining the components may allow greater output for shorter actuators. Maximum work for the 4mm x 15mm and 4 mm x 10mm actuators is not depicted in the Fig. 14b-c, however at the maximum work measured, these actuators produced very little displacement (Fig 13.) These results

suggest that the increased stiffness of the actuator created by the curling effect of the actuator could be useful in applications where higher loads and very small displacements are required.

Work density can be used to compare actuators of different geometries, performance and efficiency and is also a good measure to determine the actuators that are not performing near their potential. Fig. 15 compares the work density versus load of the 20mm long actuators with varying width. For these configurations, the 3mm wide actuator shows a significantly higher work density than both the 2mm and 4mm wide actuators. The 2mm and 4mm actuators exhibited similar work densities, and further testing would be required to determine which is more efficient. The 2mm actuator has a similar density to the 3mm wide actuator at lower loads (<1mN), but has significantly lower work density at higher loads. The 4mm actuator is less efficient than the 3mm actuator, due to a proportion of the input energy being contributed to curling across its width. Because of its higher work density, the 3mm wide actuator would be more preferable when constraints such as cost and volume exist. However, because the 2mm and 4mm wide actuators have a lower work density than the 3mm wide actuator, there is greater potential for their work output to be improved. If the efficiency of the 4mm wide actuator was improved then it would have a higher work output than the 3mm, which is shown in Figure 14.

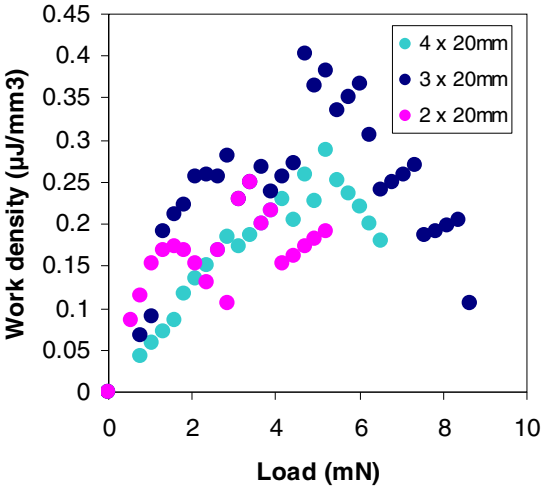


Fig. 15. Work density of PPy linear actuator with 20mm long components

5 Conclusions

A linear actuator system based on the bending motion of conducting polymer actuators operating in air is presented, including an analytical model to estimate the linear movement and the force output of the mechanism. The mechanism is basically a motion and force transmission system, converting the bending work provided by the electroactive polymer actuators into Cartesian work. The experimental results

presented demonstrate that the conducting polymer actuators generate enough displacement and force to handle a range of practical payloads. Another outcome of this study is that when the bending type- low power consuming polymer actuators are tailored properly, they can be used to generate a rectilinear motion with enough force output.

Future work involves deriving a more accurate analytical model taking into account the deflections of the mechanism links and verifying the model experimentally. Improvements may also be made to the hinge connections of the linear actuator, by replacing the copper connections with an inert, conductive material such as gold or platinum.

Acknowledgements. The authors thank Dr Stephen W. John for his help in synthesizing the bulk actuator sheet and construction of the actuation module. This work was supported in part by the ARC Centre of Excellence for Electromaterials Science (CE0561616).

References

1. Thompson, F.: Hybrid linear actuator based on conducting polymers. Faculty of Engineering. University of Wollongong (2007)
2. Alici, G., Metz, P., Spinks, G.: A mathematical model to describe bending mechanics of Polypyrrole (PPy) Actuators. In: Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Monterey, California (2005)
3. EAMEX Corporation, `http://www.eamex.co.jp/index_e.html`
4. Alici, G., Spinks, G., Huynh, N., Sarmadi, L., Minato, R.: Establishment of a Biomimetic Device Based on Tri-layer Polymer Actuators – Propulsion Fins. Bioinspr. Biomim. 2, 18–30 (2007)
5. Alici, G., Huynh, N.: Performance Quantification of Conducting Polymer Actuators for Real Applications: A Microgripping System. ISEE-ASME T Mech. 12, 73–84 (2007)
6. Otero, T., Cortes, M., Vazquez Arenas, G.: Linear movements from two bending triple-layers. Electrochim Acta 53, 1252–1258 (2007)
7. Yamakita, M., Kamamichi, N., Yashuaki, K., Aska, K., Luo, A.: Development of an artificial muscle linear actuator using ionic polymer metal composites. Adv. Robotics 13, 383–399 (2004)
8. Metz, P., Alici, G., Spinks, G.: A finite element model for bending behaviour of conducting polymer electromechanical actuators. Sensor Actuat A-Phys., 130–131, 1–11 (2006)
9. Alici, G.: Determination of Singularity Contours for Five-bar Planar Parallel Manipulators. Robotica 18, 569–575 (2000)
10. Alici, G., Shirinzadeh, B.: Optimum synthesis of parallel manipulators based on kinematic isotropy and force balancing. Robotica 22, 97–108 (2004)
11. Alici, G., Shirinzadeh, B.: Enhanced Stiffness Modelling, Identification and Characterisation for Robot Manipulators. IEEE T. Robot. 21, 554–564 (2005)
12. John, S., Alici, G., Cook, C.: Validation of a Resonant Frequency Model for Polypyrrole Trilayer Actuators. ISEE-ASME T. Mech. 13, 401–409 (2008)

Grasping Spatial Integrity in Distributed Unmanned Systems

Peter Simon Sapaty

Institute of Mathematical Machines and Systems, National Academy of Sciences
Glushkova Ave 42, 03187 Kiev, Ukraine
sapaty@immsp.kiev.ua

Abstract. Due to the increased complexity of tasks delegated to unmanned systems their collective use is becoming of paramount importance for performing any reasonable jobs. An approach is offered where integral group behaviors are achieved automatically rather than set up manually. Missions in a high-level Distributed Scenario Language can be executed jointly by communicating interpreters embedded in key system units. Robotic scenarios like reconnaissance, camp security, convoy, mule and explosive ordnance disposal oriented on different numbers of cooperating units are demonstrated. The approach allows us to effectively manage any teams, from manned to purely unmanned, regardless of the number of components in them. Other researched applications of the technology are outlined as well as its relation to gestalt philosophy, where the whole dominates over parts, defining their sense and even existence. The paradigm discussed may also be considered as a universal globally programmable distributed super-machine operating with both information and physical matter, ruling the world covered by it at runtime.

Keywords: Unmanned Systems, Distributed Scenario Language, Networked Interpretation, System Integrity, Robotic Swarms, Reconnaissance, Camp Security, Convoys, Explosive Ordnance Disposal, Gestalt, World Super-machine.

1 Introduction

With the world dynamics increasing due to global warming, numerous natural and manmade disasters, military conflicts, and international terrorism, using unmanned (ground, sea, underwater, and air) systems can alleviate many problems and save lives in hazardous environments. Because of the complexity of tasks delegated to unmanned solutions and still insufficient capabilities of existing robotic vehicles, their simultaneous, collective use may be of paramount importance to perform any reasonable jobs. Operating together, the unmanned groups, often called *swarms*, can fulfill the required objectives despite possible runtime damages to individual units.

We are offering a novel approach to organization of unmanned systems, oriented from the very beginning on parallel solutions in physical spaces, with swarm behaviors resulting naturally and accomplished automatically, rather than programmed manually. This approach, symbolically called “overoperability” from the

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 79–97.
springerlink.com

previous publications [1,2], allows us to create, modify, analyze, process, and manage any distributed systems, establishing local and global dominance over them.

Within the overoperability philosophy, an integral mission scenario, written in a special high-level scenario language [2,3] and reflecting semantics of what to do in a distributed space rather than details of implementation, is executed in a parallel and cooperative manner by dynamically networked unmanned units. During the scenario evolution, any operations can be accomplished in the world, along with the needed movement of code, equipment and "doers" (both artificial and biological) and creation and maintenance of physical and virtual infrastructures supporting the missions.

This paper has been essentially inspired by the European Land Robotic Trial [4] in which the author participated. It was conducted to provide trials as close as possible to operational scenarios for UGVs/UAVs with focus on short-term realizable robot systems. The day and night trials were organized within the following five main scenarios: non-urban reconnaissance, camp security, transport convoy, transport mule, and explosive ordnance disposal. Only a limited number of robotic units were engaged in every scenario, just one or two, whereas every scenario could potentially be executed with much higher efficiency if using robotic teams with many units, which cooperate with each other.

The paper also reflects activity on the project started under the sponsorship of Alexander von Humboldt Foundation (AvH) in Germany. One of its aims is formalization of known mission scenarios in such a way that they could be performed by any available numbers of robotic vehicles, with the management burden effectively shifted to self-organized robotic teams -- thus relieving human operators from traditional routines and allowing them concentrate on mission goals and global efficiency instead.

2 Distributed Scenario Language (DSL)

The approach described here is based on the Distributed Scenario Language (DSL) which allows us to set what to do in a distributed world on a semantic level, abstracting from details of how to do this and with which resources, delegating these to the intelligent automatic interpretation. Being a universal programming language with advanced parallel and distributed capabilities, DSL can also describe tasks and behaviors on any levels and their mixtures, if needed. The language can be used by humans who should follow its instructions individually or collectively, or can be directly executed by robots and their teams. Any mixed human-robotic organizations can be managed in DSL too.

2.1 DSL Worlds

The worlds DSL operates with can be virtual, physical, or combined.

- *Virtual World (VW)* is discrete and consists of nodes and links connecting these nodes. Any information can be associated with both nodes and links in the form of their names (contents). Nodes have unique addresses in VW, whereas their names (same as names of links) may repeat throughout the VW. Nodes can be accessed

directly, globally, by their names or addresses, or locally, from each other, via the (named) links, whereas links can be accessed only locally--from the adjacent nodes. A variety of broadcasting possibilities are available in the VW, both in a global and local way, for example, from outside to all nodes, from a node directly to all other nodes, or from a node to all neighboring nodes via the selected or all adjacent links.

- *Physical World (PW)* is continuous. Any point in it can be identified and accessed by the coordinates expressed in a certain coordinate system, also with certain precision. Staying in a PW point, you can lift local physical parameters from the world and, possibly, also change them, impacting the world locally too.
- *Virtual-Physical World (VPW)* is the one where VW nodes additionally associate with certain coordinates in the PW. VPW is discrete on a snapshot, but the nodes can change their physical coordinates overtime. The VPW nodes can be globally accessed by their names, addresses, or physical coordinates (for the latter, more correctly: by coordinates of the expected center and a radius of the region, due to limited precision of the coordinates). Also locally, via links--same as for the pure VW. In addition to the broadcasting capabilities of VW, nodes in VPW can also be massively accessed/entered by identifying a probable region in PW where they are expected to exist--by the region's center and a range (radius) from this center, where the latter may be of any value.

2.2 High-Level Scenarios

The world, as described above, is navigated and processed in a parallel and distributed way by high-level DSL scenarios having the following main features.

General Features. These define runtime coverage, or grasping, of distributed spaces.

- A DSL scenario (or program, in a conventional notation) describes development of activities in a distributed world as parallel transitions between sets of progress points, or *props.*
- Starting from a prop, a program action may result in one or more new props, or remain in the same prop.
- Each prop has a resultant *value* and a resultant *state*.
- Different actions (whatever complex they might be), starting from the same prop, may evolve independently or interdependently, and sequentially or in parallel, each contributing to the resultant set of props on this group of actions.
- Actions may also succeed each other in the space of props, with new actions applied in parallel from all props reached by the previous actions, resulting altogether in the integrated set of props on all these applications.
- Elementary operations can be defined on the values of props reached by other actions (the latter of any complexity), leading to the resultant prop with associated value (which may be multiple) and resultant state.
- The scenarios can form new or remove existing nodes and links in the distributed VW or VPW, allowing us to create, modify, and process any graph-based infrastructures in these worlds.

Association with World Nodes/Points. Scenarios evolve directly in distributed worlds while matching them, where:

- Any prop can be associated with a *node* in VW or a *position* in PW, or both, like in the case of VPW.
- A prop can also be linked separately with VW nodes and PW positions, allowing us to operate with the two worlds independently.
- Any number of props can be associated simultaneously with same points of the worlds.
- Staying with nodes/positions in the worlds, a prop allows us to directly access local data in these points, both virtual (information) and physical (matter).

Different Types of Variables. Used together, they allow us to create efficient spatial algorithms which work *in between* components of distributed systems, rather than *in* them. The variables are classified as:*Heritable variables* – these are starting in a prop and serving all subsequent props, which can share them in both read & write operations.

- *Frontal variables* – are an individual and exclusive prop's property (not shared with other props), being transferred between the consecutive props, and replicated if from a single prop a number of props emerge.
- *Environmental variables* – are accessing different elements of physical and virtual words when navigating them, also a variety of parameters of the internal world of DSL interpreter.
- *Nodal variables* – allow us to attach an individual temporary property to VW and VPW nodes; they can be accessed and shared by any props associated with these nodes.

Hierarchical Control. The scenarios can be locally and globally controlled, with flexible watershed between the two options.

- DSL scenarios can use a variety of spatial control rules, allowing us to assess local and remote states, make local and global decisions, and invoke or skip subsequent and terminate current operations, on the results of these decisions.
- Nested control infrastructures, embracing the whole scenario, provide interdependent local and global decisions associated with points of the worlds.

2.3 The Language Syntax

DSL has a recursive syntax shown below together with the names of its main constructs (where square brackets are for an optional construct, braces mean construct's repetition with a delimiter at the right, and vertical bar separates alternatives, whereas the whole program is historically called a *wave* [3]).

wave → constant | variable | [rule] ({ wave , })

constant → number | string | special

variable → identifier | reserved

rule	→	expand | transfer | modify | branch | advance | repeat | grant | echo | arithmetic | structural | assign | compare | timing | type | usage | identifier
special	→	abort | thru | done | fail | any | random | all |out | in | infinite | nil | empty | first | last random | virtual | physical | combined | neighbors | global | local | direct
reserved	→	N{alphameric} | H{alphameric} | F{alphameric} | TYPE | QUALITIES | NAME | ADDRESS | PLACE | WHERE | BACK | PREVIOUS | LINK | DIRECTION | ORDER | WHEN | TIME | SPEED | STATE | VALUE | COLOR | RESOURCES | DOER | USER | START
expand	→	hop | move | create | linkup
transfer	→	run | call | output
modify	→	split | partition | select | replicate | integer
branch	→	par | sequence | if | while | or | par or | and | par and
advance	→	advance | sync advance
repeat	→	cycle | loop | sling | repeat | repeat sync
grant	→	free | release | quit | none | lift | stay | grasp
echo	→	rake | min | max | sort | sum | average | product | count | state
arithmetic	→	add | subtract | multiply | divide | degree
structural	→	separate | unite | concatenate | append | intersect | content | index | rand
assign	→	assign | assign peers
verify	→	equal | not equal | less | less equal | more | more equal | empty | nonempty | belong | not belong | inside | not inside
timing	→	sleep | remain
type	→	nodal | heritable | frontal | environmental | info | matter | number | string
usage	→	address | name | place | center | range | time | speed | doer | node | link | unit

The basic construct, *rule*, can represent any action, decision or a guideline, for example:

- elementary arithmetic, string or logic operation;
- hop in a physical, virtual, or combined space;
- hierarchical fusion and return of (remote) data;
- distributed control, both sequential and parallel;
- a variety of special contexts for navigation in space, influencing different operations and decisions;
- type or sense of a value or its chosen usage, assisting automatic interpretation.

Different variants of this syntax and semantics had been implemented for previous DSL subsets [2,3], where conventional expression of operations and delimiters between program parts can be used too, for better readability and compactness. For example:

`3 + 5 + 7` – is the same as `add(3, 5, 7)`,
`w1; w2; w3` – is the same as `advance(`*`w1, w2, w3`*`)`,

where *w1* to *w3* may be arbitrary DSL programs (*waves*) themselves. The first example could have any programs instead of just numbers, each returning its (possibly, remote and multiple) results:

`w1 + w2 + w3` – same as `add(`*`w1, w2, w3`*`)`.

3 Distributed Interpreter

A variety of options may be available for automatic interpretation of DSL scenarios – from fully centralized and sequential to fully distributed and parallel. Due to peculiar syntax and semantics, the language interpretation in distributed systems is transparent and straightforward. Some basic features of the DSL interpretation are as follows.

- Direct association of props with world points drastically simplifies bringing data from the points to scenarios or vice versa: scenarios or their parts to world points.
- Chained actions can self-navigate and match the world, while omitting used "heads" and forwarding remaining "tails" further.
- Independent actions can be launched in parallel, developing autonomously in parts of the world.
- The interpreter copy can be installed in internet hosts, mobile robots, laptops, mobile phones, smart sensors, or implanted into biological units.
- The interpreter can also be a human being, performing manually of what is for herself while passing other parts of the scenario to other human or electronic interpreters and establishing dynamic command and control infrastructures between them.
- Any other systems can be accessed via the networked interpreters, the latter forming a supervisory layer managed in DSL.
- The interpreter copies may be concealed inside the systems to be impacted, even without their knowledge (to work in hostile environments).
- The interpreters can also migrate in the worlds to be managed, collectively executing (mobile too) mission scenarios, resulting altogether in a flexible and ubiquitous system organization.
- The DSL interpreter consists of a number of specialized modules working in parallel and handling and sharing specific data structures, which are supporting both persistent virtual worlds and temporary spatial control mechanisms [2,3].
- The heart of the distributed interpreter is its *spatial track system* enabling hierarchical command and control and remote data & code access, also providing high integrity of emerging parallel and distributed solutions, achievable without central facilities.

In application to robotic communities, the approach allows us to convert any group of mobile robots into a goal-directed cooperative system by integrating copies of the DSL interpreter, represented as a universal control module U in Fig. 1, with

traditional robotic functionalities (as, say, in [5]). (The figure exhibits mobile robots participated in the ELROB trial [4].)

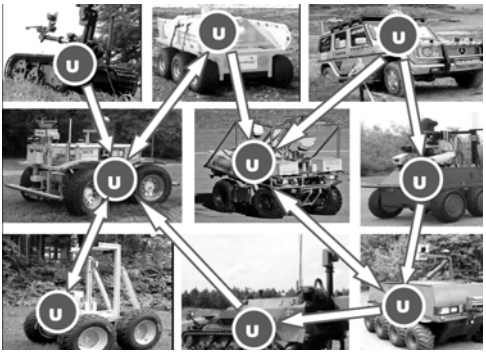


Fig. 1. Robotic teaming using embedded DSL interpreters

Any mission scenario in DSL can start from any robot, covering and tasking the whole system (or its parts needed) at runtime and in parallel. Subordination between the units, and dynamic command and control are established automatically, as a derivative of the mission scenario and current state of environment.

Due to fully interpretive nature of the technology, the scenarios can self-recover from any points, timely reacting on failures of robots. The whole group may remain fully functional and global-goal-oriented even in case of indiscriminate damages to individual units.

4 Elementary Example

An elementary task to be programmed in DSL may look like follows:

Go to given physical locations of the disaster zone (represented in a proper system of coordinates by the three locations): (50.433, 30.633), (50.417, 30.490), and (50.467, 30.517). Evaluate damage in each location, then find and return the maximum destruction value on all locations to a management center.

The corresponding program in DSL will be as simple as:

```
maximum(
  move((50.433, 30.633),(50.417, 30.490),(50.467, 30.517));
  assess(damage))
```

This program reflects semantics of the task to be performed in a distributed space, regardless of possible equipment that can be used for this. The latter may, for example, be a set of sensors scattered in advance throughout the disaster zone, where hopping by coordinates may result in a wireless access to the sensors already present there, not necessarily moving into these points physically.

As another solution, the program may order mobile robots to move into these locations and perform the needed damage assessment upon reaching the destinations. Some snapshots of the behavior of three available robots (R1, R2, and R3),

cooperatively performing this task, are shown in Fig. 2. After the initial injection of DSL program into R1 (Fig. 2*a*), a distributed interpretation infrastructure covering all three robots is created, and the scenario is partitioned and spread between the robots via this infrastructure (Fig. 2*b*).

All three robots then move independently to the locations automatically chosen (Fig. 2c). In each location reached independently by a corresponding robot, the damage assessment takes place, and all the results obtained are then being collected by R1 which finds the maximum value among the three ones on different locations. This final result will constitute the resultant value of the whole DSL program which will automatically be returned to the management center which remotely injected the DSL program into robot R1 (Fig. 2*d*).

As can be seen from the examples above, a semantic level scenario describing what to do in the distributed space can be interpreted by robotic teams autonomously, and by different numbers of cooperating robots (we could use two or a single robot instead, in which cases some robots will have to move to and serve more than one location in a sequence). The number of robots can also vary at runtime, during the scenario evolution, as some may get damaged while others involved at any stage.

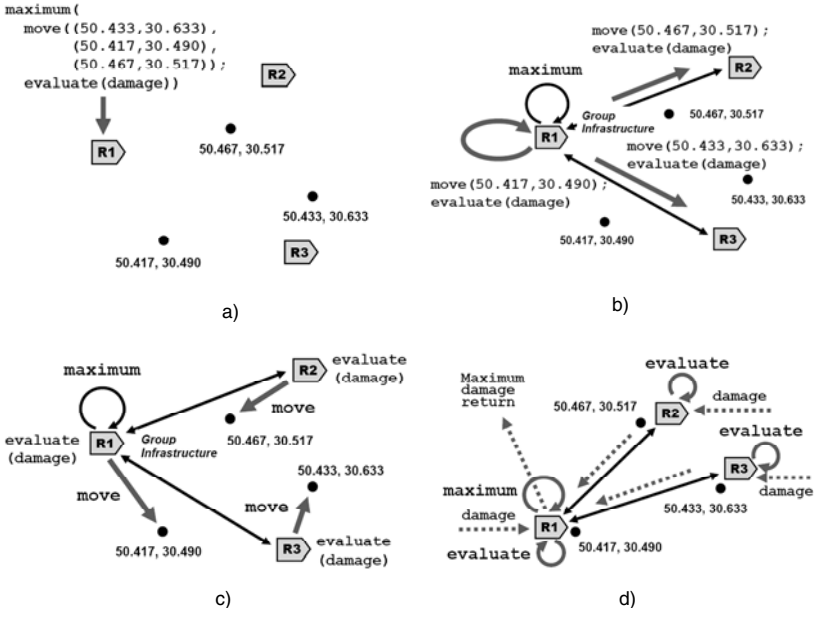


Fig. 2. Three-robot solution. a) initial scenario injection; b) infrastructure creation with tasking all robots; c) simultaneous robot movement; d) parallel damage assessment, finding and returning maximum.

5 More Robotic Scenarios

We will be using here the main scenarios that formed the basis of ELROB [4].

5.1 Non-urban Reconnaissance

For this scenario, it is supposed that a group of unknown vehicles is located in some distance in a non-urban area (defined, say, with the position of a center and area's radius), with security situation unclear there, so the reconnaissance should be done by robotic vehicles for not risking own personnel. The objective is to go to this target area and search for vehicles with specific characteristics. If found, they should be examined in detail, with their parameters collected and reported to the control station.

The general situation is shown in Fig. 3*a*, where the reconnaissance facilities should first go to the target area (towards its center), observe the area by cameras/sensors to roughly locate most probable targets (by their size, for example). The next will be to move directly to these selected targets and sense and collect their detailed parameters, with sending the results to the control point where they are accumulated and analyzed.

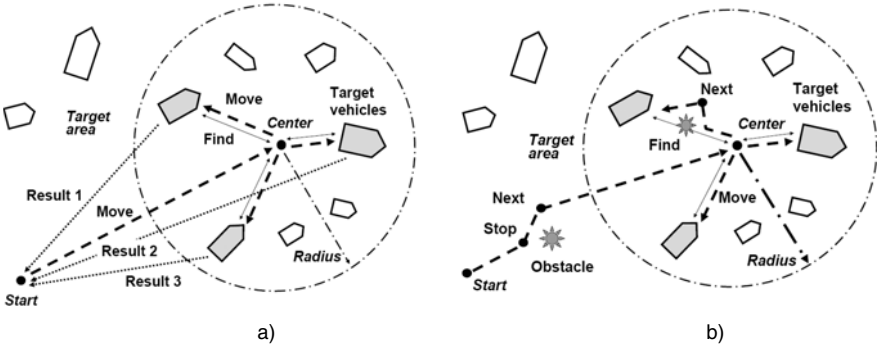


Fig. 3. The reconnaissance scenario. a) moving to and searching the area; b) avoiding obstacles.

Parallel Solution. This, in DSL, may allow us to use as many reconnaissance vehicles as possible (a single one including), potentially involving individual vehicles for each target identified, for their detailed examination.

```
USER = (move(start); WHERE = center;
Targets = recognize(radius, features);
split(Targets); WHERE = VALUE; collect(size, type, speed))
```

Explicitly Sequential Solution. The following DSL program just details navigation and organization procedures to execute the reconnaissance scenario in a strictly sequential way, which may be useful for optimization of the use of a single vehicle only.

```
move(start); WHERE = center;
  Targets = recognize(radius,features);
loop(WHERE = withdraw(Targets, 1);
     Result &= collect(size, type, speed)); USER = Result
```

Avoiding Obstacles. The movement to the target area and inside it may be complicated due to the presence of obstacles, as shown in Fig. 3*b*. The following DSL program, for the move from *Start* to *Center*, uses an external procedure

`approach_or_stop` to detect obstacles and stop to avoid collision, and the procedure `suitable` to find next suitable waypoint on the way to the destination, from which the move may continue.

```
move(start); loop(approach_or_stop(center);
  not_close(WHERE, center); WHERE = suitable(depth, center));
```

5.2 Camp Security

For the camp security scenarios, a defined urban area has to be monitored (think military camp) and this should be executed by robotic vehicles too, to minimize risk to human personnel. The objective is to detect and report irregularities in the area, like intruders, while acquiring their positions and imagery, and transmitting to control station.

The general picture is shown in Fig. 4, where the camp units (numbered 1 to 6) are simultaneously patrolled by a number of robotic vehicles moving along the paths between and around the buildings.

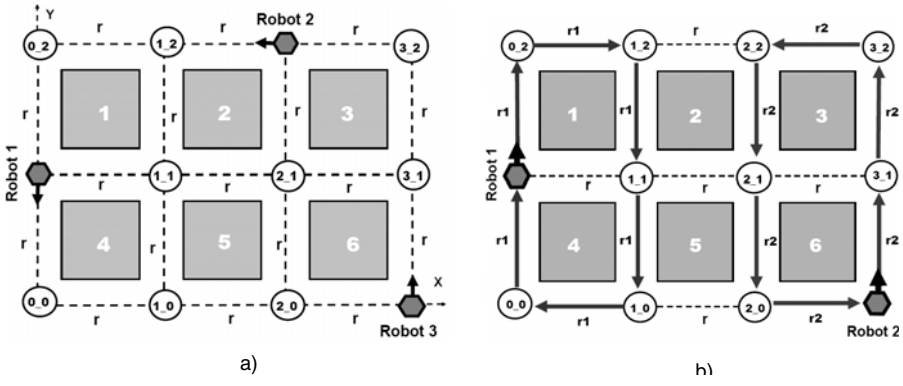


Fig. 4. Camp security solutions. a) random movement; b) using predetermined routes.

Distributed Campus Map. The proper routing of vehicles and resolution of possible conflicts between them (like collision avoidance) can be assisted by the creation and use of a distributed map of the campus area (just reflecting Fig. 4*a*) by the following DSL program (with node names reflecting X-Y coordinates of the crossings, and all links named `r`):

```
create(#3_1; F1 = A; r#2_1; F2 = A; r#1_1; F3 = A; r#0_1;
 (r#0_2; r#1_2; r#F3, (r#2_2; r#F2,(r#3_2; r#F1))),
 (r#0_0; r#1_0; r#F3, (r#2_0; r#F2,(r#3_0; r#F1))))
```

Random Movement. The next program organizes the duty performance by three parallel processes (which may be executed by three robots) using the created distributed map, with random choice of the next hop crossing and activation of the external service procedure `move_check_report` to analyze the local security situation while on the move.

```
hop(0_1, 2_2, 3_0); WHERE = CONTENT;
repeat(or((hop(link(random)); grasp(Mark == nil; Mark = 1);
  (hop(BACK); Mark) = nil; move_check_report(CONTENT)), stay))
```

Movement Via Predetermined Routes. If to use predetermined routes only, like the ones shown in Fig. 4b (one route using links named `r1` and another one `r2`), the collisions between robots can be avoided in full. These additional links `r1` and `r2` in the campus map can be installed by the following DSL program:

```
linkup(
 (#0_2; r1#1_2; r1#1_1; r1#1_0; r1#0_0; r1#0_1; r1#0_2),
 (#3_2; r2#2_2; r2#2_1; r2#2_0; r2#3_0; r2#3_1; r2#3_2))
```

And two independent spatial processes navigating the campus via the new links (which may engage two robots) can be organized by the following parallel DSL code:

```
 (hop(0_1); Flink = +r1),(hop(3_0); Flink = +r2);
 WHERE = CONTENT;
 repeat(hop(link(Flink)); move_check_report(CONTENT))
```

Any imaginable combinations of different types of simultaneous movement through the camp (like those by predetermined routes and/or by free, random, wandering) with collision avoidance can also be easily organized in DSL.

5.3 Transport Convoy

Imagine there is a delivery for a camp located in some distance. The objective is to move at least two vehicles to the target location, where only the first one can be manned and the second should follow the route of the first one, on a certain distance from it. We will consider a fully robotic solution for such a convoy, with two and also any number of vehicles, where only the first vehicle knows (and follows) waypoints toward the target location, while others dynamically chaining with, and following the previous ones on the move.

Two-unit Convoy. It is represented by the communicating `Leader` and `Follower`, where the first one defines its movement by a sequence of waypoints, and the second one, regularly requesting the Leader, moves to the positions previously occupied by it, while keeping a certain threshold distance. This is shown in Fig. 5, and by the DSL program that follows.

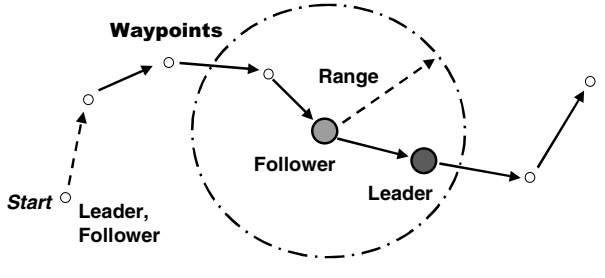


Fig. 5. Two-unit convoy

```
move(start);
(create(Leader); Waypoints = (w1, w2, w3, …);
 loop(WHERE = withdraw(Waypoints, 1))),
(create(Follower); sling(Lcoord = (hop(range, any); WHERE);
 distance(WHERE, Lcoord) > threshold; WHERE = Lcoord))
```

Multiple-unit Convoy. A scenario for the convoy with any number of chained processes (to be materialized by robotic units) is described by the following DSL program and depicted in Fig. 6. For this case, only the first process is a pure leader and the last process a pure follower, while all other processes combine both functionalities, i.e. being followers for the previous processes and leaders for the subsequent ones.

```
move(start); cycle(N < number; create(N += 1));
 (NAME == 1; Waypoints = (w1, w2, w3, …);
  Loop(WHERE = withdraw(Waypoints, 1))),
 (NAME != 1; sling(Lcoord = (hop(range, NAME-1); WHERE);
  distance(WHERE, Lcoord) > threshold; WHERE = Lcoord))
```

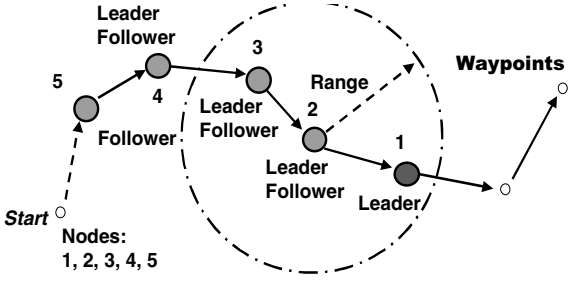


Fig. 6. Multiple-unit convoy

5.4 Transport Mule

Fir this scenario, there are two camps with a certain distance in between, and a cargo with a given weight should be transferred between the camps. We will consider here different possibilities to deliver the payload between the camps, using unmanned vehicles as "mules".

Delivery in a Single Piece. This may be the case if cargo's weight allows it to be put on a single vehicle, as shown ion Fig. 7.

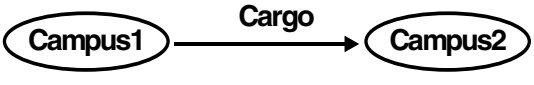


Fig. 7. Single piece cargo delivery

The related DSL program will be as follows:

```
move(Campus1); frontal(Cargo) = "substance";
move(Campus2); Store = Cargo
```

Shuttling between Camps. For this option, the process shuttles as often as possible between the two camps after partitioning the cargo into portions of the weight allowed, unless all the cargo is delivered, as shown in Fig. 8 and by the following program.

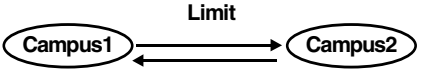


Fig. 8. Shuttling delivery

```
move(Campus1); frontal(Load); Cargo = "substance"; Limit = 50;
loop(or((weight(Cargo) > Limit; Load = withdraw(Cargo, Limit)),
        (weight(Cargo) > 0; Load = Cargo));
     hop(Campus2); Store += Load; hop(Campus1))
```

Multiple, Parallel Delivery. For this case, different processes (vehicles) are considered to be independent from each other, each moving to the destination as quickly as possible on its own (see Fig. 9 and the following program).

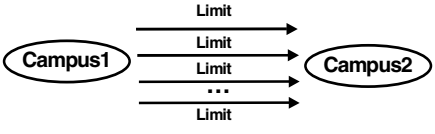


Fig. 9. Parallel cargo delivery

```
move(Campus1); frontal(Load); Cargo = "substance"; Limit = 50;
cycle(or((weight(Cargo)> Limit; Load = withdraw(Cargo, Limit)),
         (weight(Cargo)> 0; Load = Cargo)));
move(Campus2); Store += Load
```

Multiple, Convoy Delivery. For this scenario, the vehicles, each with a limited partition of cargo, are dynamically chaining in a column for a cohesive movement towards the destination (see Fig. 10 and the subsequent DSL program).

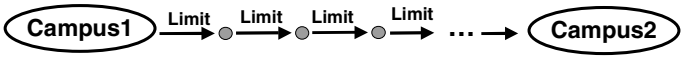


Fig. 10. Delivery in a convoy

```
move(Campus1); frontal(Load); Cargo = "substance"; Limit = 50;
cycle(or((weight(Cargo)> Limit; Load = withdraw(Cargo, Limit)),
         (weight(Cargo)> 0; Load = Cargo)); create(N += 1));
(NAME == 1; move(Campus2)),
(NAME != 1; loop(not_close(WHERE, Campus2);
   WHERE = (hop(NAME - 1); WHERE)));
Store += Load
```

5.5 Explosive Ordnance Disposal

Explosive Ordnance Disposal (EOD) means the detection, identification, onsite evaluation, rendering safe, recovery, and final disposal of Unexploded Ordnance (UXO) including detonation and burning. It is often said that the EOD operation is a 3 Ds one, which is Dangerous, Dirty and Demanding (or Difficult) job. Using robotic vehicles, especially multiple ones, is therefore becoming the most promising EOD option. Various kinds of EOD scenarios for navigation and examination of the target territory may be offered. We will just hint here on the simplest two options, easily expressible in DSL.

Sequential Territory Search. This represents a single-threaded process (oriented on a single vehicle), where the whole territory is incrementally scanned unless all being searched, as described by the following program and depicted in Fig. 11*a*.

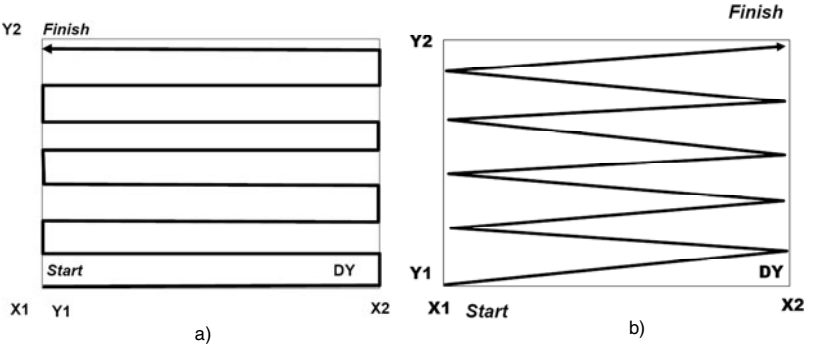


Fig. 11. Sequential territory search. a) conventional; b) zigzag-like.

```
X1 =…; X2=…; Y = Y1 =…; Y2 =…; DY =…;
loop(WHERE = (X1,Y); WHERE = (X2,Y); (Y += DY) < Y2;
     WHERE = (X2,Y); WHERE = (X1,Y); (Y += DY) < Y2)
```

The sequential coverage of the territory can be organized with minimum waypoints to pass, in a zigzag way, as shown in Fig. 11*b* and by the following program.

```
X1 =…; X2=…; Y = Y1 =…; Y2 =…; DY =…;
loop(WHERE = (X1, Y); (Y += DY) < Y2;
     WHERE = (X2, Y); (Y += DY) < Y2)
```

Parallel Territory Search. This can be represented by a number of independent processes, each starting from a different location, and navigating altogether the whole region in parallel, as depicted in Fig. 12*a*, and explained by the DSL program that follows (all processes follow the predetermined routes for this case).

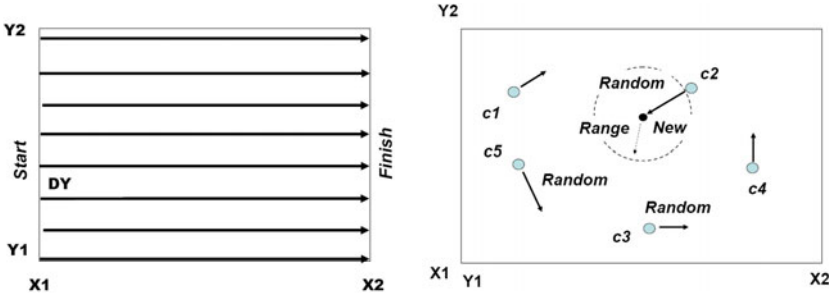


Fig. 12. Parallel territory search. a) with predetermined routes; b) random movement.

```
X1 =…; X2=…; Y1 =…; Y2 =…; DY =…; frontal(Y) = Y1; DDY = 0;
cycle((Y += DDY)<Y2; DDY += DY); WHERE =(X1,Y); WHERE =(X2,Y)
```

Parallel search of the territory can also be organized in a random way, where each process chooses its next hop randomly, also taking into account that the chosen next destination was not visited before (at least looking like this, say, with the help of visual sensors detecting prints after wheels). Parallel random search may have an advantage before predetermined search in that it can eventually cover all the territory despite possible failures of individual processes (robots). Such a search, with processes starting from some initial points (named *c1* to *c5*), also keeping certain threshold distance from each other, is shown in Fig. 12*b* and by the following program.

```
move(c1, c2, c3, c4, c5); Range = …;
X1 =…; X2=…; Y1 =…; Y2 =…; D =…; loop(
 New = WHERE + 2(random(-D, D)); inside(New, (X1, X2, Y1, Y2));
 hop(New, Range) & seen(New) == nil; shift_check_act(New))
```

6 Other Applications

Many other applications of the paradigm are possible, as follows, some of which already investigated, tested, and results published [1-3, 6-10].

Emergency Management. Using interpreters installed in massively wearable devices may allow us to assemble workable systems from any wreckage after the disasters, using any remaining communication channels, manual including. These emergent systems can provide distributed self-awareness, collect statistics of casualties, guide the delivery of relief goods, and coordinate collective escape from the disaster zone.

Directed Energy Systems. The technology can provide high flexibility in organizing directed energy (DE) systems, especially in crisis situations, making automatic distributed decisions with the “speed of light” too. It may also help automate the global power dominance by optimized delivery of directed energy into any world points via dynamically organized networks of relay mirrors.

Distributed Avionics. Implanting interpreter copies into main control points of the aircraft may provide a higher, intelligent, layer of its self-analysis and recovery, by the spreading recursive scenarios starting from any point and collecting & fusing key data from other points. The embedded interpretation network with local, dynamic, and emergent links will be fully functional under any damages, especially with wireless communications between the interpreters. This may always provide global control integrity, even in a physically disintegrating object, helping to save lives and complete missions.

Sensor Networks. Wireless sensors may be dropped from the air massively, as "smart dust". Having a limited communication range, they must operate in a network to do nonlocal jobs in a distributed environment. With the technology offered, we can convert their emergent networks into a universal parallel computer operating in DSL. It can effectively solve complex distributed problems--from just collecting and fusing scattered data to outlining and assembling images of the distributed phenomena like, for example, flooding, smog, flocks of birds, movement of troops, etc.

Advanced Command and Control. In DSL, it is possible to define high-level scenarios concentrating on mission goals and top decision-making while delegating C2 routines, appearing at runtime as a derivative of the mission and environment states, to automatic interpretation. It is also convenient to express in DSL any theoretical and practical issues of advanced C2 explicitly.

Infrastructure Protection. Navigating the systems at runtime, the technology can analyze safety and integrity of critical infrastructures and key resources, establishing protective networked mechanisms throughout them. Other systems can be involved from the DSL interpretation layer for emergent infrastructure protection and recovery, including air and space defense, police and army. In relation to energy infrastructures, the technology can help observe power networks from the air or ground, trace electric, gas or oil supply lines, sensing their states (and, if needed, directly accessing the disaster zones), also providing regular or emergent sentry duties at power installations, etc.

Global and Battlespace Dominance. The DSL scenarios, using any electronic media, can self-spread, outline, and grasp distributed systems of different natures while establishing global dominance over them. They can analyze their internal infrastructures, finding strong and weak points, orient behavior, or destroy the infrastructures or the system as a whole if required. The approach, as an intelligent self-recovering super-virus, which is difficult to discover and kill by traditional means, can effectively employ advanced robotic facilities, like swarms of aerial and ground vehicles, to attack adversarial systems.

7 Gestalt-Related

Our approach may be considered as one of the first attempts to formalize and implement the notion of *gestalt* [11,12], under which the whole dominates over parts (being greater than the sum of them) with parts having sense only in the context of the whole, rather than vice versa. Gestalt theory represented the main departure from

atomistic vision of systems at the beginning of the last century. Many existing systems, especially distributed ones, are still based on the concept of predetermined parts (agents) that communicate with each other in an attempt to get the global behavior needed. The latter, with rapidly growing number of agents in complex systems (and starting from the agents level) is becoming more and more problematic.

Within the approach offered, we have come to quite a different and higher level model of the system organization. Abstracting from system parts and their interactions, which may be emergent and varying at runtime, we can describe the needed global system behavior on a semantic level, where parts and their interactions may not be known in advance, and may dynamically appear (disappear too) just to maintain the global behavior needed.

The technology developed allows us to automatically interpret global system scenarios in any networked systems (comprising internet hosts, laptops, mobile robots, mobile phones, smart sensors, and/or humans themselves). It allows us to get even higher—to describe *what the system should do* on the highest level, whereas its local and global behavior becomes a *derivative of this description*, which, in its turn, makes the system structures and operation as a *further derivative*.

8 Conclusions

A novel ideology and technology, converting any distributed system into a universal spatial machine capable of solving complex problems on itself and on the surrounding environments, has been presented. This conversion can be achieved by implanting into the system sensitive points and its active doers, humans and robots including, of the same copy of a universal control module, communicating with other such modules via available channels. Their entire network, which may be dynamic and emergent, collectively interprets mission scenarios written in a special high-level language, which are defining system's internal and external behavior.

Created and modified on the fly, the scenarios can start from any component, covering the system at runtime through the cooperating interpreters. During the scenario evolution, any operations can be carried out throughout the distributed world, along with the needed movement of code, equipment and artificial or biological doers, humans including, as well as creation and maintenance of physical and virtual infrastructures supporting the missions.

The approach offered can dramatically simplify application programming in distributed systems, especially robotized ones. As can be seen from the examples throughout this paper, programming multi-robot scenarios in distributed and dynamic environments in DSL may not be more difficult than, say, programming of routine data processing tasks in traditional languages like Fortran, C, or Java.

The distributed robotized systems are of rapidly growing importance in many areas, and especially in defense, where robotic swarming on asymmetric battlefields is becoming a major dimension of the new military doctrine for the 21st century [13]. The written above is much in line with these trends, allowing us to flexibly combine loose swarming with more classical command and control, which can help gradually transform fully manned into mixed and ultimately totally unmanned systems.

Other prospective applications of this work can be linked with economy, ecology and weather prediction—by using the whole networked world as a spatial supercomputer, self-optimizing its performance.

The approach offered may also be compared with the invention of the first world computers and first high-level programming languages [14,15]. In our case, this computer may not only operate with data stored in a localized memory, but *can cover, grasp, and manage any distributed systems, the whole world including, and can work not only with information but with physical matter or physical objects too*.

Acknowledgements. This research has been partially funded by the Alexander von Humboldt (AvH) Foundation in Germany. The support from international ICINCO conferences and personally by Joaquim Filipe, Marina Carvalho, and Vitor Pedrosa was invaluable for the development and testing of the ideas presented. And years of cooperation with the department of simulation of complex systems at the Ukrainian Academy of Sciences and numerous hot discussions (from ideology to politics to world dynamics to emerging threats) with Anatoly Belyaev, Eugene Bondarenko, Igor Gorban, Vitaly Klimenko, Yuri Korovitsky, Vladimir Voloboev and others were particularly stimulating for this work.

References

1. Sapaty, P.S.: Over-Operability in Distributed Simulation and Control. The MSIAC's M&S Journal Online 4(2) (Winter Issue) (2002)
2. Sapaty, P.S.: Ruling Distributed Dynamic Worlds. John Wiley & Sons, New York (2005)
3. Sapaty, P.S.: Mobile Processing in Distributed and Open Environments. John Wiley & Sons, New York (1999)
4. M-ELROB-2008. Military European Land-Robot Trial. Hammelburg, Germany (2008)
5. Kuhnert, K.-D., Krödel, M.: Autonomous Vehicle Steering Based on Evaluative Feedback by Reinforcement Learning. In: Perner, P., Imiya, A. (eds.) MLDM 2005. LNCS (LNAI), vol. 3587, pp. 405–414. Springer, Heidelberg (2005)
6. Sapaty, P.: Intelligent management of distributed sensor networks. In: Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense VI. Proc. of SPIE, vol. 6538, p. 653812 (2007)
7. Sapaty, P.: Distributed Technology for Global Dominance. In: Suresh, R. (ed.) Proc. of SPIE. Defense Transformation and Net-Centric Systems, vol. 6981, 69810T (2008)
8. Sapaty, P.: Grasping the Whole by Spatial Intelligence: A Higher Level for Distributed Avionics. In: Proc. International Conference Military Avionics 2008, Cafe Royal, London (2008)
9. Sapaty, P., Sugisaka, M., Finkelstein, R., Delgado-Frias, J., Mirenkov, N.: Advanced IT Support of Crisis Relief Missions. Journal of Emergency Management 4(4) (2006)
10. Sapaty, P., Morozov, A., Sugisaka, M.: DEW in a Network Enabled Environment. In: Proc. International Conference Directed Energy Weapons 2007, Le Meridien Piccadilly, London, UK (2007)
11. Wertheimer, M.: Gestalt Theory. Erlangen, Berlin (1924)

12. Sapaty, P.: Gestalt-Based Ideology and Technology for Spatial Control of Distributed Dynamic Systems. In: Proc. International Gestalt Theory Congress, 16th Scientific Convention of the GTA, University of Osnabrück, Germany (2009)
13. Singer, P.W.: Wired for War: The Robotics Revolution and Conflict in the 21st Century, Penguin (2009)
14. Rojas, R.: Konrad Zuse's Legacy: The Architecture of the Z1 and Z3. IEEE Annals of the History of Computing 19(2) (1997)
15. Zuse, K.: Uber den Plankalk, als Mittel zur Formulierung schematisch kombinativer Aufgaben. Archiv Mathematik, Band I (1948/1949)

Convergent Iterative Feedback Tuning of State Feedback-Controlled Servo Systems

Mircea-Bogdan Rădac[1], Radu-Emil Precup[1], Emil M. Petriu[2], Stefan Preitl[1], and Claudia-Adina Dragoş[1]

[1] "Politehnica" University of Timisoara, Dept. of Automation and Applied Informatics
Bd. V. Parvan 2, 300223 Timisoara, Romania
[2] University of Ottawa, School of Information Technology and Engineering
800 King Edward, Ottawa, ON, K1N 6N5, Canada
{mircea.radac,radu.precup,
stefan.preitl,claudia.dragos}@aut.upt.ro,
petriu@site.uottawa.ca

Abstract. This paper presents a new Iterative Feedback Tuning (IFT)-based optimal state feedback control solution dedicated to a class of second-order servo systems with integral component. The state feedback controllers for these controlled plants are extended with an integrator to ensure the rejection of constant load type disturbances. Original IFT algorithms are suggested for the accepted state feedback controllers such that to set the step size in order to guarantee the convergence. An attractive convergence theorem based on the application of Popov's hypertability theory to the parameter update law in the IFT algorithms is offered. The theoretical results are validated by a case study concerning the position control of a DC servo system with backlash. Implementation issues are discussed and exemplified by real-time experimental results.

Keywords: Implementation Issues, Iterative Feedback Tuning, Second-order Servo Systems, Real-time Experimental Results.

1 Introduction

The second-order servo systems with integral component play an important role as controlled plants in a large category of industrial applications including mechatronics, electrical drives, sub-systems in power plant control systems, positioning systems in manipulators, mobile robots, machine tools, flight guidance and control [1]–[8]. Those controlled plants are viewed as special cases of benchmark systems [9]–[11]. A convenient way to develop control solutions dedicated to these controlled plants makes use of linearized models at certain operating points. Therefore the parameter variation associated with the linearization is challenging when very good control system performance indices are required. The design and implementation involve more challenges when low-cost automation solutions are needed.

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 99–111.
springerlink.com

The control solutions based on state feedback control systems are can cope with the accepted class of plants. The optimal state feedback control systems can fulfill the main control aims i.e. high performance indices in reference input tracking and regulation with respect to several types of load disturbance inputs. With this regard the improvement of the control system performance indices (e.g. settling time and overshoot) is enabled by the minimization of appropriately defined objective functions. The Iterative Feedback Tuning (IFT) [12], [13] is an alternative to the experiment-based minimization of the objective functions. The IFT algorithms employ the input-output data measured from the closed-loop system during its operation to calculate the estimates of the gradients and eventually Hessians of the objective functions. Several experiments are conducted per iteration and the updated controller parameters are calculated on the basis of the input-output data and the estimates.

The extension of IFT according to [14] offers additional steps to improve the convergence properties of IFT while rejecting the disturbances. Several extensions of IFT to Multi Input-Multi Output (MIMO) systems are discussed in [15]–[18]. Linear applications to digitally simulated benchmarks are illustrated in [15], [16]. The need for faster gradient approximations and local convergence in IFT for multivariable processes are thoroughly discussed in [17]. Recently reported industrial applications of IFT include the control of chemical plants [18] and servo drives [19].

Many experiments are needed for more state feedback control systems and MIMO systems. The need to reduce the number of experiments per iteration has been highlighted in [14]–[17].

The IFT-based state feedback control meant for second-order servo systems with integral component has been suggested in [20]. Building upon [20] this paper presents twofold new contributions. First, original convergent IFT algorithms are suggested. They are based on the formulation of the parameter update law in the IFT algorithms as a nonlinear dynamical feedback MIMO system in the parameter space and iteration domain. Popov's hyperstability analysis results [21], [22] are applied in this context to derive a new stability theorem which guarantees the convergence of the IFT algorithms and gives an useful condition to set the step size. Second, a thorough discussion of an extensive set of real-time experimental results is done.

This paper is organized as follows. The state feedback control system structure dedicated to the second-order servo systems with integral component is presented in Section 2. Section 3 focuses on the new IFT algorithm. Section 4 discusses implementation issues. A case study concentrated on the state feedback position control of a DC servo system with backlash is described in Section 5. The real-time experimental results validate the new control solution and IFT algorithm. Section 6 highlights the conclusions.

2 Control System Structure

The second-order servo systems as controlled plants are characterized by the state-space model [20].

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & -\dfrac{1}{T_s} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \dfrac{K_s}{T_s} \end{bmatrix} u, \quad \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \mathbf{I}_2 \, \mathbf{x}, \tag{1}$$

where $\mathbf{x} = [x_1 = \alpha \quad x_2 = \omega]^T$ is the state vector, α is the (angular) position, ω is the (angular) speed, u is the control signal, y_1 and y_2 are the controlled outputs, $\mathbf{I}_2$ is the second-order identity matrix, and the superscript T indicates the matrix transposition. The two parameters in (1) are the process gain $K_s, K_s > 0$, and the small time constant or the sum of parasitic time constants $T_s, T_s > 0$.

The two transfer functions considering the input u and output ω, and the input u and output α are $P_{\omega,u}(s)$ and $P_{\alpha,u}(s)$, respectively, with the following expressions:

$$P_{\omega,u}(s) = \frac{K_s}{(1+sT_s)}, P_{\alpha,u}(s) = \frac{K_s}{s(1+sT_s)} \cdot \tag{2}$$

The integral component can be observed in $P_{\alpha,u}(s)$ when the controlled output is $y_1 = x_1 = \alpha$. Such situations correspond to positioning systems.

The state feedback control system structure is presented in Fig. 1. The dotted connection is used only when the experiments specific to IFT are conducted. That connection is not applied during the normal control system operation, therefore the control system structure is not a model reference adaptive one.

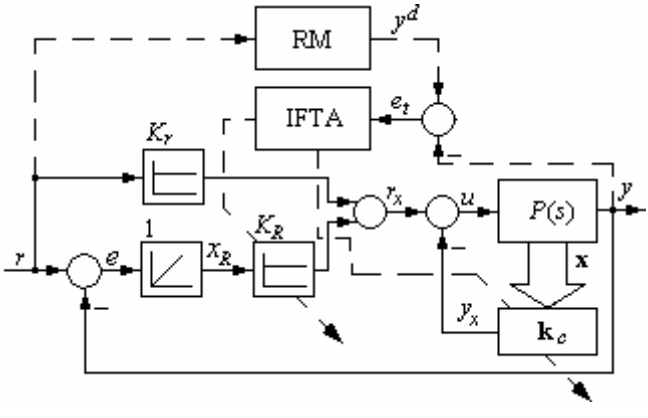


Fig. 1. IFT-based state feedback control system structure with integrator in the state feedback controller

The main variables and blocks in Fig. 1 are: IFTA – the IFT algorithm, RM – the reference model, r – the reference input, $e = r - y$ – the control error, $\mathbf{k}_c = [K_1 \quad K_2]$ – the state feedback gain matrix to be tuned by means of the IFT algorithm, $P(s) = P_{\alpha,u}(s)$ – the transfer function of the controlled plant with the controlled

output $y = x_1 = \alpha$, y^d – the reference model (desired) output, $e_t = y^d - y$ – the tracking error.

The state-space model (1) can be reconsidered by including one additional state variable $x_3 = x_R$ which corresponds to the integrator inserted to the state feedback controller. Thus its gain K_R will be subject to IFT as it is illustrated in Fig. 1. The extended state-space model of the process becomes then

$$\begin{aligned} \dot{\mathbf{x}}_E &= \begin{bmatrix} 0 & 1 & 0 \\ -K_s K_1 & -\frac{1}{T_s} - K_s K_2 & K_s K_R \\ -1 & 0 & 0 \end{bmatrix} \mathbf{x}_E + \begin{bmatrix} 0 \\ \frac{K_s}{T_s} K_r \\ 1 \end{bmatrix} r, \\ \mathbf{y} &= \mathbf{I}_3 \, \mathbf{x}_E, \end{aligned} \tag{3}$$

where $\mathbf{x}_E = [\alpha \quad \omega \quad x_R]^T = [\mathbf{x} \quad x_R]^T$ is the extended state vector, $\mathbf{I}_3$ is the second-order identity matrix, $\mathbf{y} = [y_1 \quad y_2 \quad y_3]^T$ is the controlled output vector, and the parameter K_r is not included in the tuning scheme. K_r is set prior to the application of IFT. One way to choose K_r is to keep a connection between the steady-state value of r and the steady-state value of r_x for which the desired r can be tracked by the steady-state value of y. That value of r_x can be subject to the experimental identification of the state feedback control system.

3 Convergent Iterative Feedback Tuning Algorithm for State Feedback Controllers

The objective function J defined over the finite time horizon N is

$$J(\boldsymbol{\rho}) = \frac{1}{2N} \sum_{t=1}^{N} (e_t(\boldsymbol{\rho}))^2 \ , \tag{4}$$

where $\boldsymbol{\rho} \in R^m$ is the parameter vector, which for $m = 2$ is $\boldsymbol{\rho} = [\rho_1 = K_1 \quad \rho_2 = K_2]^T = \mathbf{k}_c{}^T$.

The IFT algorithms [12]–[20] are applied to find the solution $\boldsymbol{\rho}^*$ to the optimization problem

$$\boldsymbol{\rho}^* = \arg\min_{\rho \in SD} J(\boldsymbol{\rho}) \ , \tag{5}$$

where several constraints can be imposed in relation with the controlled plant and the state feedback control system. One of these constraints concerns the stability of the system and *SD* stands for the stability domain with this regard [23], [24].

Solving the optimization problem (6) requires finding the parameter vectors that make the gradient $\frac{\partial J}{\partial \boldsymbol{\rho}}$ equal to zero:

$$\frac{\partial J}{\partial \boldsymbol{\rho}} = [\frac{\partial J}{\partial \rho_1} \quad \ldots \quad \frac{\partial J}{\partial \rho_m}]^T = \mathbf{0} \ . \tag{6}$$

Since the controlled output depends on $\boldsymbol{\rho}$ as $y(\boldsymbol{\rho})$ and y^d not, use is made of (4) and (6) becomes

$$\frac{1}{N}\sum_{t=1}^{N}\frac{\partial y}{\partial \boldsymbol{\rho}}^T [y(\boldsymbol{\rho})-y^d]=\mathbf{0}\ . \tag{7}$$

The partial derivatives $\frac{\partial y}{\partial \rho_i}$ should be calculated to obtain the components of the gradient, $\frac{\partial J}{\partial \rho_i}$, $i=\overline{1,m}$. The experiments specific to IFT are conducted to obtain those components. Use is made of the notation [20]

$$\alpha'=\frac{\partial \alpha}{\partial \rho_i} \tag{8}$$

to highlight the partial derivative of the variable α with respect to the parameter ρ_i, $i=\overline{1,m}$.

To derive the IFT algorithms the following relations can be extracted from the state feedback control system structure presented in Fig. 1:

$$\begin{aligned} &\mathbf{y}=\mathbf{P}\,u\ ,\\ &r_x=K_r r+K_R x_R\ ,\\ &u=K_r r+\mathbf{K}_c\mathbf{x}_E\ , \end{aligned} \tag{9}$$

where the extended state feedback gain matrix $\mathbf{K}_c$ is

$$\mathbf{K}_c=[-K_1 \quad -K_2 \quad K_R]=[-\mathbf{k}_c \quad K_R]\ . \tag{10}$$

The matrix $\mathbf{K}_c$ in (10) highlights the parameter vector $\boldsymbol{\rho}$ which for $m=3$ is

$$\boldsymbol{\rho}=[K_1 \quad K_2 \quad K_R]^T\ . \tag{11}$$

Since y and u are functions of $\boldsymbol{\rho}$ the first and third equations in (9) yield

$$\begin{aligned} &\mathbf{y}'=\mathbf{P}\,u'\ ,\\ &u'=\mathbf{K}_c'\mathbf{x}_E+\mathbf{K}_c\mathbf{x}_E'\ . \end{aligned} \tag{12}$$

Next the second equation in (3) enables the following transformation of the second equation in (12):

$$u'=\mathbf{K}_c'\mathbf{y}+\mathbf{K}_c\mathbf{y}'\ . \tag{13}$$

The first term in the right-hand side of (13), $\mathbf{K}_c'\mathbf{y}$, needs to be added to the control signal to obtain the experimental scheme. That term contains the unmodified output vector (in the MIMO framework) obtained from the first experiment [13]. The second term in the right-hand side of (13), $\mathbf{K}_c\mathbf{y}'$, is measured from the state feedback control

system structure. Therefore the experimental scheme to calculate the gradients is presented in Fig. 2 where the blocks RM and IFTA (in Fig. 1) are omitted for simplicity.

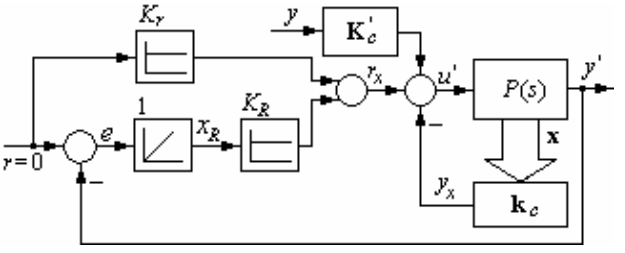


Fig. 2. Experimental scheme to calculate the gradients in the IFT-based state feedback control system structure

The first experiment specific to IFT, referred to also as the normal one, is conducted with the control system structure presented in Fig. 1 to measure the controlled output y. The next $m = 3$ experiments, i.e. the gradient experiments, are conducted with the experimental scheme presented in Fig. 2. These experiments are done separately for each parameter in $\mathbf{K}_c$ considering the zero values of the other $m-1=2$ parameters because their derivatives with respect to the currently considered parameter are zero.

The parameter vector must be updated after the experiments are finished. Newton's algorithm is generally used as a convenient technique which iteratively approaches a zero of a function without knowing its expression. The update law to calculate the next parameter vector $\boldsymbol{\rho}^{i+1}$ is

$$\boldsymbol{\rho}^{i+1} = \boldsymbol{\rho}^{i} - \gamma_i \, \mathbf{R}_i^{-1} \, est[\frac{\partial J}{\partial \boldsymbol{\rho}}(\boldsymbol{\rho}^{i})] \ , \tag{14}$$

where i is the index of the current iteration / experiment, $\gamma_i > 0$ is the step size, $est[\frac{\partial J}{\partial \boldsymbol{\rho}}(\boldsymbol{\rho}^{i})]$ is the estimate of the gradient, and the regular matrix $\mathbf{R}_i$ can be the estimate of the (positive definite) Hessian matrix or the identity matrix.

Popov's hyperstability analysis results will be applied as follows to the parameter update law (14) in order to derive a condition to guarantee the convergence of the IFTAs. First the domain of attraction of the update law as part of an IFTA is defined in terms of the following definition which is equivalent to that presented in [25].

Definition 1. Let $\boldsymbol{\rho}^{*}$ be the global minimum of the function $J : R^m \to R^+$ defined in (4). The set $\Pi \subset R^m$ is called a domain of attraction of the update law (14) for the function J if

$$\lim_{i \to \infty} \boldsymbol{\rho}^{i} = \boldsymbol{\rho}^{*}, \ \forall \boldsymbol{\rho}^{0} \in \Pi \ , \tag{15}$$

where $\boldsymbol{\rho}^{0}$ is the initial parameter vector.

The convergence result is next expressed in terms of Theorem 1.

Theorem 1. Let $i_1 \geq 0$ be an arbitrary index of experiments / iterations and $\varepsilon_0 = \text{const}$, $\varepsilon_0 \neq 0$. If the condition

$$\Pi = \left\{ \boldsymbol{\rho} \middle| I(i_1) = \sum_{j=0}^{i_1} \gamma_j (est[\frac{\partial J}{\partial \boldsymbol{\rho}}(\boldsymbol{\rho}^j)])^T (\mathbf{R}_j^{-1})^T \boldsymbol{\rho}^j \geq -\varepsilon_0^2, \ \forall i_1 \geq 0 \right\} \tag{16}$$

is satisfied for the isolated global minimum $\boldsymbol{\rho}^*$ then Π is a domain of attraction of $\boldsymbol{\rho}^*$.

Proof. The relationship (14) is expressed as follows as a dynamical feedback MIMO system in the parameter space and iteration domain in terms of the structure presented in Fig. 3, where the feedforward discrete-time linear time-invariant (LTI) block is

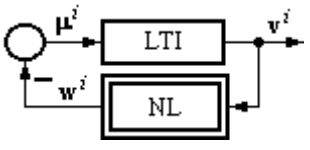


Fig. 3. Dynamical feedback MIMO system structure of (14) used in the convergence analysis

$$\begin{aligned} &\boldsymbol{\rho}^{i+1} = \mathbf{A}\,\boldsymbol{\rho}^i + \mathbf{B}\,\boldsymbol{\mu}^i, \\ &\mathbf{v}^i = \mathbf{C}\,\boldsymbol{\rho}^i + \mathbf{D}\,\boldsymbol{\mu}^i, \\ &\mathbf{A} = \mathbf{B} = \mathbf{C} = \mathbf{I}, \ \mathbf{D} = \mathbf{0}, \end{aligned} \tag{17}$$

and the nonlinear (NL) feedback block is

$$\mathbf{w}^i = \gamma_i \, \mathbf{R}_i^{-1} est[\frac{\partial J}{\partial \mathbf{v}}(\mathbf{v}^i)] \cdot \tag{18}$$

LTI is completely controllable and completely observable. The discrete transfer function matrix of the LTI, $\mathbf{H}(z)$, obtains the expression

$$\begin{aligned} &\mathbf{H}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} = (z\mathbf{I} - \mathbf{I})^{-1} \\ &= \text{diag}(1/(z-1), \quad 1/(z-1), \quad ..., \quad 1/(z-1)), \end{aligned} \tag{19}$$

and it is a positive real discrete transfer function matrix. Since NL satisfies the Popov type inequality.

$$I(i_1) = \sum_{j=0}^{i_1} (\mathbf{w}^j)^T \mathbf{v}^j \geq -\varepsilon_0^2, \ \forall i_1 \geq 0 \ , \tag{20}$$

which is equivalent to (16), the conditions for the hyperstability of the system (17), (18) are fulfilled [21], [22]. Thus the convergence of the IFTAs with the update law (14) is guaranteed.

The new family of IFTAs dedicated to the considered class of stat feedback control system consists of the following steps.

Step 1. Conduct the normal experiment making use of the control system structure presented in Fig. 1, measure $\mathbf{y}(\boldsymbol{\rho}^i) = \mathbf{x}$ and calculate the reference model output vector $\mathbf{y}^d$. Next conduct the three gradient experiments in terms of the experimental scheme presented in Fig. 2 and measure the outputs that give the gradient of the controlled output $\frac{\partial \mathbf{y}}{\partial \boldsymbol{\rho}}(\boldsymbol{\rho}^i)$.

Step 2. Calculate the estimate of the gradient of the objective function

$$est[\frac{\partial J}{\partial \boldsymbol{\rho}}(\boldsymbol{\rho}^i)] = \frac{1}{N}\sum_{i=1}^{N}[\frac{\partial \mathbf{y}}{\partial \boldsymbol{\rho}}(\boldsymbol{\rho}^i)]^T[\mathbf{y}(\boldsymbol{\rho}^i) - \mathbf{y}^d] \cdot \tag{21}$$

Step 3. Calculate the Popov sums $I(i_1)$ in (16) and set the step size γ_i such that to fulfill the convergence condition (16) for any $i_1,\ 0 \le i_1 \le i$ and any $\varepsilon_0 = \text{const},\ \varepsilon_0 \ne 0$.

Step 4. Calculate the next set of parameters $\boldsymbol{\rho}^{i+1}$ according to the parameter update law (14).

4 Implementation Issues

Setting the reference model (Fig. 1) is important. This aspect should be related to the imposed control system performance indices and the accepted model of the controlled plant although it is achievable. Over ambitious reference models can results in the impossible fulfillment of the performance specifications.

The first task of the state feedback controller is to ensure an initially stable control system. The pole placement design can be used with this regard in order to set the initial parameter vector $\boldsymbol{\rho}^0$.

The identity matrix is recommended to play the role of the positive definite matrix $\mathbf{R}_i$ in (14) for low-cost implementations. Another alternative is to calculate the estimate of the Hessian matrix to play the role of $\mathbf{R}_i$. That calculation should be included in the step 3 of the IFTAs and an additional experiment can be employed with this regard.

In many cases the actuator is characterized by a nonlinear input-output map caused by the actuator saturation. That is a problem because it introduces usually nonlinear behaviors in the plant's dynamics, thus it must be avoided. When making use of the integrator in the controller the importance of the actuator saturation is increased because actuators which enter deep saturation regions require usually longer time periods to re-enter the active regions of normal operation.

The structure illustrated in Fig. 2 highlights that when the state vector is fed over the control signal it may cause the saturation effects. Therefore the experiments will provide estimates of the gradients which are different with respect to the correct ones.

For the sake of simplicity an actuator with the active input range within −1 to +1 is considered as follows. One solution to solve the actuator saturation problem is to design the experiments such that the actuator does not enter saturation. Therefore the injected variable (in Fig. 2) must be in the active region of the actuator's input-output

static map. The injected variable can be scaled to its maximum value M, $M > 0$, of its measured dynamics. Generally speaking for an injected variable z_t, $t = \overline{1,N}$, its scaled value added to the control signal is

$$(z_s)_t = z_t / M,\ M = \max_{k=\overline{1,N}} | z_t | \cdot \tag{22}$$

Therefore it is guaranteed that the new variable to be injected, $(z_s)_t$, is within the accepted domain of the actuator input. Some details concerning the way that the gradient experiments are influenced are offered in [20].

5 Case Study and Real-Time Experimental Results

The validation of the new state feedback control solution and IFT algorithms is done in terms of a case study dedicated to the position control of a DC servo system with backlash. The experimental setup illustrated in Fig. 4 is built around the INTECO DC motor laboratory equipment. It makes use of an optical encoder for the angle measurement and a tacho-generator for the measurement of the angular speed. The tacho-generator measurements are very noisy. The speed can also be observed from the angle measurements. The control system performance indices such as settling time and overshoot can be assessed easily.

It is accepted that $y = x_1 = \alpha$ in relation with (1) and (3) and Fig. 1. The plant is characterized by the parameters $K_s = 139.88$ and $T_s = 0.9198$ s, obtained in terms of experimental identification. Part of the real-time experiments is presented here and it makes of the initial parameter vector set to set to $\boldsymbol{\rho}^0 = [0.0132 \quad 0.0126 \quad 0.005]^T$ in order to stabilize the control system.

A $r = 150$ rad step type modification of the reference input was applied. Therefore the parameter K_r was tuned at the value $K_r = 0.0133$ and dropped out of the objective function (4). That value of K_r was obtained by steady-state calculations to ensure a desired gain between r and *y*. The sampling period was set to 0.01 s. The continuous-time RM is characterized by the transfer function

$$G_{RM}(s) = 1/(s^2 + 1.5s + 1) \tag{23}$$

which is next discretized.

Fig. 4. Experimental setup

The real-time experimental results that illustrate the behavior of the control system in terms of the evolution of the controlled output before the application of the IFT algorithm are presented in Fig. 5.

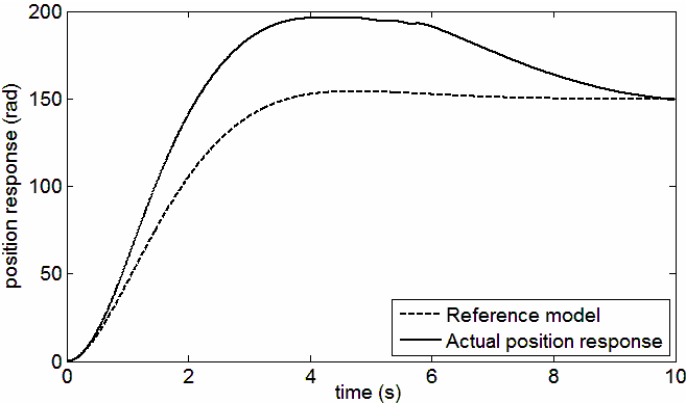


Fig. 5. y^d (dotted line) and y (continuous line) versus time considered before IFT

The IFTA presented in Section 3 was applied. The parameters in the IFTA were set to $\gamma_i = 0.0001$ and $\mathbf{R}_i = \mathbf{I}_3$. The evolution of the controlled output after 12 iterations is presented in Fig. 6.

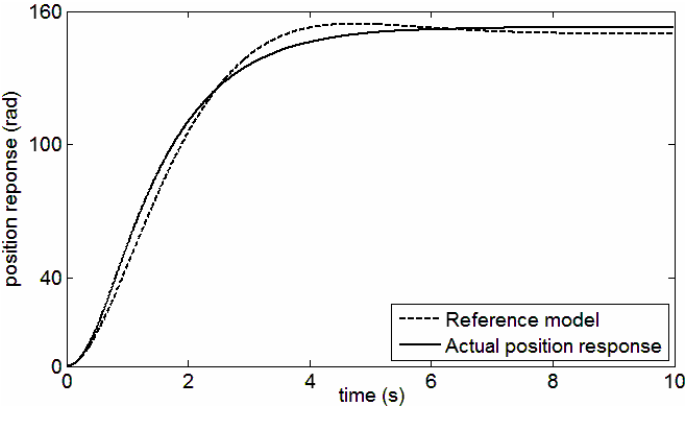


Fig. 6. y^d (dotted line) and y (continuous line) versus time considered after IFT

The control system performance enhancement due to IFT is highlighted. It is reflected by smaller overshoot and settling time. The control system performance enhancement is also pointed out in the evolutions of the control signal and speed $x_2 = \omega$ illustrated in Fig. 7 and Fig. 8, respectively. The oscillations in the control signal and speed are caused by the backlash.

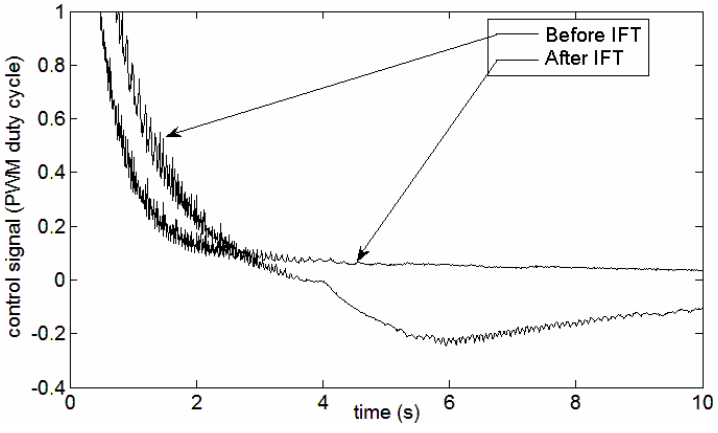


Fig. 7. u versus time considered before and after IFT

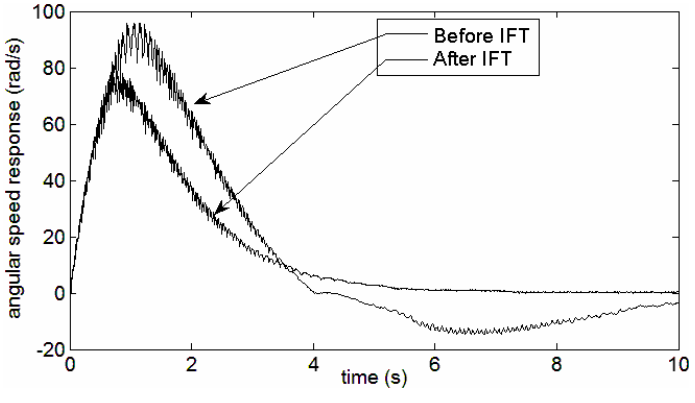


Fig. 8. x_2 versus time considered before and after IFT

6 Conclusions

This paper has presented a new approach to the convergent IFT-based design of a class of state feedback control systems meant for a class of second-order systems with integral component. The convergent IFT algorithm proposed here is sufficiently general to be applied without additional difficulties to the state feedback control of systems of arbitrary order.

The real-time experimental results in the case study validate the optimal state feedback control solution offered in this paper. It shows the power of IFT because the IFT-based design of the control system offers better performance indices compared to the situation prior to the application of the IFT algorithm. The performance indices are very good for the nonlinear controlled plant although the theoretical approach is based on the linear or linearized mathematical model of the controlled plant.

The future research will compensate for the limitations of the theoretical approach presented here. They involve the initial controller tuning and the guarantee of obtaining of the global optimum after a relatively small number of iterations. Quasi-optimal state feedback control systems can be designed currently. Emphasis will be given to the transfer of the results from the linear case to the fuzzy one and to the extension to other classes of controlled plants [26]–[32].

Acknowledgements. This work was supported by the CNCSIS and CNMP of Romania and the co-operation between the Óbuda University, Budapest, Hungary, the University of Ljubljana, Slovenia, and the "Politehnica" University of Timisoara, Romania, in the framework of the Hungarian-Romanian and Slovenian-Romanian Intergovernmental Science & Technology Cooperation Programs. This work was partially supported by the strategic grant POSDRU 6/1.5/S/13 (2008) of the Ministry of Labor, Family and Social Protection, Romania, co-financed by the European Social Fund – Investing in People.

References

1. Blažič, S., Škrjanc, I.: Design and Stability Analysis of Fuzzy Model-Based Predictive Control - A Case Study. J. Intell. Robot. Syst. 49, 279–292 (2007)
2. Petres, Z., Baranyi, P., Korondi, P., Hashimoto, H.: Trajectory Tracking by TP Model Transformation: Case Study of a Benchmark Problem. IEEE Trans. Ind. Electron. 54, 1654–1663 (2007)
3. Vaščák, J.: Navigation of Mobile Robots Using Potential Fields and Computational Intelligence Means. Acta Polytech. Hung. 4, 63–74 (2007)
4. Orlowska-Kowalska, T., Szabat, K.: Damping of Torsional Vibrations in Two-Mass System Using Adaptive Sliding Neuro-Fuzzy Approach. IEEE Trans Ind. Informat. 4, 47–57 (2008)
5. Barut, M., Bogosyan, S., Gokasan, M.: Experimental Evaluation of Braided EKF for Sensorless Control of Induction Motors. IEEE Trans. Ind. Electron. 55, 620–632 (2008)
6. Baranyi, P., Korondi, P., Tanaka, K.: Parallel Distributed Compensation Based Stabilization of a 3-DOF RC Helicopter: A Tensor Product Transformation Based Approach. J. Adv. Comput. Intell. Intell. Inform. 13, 25–34 (2009)
7. Klančar, G., Matko, D., Blažič, S.: Wheeled Mobile Robots Control in a Linear Platoon. J. Intell. Robot. Syst. 54, 709–731 (2009)
8. Hladek, D., Vaščák, J., Sinčák, P.: Multi-Robot Control System for Pursuit-Evasion problem. J. Electr. Eng. 60, 143–148 (2009)
9. Åström, K.J., Hägglund, T.: Benchmark Systems for PID Control. In: IFAC PID 2000 Workshop, Terrassa, Spain, pp. 181–182 (2000)
10. Isermann, R.: Mechatronic Systems: Fundamentals. Springer, Heidelberg (2003)
11. Horváth, L., Rudas, I.J.: Modeling and Problem Solving Methods for Engineers. Academic Press, Elsevier, Burlington, MA (2004)
12. Hjalmarsson, H., Gunnarsson, S., Gevers, M.: A Convergent Iterative Restricted Complexity Control Design Scheme. In: 33rd IEEE Conference on Decision and Control, Lake Buena Vista, FL, pp. 1735–1740 (1994)
13. Hjalmarsson, H., Gevers, M., Gunnarsson, S., Lequin, O.: Iterative Feedback Tuning: Theory and Applications. IEEE Control Syst. Mag. 18, 26–41 (1998)

14. Huusom, J.K., Poulsen, N.K., Jørgensen, S.B.: Improving Convergence of Iterative Feedback Tuning. J. Process Control 19, 570–578 (2009)
15. Hjalmarsson, H., Birkeland, T.: Iterative Feedback Tuning of Linear Time-Invariant MIMO Systems. In: 37th IEEE Conference on Decision and Control CDC, Tampa, FL, pp. 3893–3898 (1998)
16. Hjalmarsson, H.: Efficient Tuning of Linear Multivariable Controllers Using Iterative Feedback Tuning. Int. J. Adapt. Control Signal Process. 13, 553–572 (1999)
17. Jansson, H., Hjalmarsson, H.: Gradient Approximations in Iterative Feedback Tuning for Multivariable Processes. Int. J. Adapt. Control Signal Process. 18, 665–681 (2004)
18. Huusom, J.K., Poulsen, N.K., Jørgensen, S.B.: Data Driven Tuning of State Space Control Loops with Unknown State Information and Model Uncertainty. Comput. Aided Chem. Eng. 26, 441–446 (2009)
19. Kissling, S., Blanc, P., Myszkorowski, P., Vaclavik, I.: Application of Iterative Feedback Tuning (IFT) to Speed and Position Control of a Servo Drive. Control Eng. Pract. 17, 834–840 (2009)
20. Rădac, M.-B., Precup, R.-E., Petriu, E.M., Preitl, S., Dragoş, C.-A.: Iterative Feedback Tuning Approach to a Class of State Feedback-Controlled Servo Systems. In: 6th International Conference on Informatics in Control, Automation and Robotics ICINCO 2009, Milan, Italy, vol. 1, pp. 41–48 (2009)
21. Popov, V.M.: Hyperstability of Automatic Control Systems. Springer, New York (1973)
22. Landau, Y.-D.: Adaptive Control: The Model Reference Approach. Marcel Dekker, New York (1979)
23. Precup, R.-E., Preitl, S., Rudas, I.J., Tomescu, M.L., Tar, J.K.: Design and Experiments for a Class of Fuzzy Controlled Servo Systems. IEEE/ASME Trans. Mechatronics 13, 22–35 (2008)
24. Precup, R.-E., Preitl, S., Tar, J.K., Tomescu, M.L., Takács, M., Korondi, P., Baranyi, P.: Fuzzy Control System Performance Enhancement by Iterative Learning Control. IEEE Trans. Ind. Electron. 55, 3461–3475 (2008)
25. Eckhard, D., Bazanella, A.S.: Optimizing the Convergence of Data-Based Controller Tuning. In: European Control Conference ECC 2009, Budapest, Hungary, pp. 910–915 (2009)
26. Savaresi, S.M., Tanelli, M., Taroni, F., Previdi, F., Bittanti, S., Prandoni, V.: Analysis and Design of an Automatic Motion-Inverter. IEEE/ASME Trans. Mechatronics 11, 346–357 (2006)
27. Johanyák, Z.C., Kovács, S.: A Brief Survey and Comparison on Various Interpolation Based Fuzzy Reasoning Methods. Acta Polytech. Hung. 3, 91–105 (2006)
28. Johanyák, Z.C., Tikk, D., Kovács, S., Wong, K.K.: Fuzzy Rule Interpolation Matlab Toolbox - FRI Toolbox. In: 15th International Conference on Fuzzy Systems FUZZ-IEEE 2006, Vancouver, BC, Canada, pp. 1427–1433 (2006)
29. Barai, R.K., Nonami, K.: Optimal Two-Degree-of-Freedom Fuzzy Control for Locomotion Control of a Hydraulically Actuated Hexapod Robot. Inf. Sci. 177, 1892–1915 (2007)
30. Hwang, C.-L.: Microprocessor-Based Fuzzy Decentralized Control of 2-D Piezo-Driven Systems. IEEE Trans. Ind. Electron. 55, 1411–1420 (2008)
31. Wu, M., Yan, J., She, J.-H., Cao, W.-H.: Intelligent Decoupling Control of Gas Collection Process of Multiple Asymmetric Coke Ovens. IEEE Trans. Ind. Electron. 56, 2782–2792 (2009)
32. Ahn, H.K., Anh, H.P.H.: Inverse Double NARX Fuzzy Modeling for System Identification. IEEE/ASME Trans. Mechatronics 15, 136–148 (2010)

Multimodel Gain Scheduled Quadratic Controller for Variable-Speed Wind Turbines Performances Improvement*

Nadhira Khezami[1,2], Xavier Guillaud[2], and Naceur Benhadj Braiek[1]

[1] LECAP, École Polytechnique de Tunisie, BP 743 – 2078, La Marsa, Tunisia
[2] L2EP, École Centrale de Lille, Cité Scientifique, 59651 Villeneuve d'Ascq Cedex, France
{nadhira.khezami,xavier.guillaud}@ec-lille.fr,
naceur.benhadj@ept.rnu.tn

Abstract. This paper deals with the design of a linear quadratic (LQ) control law combined with a multimodel approach for variable-speed, variable-pitch wind turbines in order to improve their performances, especially the system global dynamic. The control law is based on a linearization of the system around a selected set of operating points. Hence, a set of linear local models (sub-models) is defined to describe the wind turbine functioning in the high wind speeds operating range. A state observer is used to estimate the immeasurable state variables. Thereafter, a global asymptotic stability analysis is developed by solving a bilinear matrix inequality (BMI) feasibility problem based on the local stability of the sub-models.

Keywords: LQ Controller, Multimodel Approach, Global Stability, LMI, Ly Apunov Equations.

1 Introduction

Nowadays, the growth of the utilization of the wind turbines is more and more important since they are producing carbon-emission-free electricity. Until today, only classic control laws, such as P, PI or PID controllers, are used in the wind turbines. However, the performance of these controllers is limited by the high nonlinear characteristics of the wind turbine and by the appearance of new control objectives required by the grid-codes; the reason why advanced control research area is improving every day.

In the first axis of this paper, an LQ controller, which had been advocated by many researchers, is designed with a multimodel approach, for pitch regulated variable speed wind turbines operating at high wind speeds, in order to guarantee an optimal behavior for the studied process. However, this technique still presents some limits to satisfy all the control objectives especially those concerning the system global dynamic. This paper aims then to present an issue for this problem by adding an exponential term in the quadratic cost function.

*This work was originally presented in the ICINCO'09 Conference, Milan, 2009.
This work was supported by the CMCU project number: **08G1120**.

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 113–127.
springerlink.com

The second section deals with the asymptotic stability analysis of the global sys-tem by solving a set of BMI according to the Lyapunov theorems. In fact, the stability study is necessary and important to illustrate the effectiveness of the presented strategy.

Finally, the simulation results realized on Matlab Simulink are presented and discussed.

2 Wind Turbine Modelling

2.1 Wind Turbine Description

The considered wind turbine (Figure 1) is modeled as two inertias (the generator and the turbine inertias respectively J_g and J_T) linked to a flexible shaft with a mechanical coupling damping coefficient d and a mechanical coupling stiffness coefficient k. This model is widely used in the literature [1]; [2].

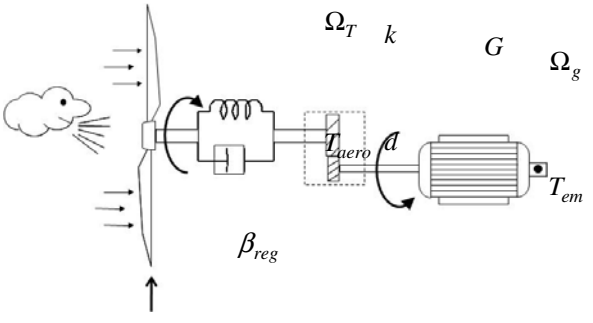


Fig. 1. Wind turbine dynamic model

where Ω_T and Ω_g are the turbine and the generator rotational speeds, T_{em} is the generator torque, T_{mec} is the drive train mechanical torque and T_{aero} is the torque caught by the wind turbine which is expressed by:

$$T_{aero} = \tfrac{1}{2} \cdot \frac{\rho \cdot \pi \cdot R^5 \cdot \Omega_T^2}{\lambda^3} . c_p(\lambda, \beta) \tag{1}$$

where ρ is the air density and R is the turbine radius.

The power coefficient c_p (Figure 2) is a non linear function of the blade pitch β and the tip speed ratio λ depending on the wind speed value v and given by:

$$\lambda = \frac{\Omega_T \cdot R}{v} \tag{2}$$

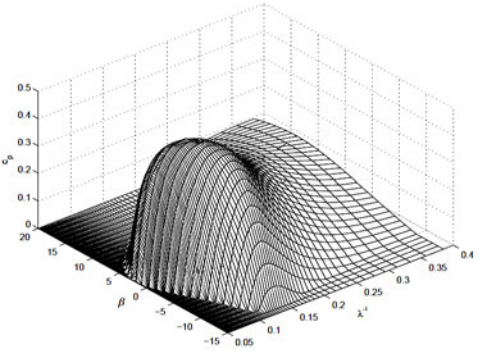


Fig. 2. Power coefficient curves

The dynamic response of the rotor is given by:

$$J_T.\dot{\Omega}_T = T_{aero} - T_{mec} \tag{3}$$

The generator is driven by the mechanical torque and braked by the electromagnetic torque. Reported to the low speed shaft, the characteristic equation is the following:

$$J_{g-ls}.\dot{\Omega}_{g-ls} = T_{mec} - G.T_{em} \tag{4}$$

where G is the gearbox gain and:

$$\begin{cases} \Omega_{g-ls} = \dfrac{\Omega_g}{G} \\ J_{g-ls} = G^2.J_g \end{cases} \tag{5}$$

And the low speed shaft torque T_{mec} results from the torsion and friction effects due to the difference between the generator and the rotor speeds [3]. It's defined by the following equation reported to the low speed shaft:

$$\dot{T}_{mec} = k.\left(\Omega_T - \Omega_{g-ls}\right) + d.\left(\dot{\Omega}_T - \dot{\Omega}_{g-ls}\right) \tag{6}$$

The pitch actuator dynamic is described by a first order system:

$$\dot{\beta} = \frac{1}{\tau_\beta}.\left(\beta_{ref} - \beta\right) \tag{7}$$

β_{ref} represents the control value of the blade-pitch angle β and τ_β is the time constant of the pitch actuator.

2.2 Linearization and State Representation

The wind turbine is a complex non linear system presenting several difficulties in study and control. It seems then more suitable to describe it with a set of linear local models valid in different operating points corresponding to different levels of wind speed values. The principle of this method is used in several techniques. In this paper, we use the multimodel approach which was the subject of many research works [4].

For the studied system, we define a multimodel base made of four local models. The equivalent instantaneous model, as described in Figure 3, is obtained by a fusion of only two valid successive models. The choice of these models depends on the wind speed value.

The weighting coefficient μ_i is the validity value of the model M_i and it can be expressed by:

$$\mu_i = 1 - r_i \tag{8}$$

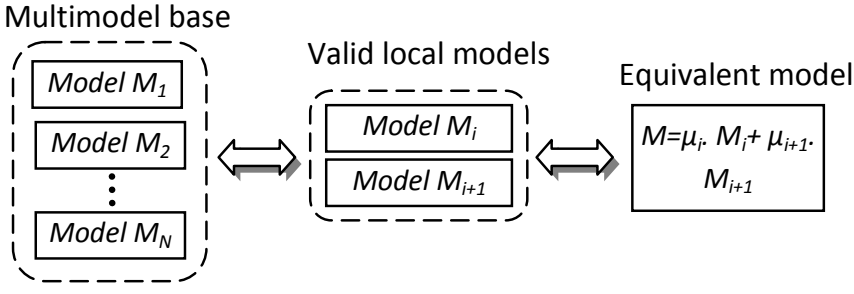


Fig. 3. Wind-turbine multimodel description

r_i is a normalized residue measuring the error between the instantaneous and the valid local model wind speed values (respectively v and v_i). When M_i and M_{i+1} are the valid models, the residue can be expressed as:

$$r_i = \frac{|v_i - v|}{v_i + v_{i+1}} \tag{9}$$

Thus, the validities satisfy the convex sum, such that: $\mu_i + \mu_{i+1} = 1$.

To obtain the local models, the system should be then linearized around the operating point. The non-linearity of the system is due to the c_p characteristic which is used in the expression of the aerodynamic torque. We need then to linearize the expression (1) of T_{aero} around an operating point (*o.p*) defined by the wind speed value $v_{i\,l}$ [1]; [7]. We can define:

$$\Delta T_{aero} = \left.\frac{\partial T_{aero}}{\partial \Omega_T}\right|_{o.p} .\Delta\Omega_T + \left.\frac{\partial T_{aero}}{\partial \beta}\right|_{o.p} .\Delta\beta = a_i .\Delta\Omega_T + b_i .\Delta\beta \tag{10}$$

where:

$$\begin{cases} a_i = \frac{1}{2}.\rho.\pi.R^3.\frac{v_i^{\,2}}{\Omega_{T\_nom}}.\left[\frac{\partial c_p(\lambda,\beta)}{\partial\lambda} - \frac{c_{p_{i\_nom}}}{\lambda_{i\_nom}}\right] \\ b_i = \frac{1}{2}.\rho.\pi.R^2.\frac{v_i^{\,3}}{\Omega_{T\_nom}}.\frac{\partial c_p(\lambda,\beta)}{\partial\beta} \end{cases} \tag{11}$$

and:

$$c_{pi\_nom} = c_p\left(\lambda_{i\_nom}, \beta_{i\_nom}\right) = \frac{2.\Omega_{T\_nom}.T_{aero\_nom}}{\rho.\pi.R^2.v_i^{\,3}} \tag{12}$$

The symbol Δ represents the deviation from the chosen operating point corresponding to (Ω_{T_nom}, $\Omega_{g\text{-}ls\text{-}nom}$, $\beta_{i\text{-}nom}$, $T_{mec\text{-}nom}$, $T_{em\text{-}nom}$ and P_{nom}) where: $T_{em\text{-}nom}$ and $T_{mec\text{-}nom}$ are respectively the nominal values of the electromagnetic and the mechanical torques.

Thereafter, the linearization of the non-linear system expressed in equations (3), (4), (6) and (7) around an operating point gives a state space representation of the form below:

$$\begin{cases} \dot{x} = A_i.x + B_i.u \\ y = C_i.x + D_i.u \end{cases} \tag{13}$$

where *x, u* and *y* are respectively the state, control and output vectors defined as:

$$x = \begin{bmatrix} \Delta\Omega_T \\ \Delta\Omega_{g-ls} \\ \Delta\beta \\ \Delta T_{mec} \end{bmatrix},\ y = \begin{bmatrix} \Delta\Omega_T \\ \Delta P \end{bmatrix} \quad \text{and} \quad u = \begin{bmatrix} \Delta\beta_{ref} \\ \Delta T_{em} \end{bmatrix} \tag{14}$$

Notice that $P=T_{em}.\Omega_g$ designates the generated electrical power. This leads to write around an operating point:

$$\Delta P = G.\left(T_{em\_nom}.\Delta\Omega_{g-ls} + \Delta T_{em}.\Omega_{g-ls\_nom}\right) \tag{15}$$

Hence, A_i, B_i, C_i and D_i, which are respectively the state, input, output and feedthrough matrices, are defined as follows:

$$\begin{cases} A_i = \begin{pmatrix} \frac{a_i}{J_T} & 0 & \frac{b_i}{J_T} & -\frac{1}{J_T} \\ 0 & 0 & 0 & \frac{1}{J_{g-ls}} \\ 0 & 0 & -\frac{1}{\tau_\beta} & 0 \\ k+\frac{a_i.d}{J_T} & -k & \frac{d.b_i}{J_T} & -d.\left(\frac{1}{J_T}+\frac{1}{J_{g-ls}}\right) \end{pmatrix}, B_i = \begin{pmatrix} 0 & 0 \\ 0 & -\frac{G}{J_{g-ls}} \\ \frac{1}{\tau_\beta} & 0 \\ 0 & \frac{d.G}{J_{g-ls}} \end{pmatrix} \\ C_i = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & G.T_{em\_nom} & 0 & 0 \end{pmatrix} \quad \text{and} \quad D_i = \begin{pmatrix} 0 & 0 \\ 0 & G.\Omega_{g-ls\_nom} \end{pmatrix} \end{cases} \tag{16}$$

3 Controller Design

The control task is based on the objective of regulating the rotor rotational speed and the generated power by acting on two control variables: the electromagnetic torque T_{em} and the regulating pitch angle β_{ref}.

The LQ control strategy had been advocated by many research works [3]; [8]; [9]; [10]; [11]. This technique presents a good compromise between the performances optimization and the minimization of the control signals by the use of a quadratic cost function. However, it also presents the disadvantage of the non possibility of controlling the global system dynamic. In this paper, a solution that can partially solve this problem is presented.

This controller aims to minimize the following quadratic criterion J :

$$J = \frac{1}{2}\int_0^{+\infty} \left(y^T.Q.y + u^T.R.u \right).e^{2\alpha t} dt \tag{17}$$

where Q and R are diagonal positive definite matrices.

The term $y^T.Q.y$ expresses the performances optimization, the term $u^T.R.u$ expresses the minimization of the control signals and the term $e^{2\alpha t}$ allows the performances improvement of the classic quadratic criterion. It leads to the placement of the system poles on the left of $-\alpha$. The criterion can be rewritten as follows with an input-state cross term:

$$J = \frac{1}{2}\int_0^{+\infty} \left(x^T.Q_1.x + 2.x^T.N.u + u^T.R_1.u \right).e^{2\alpha t} dt \tag{18}$$

where Q_1, R_1 and N are defined as:

$$\begin{cases} Q_1 = C^T.Q.C \\ R_1 = R + D^T.Q.D \\ N = C^T.Q.D \end{cases} \tag{19}$$

For this criterion, the optimal gain can be calculated from the following Riccati equation:

$$\begin{cases} {A_{i\alpha}}^T.L + L.A_{i\alpha} - \left(L.B_i + N_1\right).R_1^{-1}.\left(B_i^T.L + N_1^T\right) + Q_1 = 0 \\ A_{i\alpha} = A_i + \alpha.I \\ K_i = R_1^{-1}.\left(B_i^T.L + N_1^T\right) \end{cases} \tag{20}$$

where I is the identity matrix.

Since the dynamic of the pitch actuator should not be changed, the controller is designed in two steps. In the first step, we consider the blade pitch angle β and the electromagnetic torque T_{em} as control variables instead of β_{ref} and T_{em}. The state representation becomes then:

$$\begin{cases} \dot{x}_r = A_{i1.}x_r + B_{i1.}u_1 \\ y = C_{i1.}x_r + D_{i1.}u_1 \end{cases} \tag{21}$$

where:

$$x_r = \begin{bmatrix} \Delta\Omega_T \\ \Delta\Omega_{g-ls} \\ \Delta T_{mec} \end{bmatrix},\ y = \begin{bmatrix} \Delta\Omega_T \\ \Delta P \end{bmatrix} \text{ and } u_1 = \begin{bmatrix} \Delta\beta \\ \Delta T_{em} \end{bmatrix} \tag{22}$$

And:

$$\begin{cases} A_{i1} = \begin{pmatrix} \dfrac{a_i}{J_T} & 0 & -\dfrac{1}{J_T} \\ 0 & 0 & \dfrac{1}{J_{g-ls}} \\ k + \dfrac{a_i.d}{J_T} & -k & -d.\left(\dfrac{1}{J_T} + \dfrac{1}{J_{g-ls}}\right) \end{pmatrix}, B_{i1} = \begin{pmatrix} \dfrac{b_i}{J_T} & 0 \\ 0 & -\dfrac{G}{J_{g-ls}} \\ \dfrac{d.b_i}{J_T} & \dfrac{d.G}{J_{g-ls}} \end{pmatrix} \\ C_{i1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & G.T_{em-nom} & 0 \end{pmatrix} \text{ and } D_{i1} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{cases} \tag{23}$$

From this representation, the optimal gain $K_{i1} = \begin{bmatrix} K_{i1\_\beta} \\ K_{i1\_Tem} \end{bmatrix}$ is calculated such that:

$$\begin{cases} u_1 = \begin{bmatrix} \Delta\beta \\ \Delta T_{em} \end{bmatrix} = -K_{i1}.x_r \end{cases} \tag{24}$$

The relation (24) leads to the following optimal control law using the global state vector as shown in Figure 4:

$$\begin{cases} u = -K_i.x \\ K_i = \begin{bmatrix} K_{i\_\beta ref} \\ K_{i\_Tem} \end{bmatrix} \end{cases} \tag{25}$$

with:

$$\begin{cases} K_{i\_\beta ref} = \left(K_{i1\_\beta} + \tau_\beta.K_{i1\_\beta}.A_{i1}\right).T_1 + \tau_\beta.K_{i1\_\beta}.B_{i1\_\beta}.T_2 - \tau_\beta.K_{i1\_\beta}.B_{i1\_Tem}.K_{i1\_Tem}.T_1 \\ K_{i\_Tem} = K_{i1\_Tem}.T_1 \\ \begin{bmatrix} B_{i1\_\beta} & B_{i1\_Tem} \end{bmatrix} = B_{i1} \\ T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } T_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \end{cases} \tag{26}$$

➢ ***Remark:***

We notice that the mechanical torque is an immeasurable variable. It seems necessary then to use a multimodel state observer to estimate its value [8].

We can write:

$$\begin{cases} x = \begin{bmatrix} w = \begin{pmatrix} \Delta\Omega_T \\ \Delta\Omega_{g-ls} \\ \Delta\beta \end{pmatrix} \\ x_4 = \Delta T_{mec} \end{bmatrix} \\ \begin{pmatrix} \dot{w} \\ \dot{x}_4 \end{pmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}.\begin{pmatrix} w \\ x_4 \end{pmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}.u \end{cases} \tag{27}$$

The estimated value $\hat{x}_4 = \Delta\hat{T}_{mec}$ of $x_4 = \Delta T_{mec}$ can be designed in the following form:

$$\begin{cases} \dot{z} = \hat{A}_{21}.w + \hat{A}_{22}.z + \hat{B}_2.u \\ \hat{x}_4 = z + L.w \end{cases} \tag{28}$$

The following relations are used in order to calculate the observer matrices making the observing error $\varepsilon = x_4 - \hat{x}_4$ converge into zero:

$$\begin{cases} \hat{A}_{21} = A_{21} + \hat{A}_{22}.L - L.A_{11} \\ \hat{A}_{22} = A_{22} - L.A_{12} \\ \hat{B}_2 = B_2 - L.B_1 \end{cases} \tag{29}$$

The observing gain L is chosen by pole placement in order to give the observer a faster dynamic than the system's one.

Then, around a given operating point, the control law is calculated as given in (30):

$$\begin{cases} u = -K_i.\hat{x} \\ \hat{x} = \begin{bmatrix} \Delta\Omega_T \\ \Delta\Omega_{g-ls} \\ \Delta\beta \\ \Delta\hat{T}mec \end{bmatrix} \end{cases} \tag{30}$$

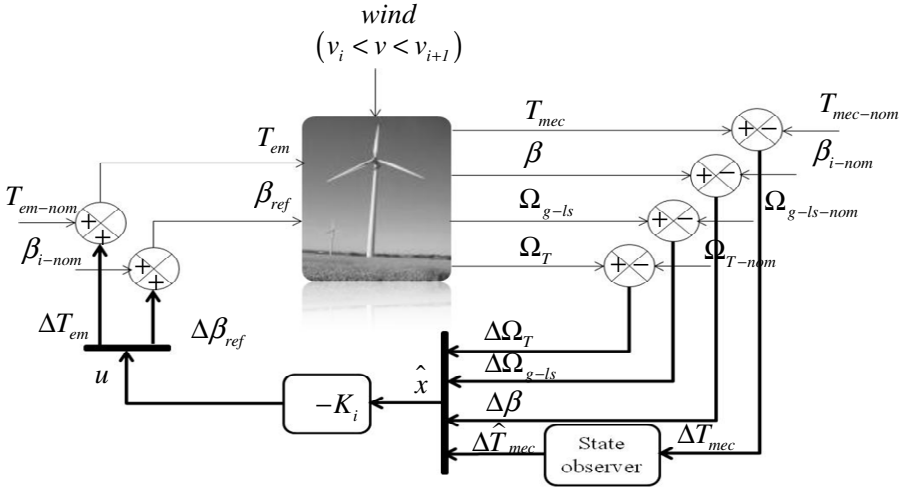


Fig. 4. The LQ controller design

4 Stability Study

The quantum advance in stability theory that allowed one the analysis of arbitrary differential equations is due to Lyapunov, who introduced the basic idea and the

definitions of stability that are in use today. The concept of Lyapunov stability plays an important role in control and system theory.

In the following study, the multimodel state observer is not taken into consideration.

As we define a global model M by fusion of two successive local models M_i and M_{i+1}, the characteristic matrices of the system (13) can be obtained by:

$$\begin{cases} A_i = \mu_k . A_k + \mu_{k+1} . A_{k+1} \\ B_i = \mu_k . B_k + \mu_{k+1} . B_{k+1} \\ C_i = \mu_k . C_k + \mu_{k+1} . C_{k+1} \\ D_i = \mu_k . D_k + \mu_{k+1} . D_{k+1} \end{cases} \tag{31}$$

The input vector is calculated by:

$$u = -\left(\mu_k . K_k + \mu_{k+1} . K_{k+1} \right) . x \tag{32}$$

Hereafter, the state vector can be represented as:

$$\begin{aligned} \dot{x} &= \sum_i \sum_j \mu_i . \mu_j . \left(A_i - B_i . K_j \right) . x \\ &= \sum_i \mu_i^2 . G_{ii} . x + 2 . \sum_{j=i+1} \mu_i . \mu_j . \left(\frac{G_{ij} + G_{ji}}{2} \right) . x \end{aligned} \tag{33}$$

where: $G_{ij} = A_i - B_i . K_j$

To study the global asymptotic stability of the above system supplied by the multimodel LQ control, the first necessity is to analyze the stability of every local model. As we focus here especially on the closed-loop system, the criterion of stabilization consists then in finding, for a local model M_i, a positive definite matrix P that satisfies the following LMI [Chedli, 2002; Liberzon and Morse, 1999]:

$$G_{ii}^T . P + P . G_{ii} < 0 \tag{34}$$

In the case of the multimodel systems, an extra condition is to add to the LMI (34) in order to guarantee the global stability [12]; [13]; [14] and it consists in:

$$Q_{ij}^T . P + P . Q_{ij} < 0, \quad i < j \tag{35}$$

where $Q_{ij} = \dfrac{G_{ij} + G_{ji}}{2}$

And in our case, only two successive local models are valid at a time, which means that this condition will be considered for $i=1$ *to* 3 and $j=i+1$.

5 Simulation Results

The proposed control approach and the stability analysis of the controlled system have been illustrated through simulations on Matlab Simulink.

The simulated wind turbine parameters are presented in Table 1.

Table 1. Wind turbine parameter values

Parameters		*Values*
Air density	ρ	*1,22 Kg/m³*
Turbine radius	R	*40m*
Nominal power	P_{nom}	*2MW*
Nominal speed	$\Omega_{T\text{-}nom}$	*18 rpm*
Optimal power coefficient	$c_{p\text{-}opt}$	*0.4775*
Optimal speed ratio	λ_{opt}	*9*
Gearbox gain	G	*92.6*
Turbine inertia	JT	*4.9×10^{6} N.m.s²*
Generator inertia	Jg	*0.9×10^{6} N.m.s²*
Mechanical coupling damping coefficient	d	*3.5×10^{5} N.m^{-1}.s*
Mechanical coupling stiffness coefficient	k	*114×10^{6} N.m^{-1}*

Table 2 describes the four local models multimodel base used for the simulations.

From this base, four optimal gains are calculated.

To calculate the linearization coefficients a_i and b_i, the following c_p empiric expression relative to 2MW wind turbines is used:

$$c_p = 0.18\times\left(\frac{90}{0.4+0.5\lambda} - 6.8 - 0.115\beta^2\right)\times e^{\frac{-8}{0.4+0.5\lambda}+0.16} \tag{36}$$

Table 2. Multimodel base parameters

Local model Mi	*Wind speed vi (m/s)*	*Pitch angle* $\beta_{i\text{-}nom}$ (°)
M1	*11.6*	*1.1*
M2	*14*	*8*
M3	*17*	*11.1*
M4	*25*	*15.4*

And thus, the stability feasibility problem consists in solving 8 LMI as shown after:

$$\begin{cases} P > 0 \quad (1\ LMI) \\ G_{ij}^{T}.P + P.G_{ii} < 0, i = 1..4 \quad (4\ LMI) \\ Q_{ij}^{T}.P + P.Q_{ij} < 0, i = 1..3,\ j = i+1 \quad (3\ LMI) \end{cases} \tag{37}$$

The simulation leads to the following result:

$$P = \begin{pmatrix} 8.848 & -8.335 & -0.008 & 0.298 \\ -8.335 & 8.101 & 0.007 & -0.258 \\ -0.53 & 0.523 & 0.026 & -0.001 \\ 0.298 & -0.258 & -0.001 & 0.084 \end{pmatrix} \tag{38}$$

Finding this positive definite matrix P is a sufficient condition proving the global stability of the control technique presented above.

For the simulations, we had chosen to place the closed loop poles for the local models at the left of -α=-0.5. This gives the following poles for each local model:

- *1st local model:* $P_1 = \begin{bmatrix} -1.032 + 12.24i \\ -1.032 - 12.24i \\ -1 \\ -1.069 \end{bmatrix}$
- *2nd local model:* $P_2 = \begin{bmatrix} -1.032 + 12.24i \\ -1.032 - 12.24i \\ -1 \\ -1.06 \end{bmatrix}$
- *3rd local model:* $P_3 = \begin{bmatrix} -1.033 + 12.24i \\ -1.033 - 12.24i \\ -1 \\ -1.049 \end{bmatrix}$
- *4th local model:* $P_4 = \begin{bmatrix} -1.037 + 12.24i \\ -1.037 - 12.24i \\ -1 \\ -1.046 \end{bmatrix}$

Thus, we can see that the pitch system pole (-1) is invariant for the four local models, and that all the other poles have their real parts less than –α.

To test the performance of this control strategy, a series of simulation for several wind steps has been performed to show the improvement of the studied controller against a classic multimodel LQ controller [8].

The local models poles for the classic LQ strategy have the following values:

- *1st local model:* $P_1 = \begin{bmatrix} -0.346 + 12.25i \\ -0.346 - 12.25i \\ -1.162 + 0.868i \\ -1.162 - 0.868i \end{bmatrix}$
- *2nd local model:* $P_2 = \begin{bmatrix} -1.2 + 12.401i \\ -1.2 - 12.401i \\ -2.693 + 2.013i \\ -2.693 - 2.0128i \end{bmatrix}$
- *3rd local model:* $P_3 = \begin{bmatrix} -1.62 + 12.587i \\ -1.62 - 12.587i \\ -3.417 + 2.287i \\ -3.417 - 2.287i \end{bmatrix}$
- *4th local model:* $P_4 = \begin{bmatrix} -2.148 + 12.845i \\ -2.148 - 12.845i \\ -4.108 + 2.418i \\ -4.1081 - 2.418i \end{bmatrix}$

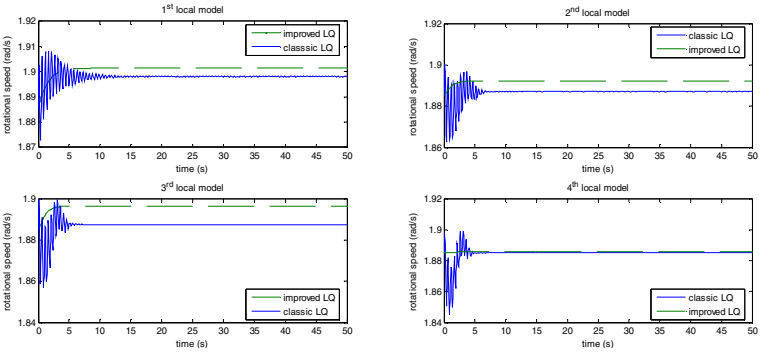


Fig. 5. Comparison simulation between classic and improved LQ control laws

The Figure 5 presents a comparison simulation between the two control laws. For this simulation, only the turbine rotational speed response for a wind step of 0.5 m/s is presented for the four local models.

In the comparison between both strategies, the system response for the proposed controller has indeed kept almost the same response time (about 4s), unlike the case with the classic LQ where the response times are varying from 12s for the 1st local model to 2s for the 4th local model.

Compared to the proposed strategy, the classic multimodel LQ control law shows responses with a more oscillating transient mode.

The improvements of the multimodel LQ controller consist in a more damped oscillatory mode and a faster dynamic than the classical control mode with an almost fixed response time for all the local models.

The simulation of the system with the proposed control strategy for a variable wind speed between 12m/s and 25m/s leads to the results presented in the curves of Figure 6.

The Figure 6 (a) illustrates a realistic aspect of the wind speed as described in a method elaborated by *C. Nichita* in [15]. From this aspect, the controller allowed a good regulation of the generated electrical power (Figure 6 (d)) and the rotation speeds of both the rotor (Figure 6 (b)) and the generator (Figure 6 (c)) around their rated values with taking into account the fatigue damage since the mechanical torque (Figure 6 (e)) maintains an almost constant value which thereby leads to have alleviated mechanical loads. As shown in Figure 6 (e), the mechanical torque is nearly equal to the estimated value.

The variations of the control signals: the electromagnetic torque presented in Figure 6 (g) and the control pitch angle in Figure 6 (h) are smooth and have acceptable variations.

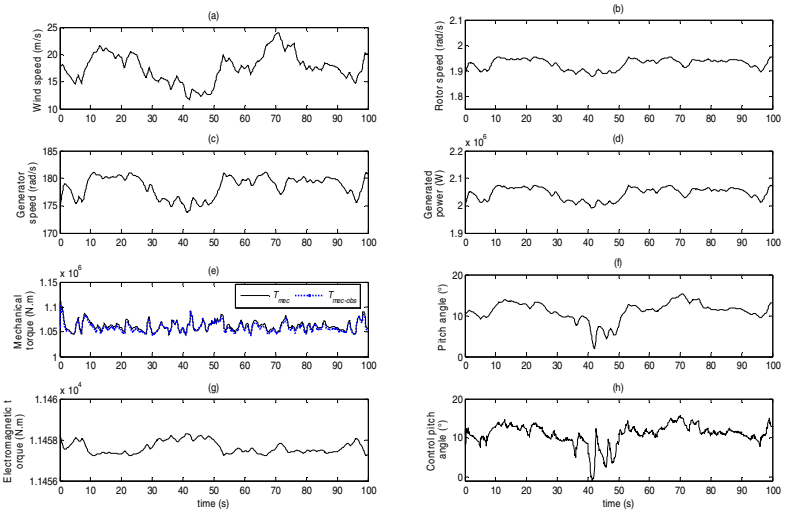


Fig. 6. Variation of the system variables

6 Conclusions

This paper dealt with a technique of designing a multimodel LQ regulator allowing to partially control the process global dynamic, and with a study of the global asymptotic stability of the controller by means of a set of LMI. The proposed strategy presented a compromise between different control objectives: optimizing the performances of the different system variables especially generating an electrical power of a good quality, minimizing the control efforts, alleviating the drive train dynamic loads and controlling the global dynamic of the studied process. The simulations results showed good performances of the controller with acceptable mechanical stress. But, satisfying such a trade-off between all these objectives is indeed difficult and the cost is however some high forces on the pitch actuator. These effects brought more challenges in the system analysis to improve the obtained results in order to control actively the system dynamic and to totally damp the oscillatory mode.

References

1. Bianchi, F.D., Mantz, R.J., Christiansen, C.F.: Control of Variable-speed Wind Turbines by LPV Gain Scheduling. Wind Energy 7(1), 1–8 (2004)
2. Camblong, H., Rodriguez, M., Puiggali, J.R., Abad, A.: Comparison of different control strategies to study power quality in a variable-speed wind turbine. In: 1^{st} world wind energy conference proceeding, Berlin (2002)
3. Boukhezzara, B., Lupua, L., Siguerdidjanea, H., Hand, M.: Multivariable control strategy for variable speed, variable pitch wind turbines. In: Science Direct, Renewable Energy, vol. 32, pp. 1273–1287 (2007)
4. Kardous, Z., Benhadj Braiek, N., Al Kamel, A.: On the multimodel stabilization control of uncertain systems – Part 1. International Scientific IFNA-ANS Journal: Problems on Nonlinear Analysis in Engineering Systems 2 (2006)
5. Kardous, Z., Benhadj Braiek, N., Al Kamel, A.: On the multimodel stabilization control of uncertain systems – Part 2. Int. J. of Problems of Nonlinear Analysis in Engineering Systems 27(1), 76–87 (2007)
6. Bianchi, F.D., De Battista, H., Mantz, R.J.: Wind turbine control systems: principles, modeling and gain scheduling design, 1st edn. Springer, London (2007)
7. Munteanu, I., Bratcu, A.I., Cultululis, N.A., Ceanga, E.: A two loop optimal control of flexible drive train variable speed wind power systems. In: 16^{th} Triennial World Congress-IFAC, Prague (2005)
8. Khezami, N., Guillaud, X., Benhadj Braiek, N.: Multimodel LQ controller design for variable-speed and variable pitch wind turbines at high wind speeds. In: IEEE International Multi-conference on Systems, Signals and Devices, Djerba (2009)
9. Poulsen, N.K., Larsen, T.J., Hansen, M.H.: Comparison between a PI and LQ-regulation for a 2 MW wind turbine. In: Risø National Laboratory-I-2320 (2005)
10. Hammerum, K., Brath1, P., Poulsen, N.K.: A fatigue approach to wind turbine control. Journal of Physics 75 (2007)
11. Cutululis, N.A., Bindner, H., Munteanu, I., Bratcu, A., Ceanga, E., Soerensen, P.: LQ Optimal Control of Wind Turbines in Hybrid Power Systems. In: European Wind Energy Conference and Exhibition, Athens (2006)

12. Chedli, M.: Stabilité et commande de systems décrits par des multimodèles. PhD thesis, Institut National Polytechnique de Lorraine (2002)
13. Liberzon, D., Morse, A.S.: Basic problems in stability and design of switched systems. In: IEEE Control Systems Magazine, vol. 19, pp. 59–70 (1999)
14. Kardous, Z., Elkamel, A., Benhadj Braiek, N., Borne, P.: On the quadratic stabilization in discrete multimodel control. In: IEEE Conference on Control Applications, vol. 2, pp. 1398–1403 (2003)
15. Nichita, C., Luca, D., Dakyo, B.: Méthodes de simulation de la vitesse du vent. In: Decentralized energy seminar, Toulouse (2003)

Conformance Testing of Soft-Core Can Controllers: A Low-Cost and Practical Approach

Imran Sheikh[1] and Michael Short[2]

[1] Embedded Systems Laboratory, University of Leicester
University Road, LE1 7RH, UK
si52@le.ac.uk
[2] Electroincs & Control Group, Teesside University
Borough Road, Middlesbrough, UK
m.short@tees.ac.uk

Abstract. Since its introduction in the early 1980's, CAN has become the de-facto communications protocol employed in vehicle and industrial control applications. Before any new device can claim to support CAN-connectivity, extensive conformance testing is normally required to demonstrate compliance with the protocol at the physical and data link layers. To help standardize the nature of the specific test plans and documentation required for this testing, the ISO has published a set of draft standards specifically for CAN conformance testing. To date, most commercial CAN controllers and transceivers have been implemented at the silicon level, either in the form of dedicated IC's or as on-chip peripherals of embedded devices. The practical implementation of CAN conformance testers has been realised using dedicated hardware and specially written analysis software; this is a practical approach when testing and verifying conformance prior to high-volume IC manufacture. However, recent years have seen an increased interest in the employment of programmable logic devices such as FPGA's for the implementation of CAN controllers and CAN-enabled devices. Such 'soft core' implementations are often realised in small-volume (or sometimes even one-off) batches; in such circumstances, cost and availability reasons may dictate that developers simply cannot employ traditional CAN-conformance testing equipment. To help alleviate this problem, this paper proposes a low-cost and easily implemented method which will allow developers to fully test a CAN soft core implementation. The method allows developers to verify a CAN core against the relevant ISO standards using only (low-cost) off-the-shelf development boards, coupled with a simple analysis tool such as Chipscope. Finally, the paper extensively describes the use of the test bed in the verification of an open-source CAN soft core implementation.

Keywords: Controller area network, Conformance testing, Soft core, Network protocol verification.

1 Introduction

Conformance testing is an integral part of the development stage of any network protocol implementation. When performed correctly conformance testing verifies, to an

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 129–141.
springerlink.com

acceptable degree of confidence, that the implementation of given set of protocol specifications has been correctly interpreted by the designers and has been instantiated in a form that is free from errors.

Since its introduction in the early 1980's, the Controller Area Network (CAN) protocol has become the de-facto communications protocol employed in vehicle and industrial control applications. In light of the popularity of CAN, the ISO has developed a standard exclusively aimed at CAN conformance testing. Before any new equipment design can claim to be CAN conformant [1] , evidence is required that shows that the testing procedures outlined in [2] standard have been performed and passed without problem. The ISO document not only specifies different types of tests that must be performed for conformance testing, but also specifies a Test Plan (TP) architecture based on the [3]. The required TP is shown in figure.1. As can be seen from this figure, the TP architecture indicates that the tester should be divided into two parts. The first component is the Lower Tester (LT) which provides the test pattern generation and analysis. The second is termed the Upper Tester (UT), which is required to contain the software to control the CAN Implementation Under Test (IUT). The UT is normally a host processor or programmable device of some kind, and also provides coordination to conduct the tests between the LT and the IUT [4]. The UT receives stimulus (with details of the test being performed) from the LT, and generates messages passed on to the IUT. The IUT then processes these messages, and both the UT and LT components monitor its behaviour for consistency with the CAN protocol. If the result is satisfactory, the test is considered passed and testing proceeds to the next conformance test. It should be noted that the testing procedures that are required to be implemented include coverage of common error conditions, randomized tests and also bit timing tests. Most tests are critical, and the latter category bit timing contains a number of tests that can be difficult to localize, and a suitable means is required to capture and display multiple logic signals over an appropriate time scale. This typically requires the use of dedicated hardware and Logic analysers [5].

Traditionally, CAN controllers and transceivers have been implemented at the silicon level, either by dedicated IC's or as on-chip peripherals of embedded devices. Practically, the implementation of CAN conformance testers has been done using dedicated hardware and specially written analysis software, which is a practical approach when testing and verifying conformance prior to high-volume IC manufacture. The motivation for the current work is as follows. Recent years have seen an increased interest in the employment of CAN-enabled devices implemented by programmable hardware devices such as FPGA's. By their very nature, such 'soft core' implementations are often needed in small-volume (or even one-off) batches. In these circumstances, cost and availability reasons often dictate that it is not practical for developers to use traditional CAN-conformance testing equipment. An ongoing project within the Embedded Systems Laboratory (ESL) at the University of Leicester requires the use of such a CAN soft core [6]. To help alleviate this problem, this paper proposes a low-cost and easily implemented method which will allow developers to test a CAN soft core implementation for conformance to the relevant standard without the need for expensive or proprietary hardware interfaces and logic analysers.

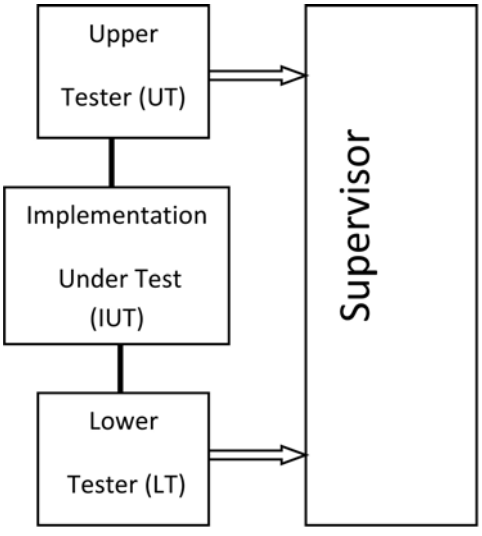


Fig. 1. ISO 9646-1 Test Plan Architecture

The remainder of the paper is organized as follows: In the next Section, we will describe the Controller Area Network in brief and then will present a review of different CAN conformance testing implementations of the before and after ISO standard evolved. Section 3 describes the formation of the current test bed for CAN conformance testing. Section 4 presents the case studies involving two of the tests being carried out by the proposed approach to CAN conformance. Section 5 will present the analytical comparison of our approach to the other techniques used for conformance testing. Section 6 presents our initial conclusions.

2 Controller Area Network

2.1 Protocol

The Controller Area Network (or simply CAN) is one of the most widely employed protocols for creating distributed embedded systems, with applications as far ranging as vehicle electronics, process control and many other important industrial applications [1]. Some of the key features of CAN that have led to its widespread use include low overheads, non-destructive bitwise message arbitration, low message latency and good error detecting abilities; all features which are required for control applications running on embedded processors [7,8]. As mentioned, the protocol employs a unique non-destructive priority-based arbitration scheme; when multiple nodes attempt to transmit messages simultaneously this mechanism ensures that the highest priority message gains first access to the bus. If priorities are carefully assigned to the messages, and appropriate timing analysis is performed, CAN may be used to implement several different types of time-critical systems (e.g. see [7,8,9]).

The wired-AND nature of the physical layer, which is used to achieve the aforementioned priority-driven arbitration, requires that all nodes in the network achieve a logical consensus on the instantaneous bit-patterns appearing on the bus lines. This particular requirement of the protocol acts to severely limit both the maximum transmission speed and bus length of a given CAN network; the maximum transmission rate is inversely proportional to the length of the bus, and has an upper limit of 1 MBit/s; due to its design a CAN frame may carry up to a maximum of 8 data bytes. In addition, this mechanism places extremely specific requirements on the nature of the conformance testing that must place for a CAN enabled device; a significant proportion of the ISO test standard is devoted to this single, critical aspect of the protocol.

2.2 CAN Conformance Testing

One of the earliest CAN prototype controllers was named DBCAN [10]. This implementation was tested using a logic analyser and a pattern generator circuit. As there was no standard for conformance testing at the time the prototype was developed, a commercial basic (as opposed to full) CAN controller was used as benchmark for verification. A major disadvantage of this scheme was the use of external interface modules to visualize the state of different DBCAN registers, and the testing procedure was somewhat limited in the number of signal channels that could be simultaneously analysed. Since this is a needed requirement in the case of ISO standard conformance testing the ability to visualize the state of large numbers of CAN registers simultaneously is a prerequisite such a setup is limited in this respect.

A Hardware emulation technique was used to verify a CAN soft core in [11]; firstly, the synthesized net list is downloaded into a hardware emulator. This emulator is configured by a PC and the communication between the two is carried via a specially designed interface card connected to the EISA bus; this emulator is also connected to 2 commercially available CAN chips. The drawback with this technique is that again, customized hardware along with software especially written to carry out the conformance testing is required. Additionally, to emulate the bus failures and potential error conditions on the bus a manual technique of connecting CAN bus to the output of individual nodes is employed, which lacks efficiency and is not robust enough to cover all the scenarios given within ISO DIS 16845.

A slightly different verification technique was reported by [12]. Their technique employed custom design boards with 8051 microcontrollers and SJA1000 CAN controllers, but this method involved the design of specialized interface hardware and boards to assist with the testing plan. Specialised verification architecture for testing automotive protocols (including CAN) at both the module and chip level was proposed by [13]. Again, this work requires a specially designed CAN verification component as part of the silicon, while the selection and implementation of actual test sequences, along with the selection of a suitable means to monitoring bus signals, is left open for the tester.

With respect to soft core CAN implementations, the CAN e-Verification (CANeVC) test bench has previously been described [14]. This commercial test facility requires a CAN specification core to be embedded in the netlist; this core then runs specific tests to verify the behaviour of the CAN soft core. Again, this technique involves a

time consuming development of a test bench using an expensive commercially available verification IP ; additionally, compatibility issues often arise when using CAN implementations other than the proprietary implementation and only a limited number of programmable logic devices are supported.

Finally, several experimental implementations (such as that reported by [15] to measure single parameters - such as CAN bit errors - rather than perform complete conformance testing have been described in the literature. Such implementations have typically used complex and non-trivial means, requiring customized hardware and software. In summary then, it can be observed that - to date specialised hardware and / or software has been required to assist with CAN testing plans. In the following Section, a novel testing approach that relies only upon the use of low-cost, standard off-the-shelf hardware and software is described.

2.3 Protocol Limitation

However, the CAN protocol itself is not without its drawbacks; although the basic raw protocol is suitable for use in many soft real-time systems, it suffers from several significant problems with respect to hard real-time systems, for example in safety-critical distributed brake-by-wire systems. These drawbacks include redundancy issues, atomic broadcast problems, lack of protection from babbling idiot failures and information throughput restrictions. Although research has shown that many of these issues can be dealt with by the creation of higher-level, software-based protocol extensions (e.g. see [7,8,16]), some proposed solutions are either complex to implement placing a significant computational overhead on the host CPU or simply cannot be implemented in software and a hardware solution must be adopted [17]. Although the proposed changes are conceptually quite straightforward, implementing them directly in silicon is costly and has proved to be problematic for small-volume requirements. A workable solution to this problem is to implement the protocol on a PLD; the required modifications may then be achieved with relative ease. This solution brings with it another, related problem; before verification of any modifications can take place, it must be shown that the basic soft core CAN implementation is fully conformant to the protocol, a potentially costly and time-consuming procedure in its own right.

To help alleviate this problem, the current authors have previously proposed a low-cost and easily implemented method in a technical report [18]. This report essentially describe the application of the techniques described in the previous Section to the conformance of a CAN soft core. The following Section contains a description of the test facility that was employed in these studies.

3 Test Bed

Real-time testing of a CAN implementation is quite a complicated procedure, and in this case for practical reasons, no specialized hardware and software was available to generate the required testing patterns and monitor the behaviour of the CAN soft core. For this reason, it was decided to use only low-cost off the shelf components.

In addition to these standard hardware parts, the Chipscope analysis tool [19] was used to visualize and capture the behaviour of the soft core, allowing verification of the

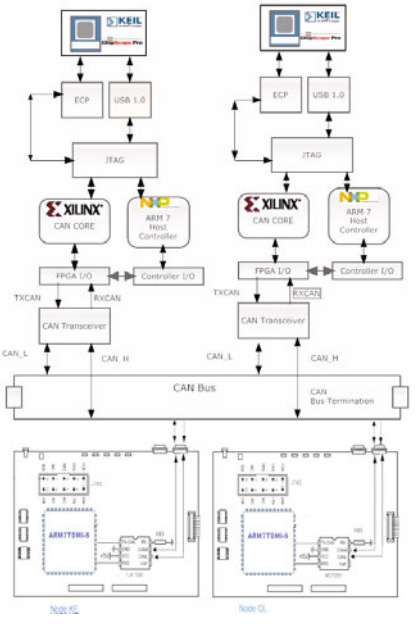


Fig. 2. Conformance Test Bed

testing results. Chipscope is a Xilinx testing tool which is implemented by inserting a small core onto the device to be monitored, allowing multiple signal channels to be captured via a JTAG interface. Up to 16 internal signal ports can be analysed in a single core, and each port can have up to 256 signals. Multiple cores can be attached in a FPGA to increase the number of signals [20]. In comparison to other means for capturing multiple FPGA signals, Chipscope retains the key features required but is a fraction of the cost. Additionally, to support one-off conformance testing plans without causing excessive costs, a fully-featured evaluation version is available for a 60 day period a full testing plan can be performed in such a time frame. Hence these features of Chipscope made it an obvious choice for our CAN conformance test bed. The new test facility is shown schematically in figure 2. In the next paragraphs we give a full list of the hardware and software components and tools used in building the Test Bed.

3.1 Hardware

1. Two Integrated boards with FPGA's (programmed with CAN soft core) and an ARM 7 working as a Host controller. These boards are named as SC1 and SC2. The purpose of using two soft cores is to verify simultaneous behaviour as a CAN Transmitter/Receiver as well as to generate special patterns on the CAN bus using VIO and additional soft modules embedded with the soft core.
2. Two ARM 7 Micro controller boards with Integrated CAN controller and transceivers (Node KE and OL in figure 2). These boards are used as CAN Receiver Nodes for further verification of the messages sent by the CAN soft core. The other

utility of these CAN nodes are to induce extra or erroneous bits which is a requirement of few test cases as mentioned in [2].

The figure 2 demonstrate a 2 node Soft Core, it can easily be extended to n number of nodes considering the protocol limitations.

3.2 Software

1. Xilinx ISE [21] for soft-core programming, synthesis, routing and programming the FPGA. The ISE is a complete IDE for FPGA development and contains some extra features like power analysis, optimal routing and timing analysis to name a few.
2. Chipscope Pro is used as analysis tool. The VIO core is used to generate and control different bit patterns. These bit patterns can be synchronous or asynchronous.
3. The Keil uVision 3 IDE [22] with free ARM tools C compiler was chosen for programming and debugging the Microcontroller boards.

As can be seen, the test bed has been made using COTS hardware and also taking in care the structure of the Test Plan given in the ISO 9646-1. The Test Bed consists of two instances of the IUT, the main purpose of using the second IUT is to generate errors on the CAN bus and special conditions which were either the pre-requisite for a test case or generating special bit stream during a test for verifying the behavior of the IUT, hence the second instance of IUT is moreover working as the LT in reference to the ISO9646-1 TP.

The ARM boards with integrated CAN Controllers were used either as receivers/ transmitters to verify the conformance of the IUT with widely used CAN controllers, and were also employed to generate bit errors on the CAN bus using an interrupt generation mechanism. This scheme is highly synchronized as the bit inversions were done at the specific point where it was required; the methodology employed for test pattern generation is described in the next Section.

3.3 Test Pattern Generation

When using pattern generators test vectors are required to be first stored, and are sent on the CAN bus only when required thus putting the IUT in different states and allowing its behaviour and responses to be analysed. In our proposed test bed we have used FPGA based pattern generation, which is not only economical as no extra price was added to the test setup but also it is added as a separate Verilog [23] module to the main CAN Core and it will work non-intrusively as this cod is part of the CAN core which is taking the role of an LT. This helped us to accurately produce special conditions; for example in a test case it was needed to produce extra dominant bits on the CAN bus after an IUT working as a transmitter send an Error Frame [2]. This test pattern was easily achieved by modifying the Verilog module for Error Flag generation to produce extra dominant bits, as illustrated by the code fragment:

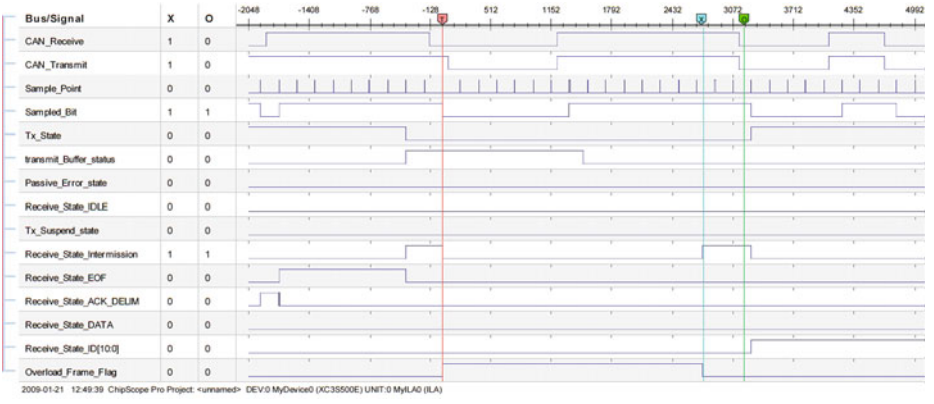


Fig. 3. Transmitter snapshot for Overload frame test case

```
reg  [3:0] Error_Flag_Counter;//changed from reg [2:0]

always @ (posedge Clock or posedge Reset)
 begin
  if (rst)
   Error_Flag_Counter <= 4'd0;
  else if (Error_Frame_End | Error_frame_Start)
   Error_Flag_Counter <=#delay 4'd0; //changed from 3'd7
 elseif (Error_Frame&Transmit_Instance&Error_Flag_Counter<4'd11))
   Error_Flag_Counter <=#delay Error_Flag_Counter + 1'b1;
 end

always @ (Error_frame or Error_Flag_Counter )
 begin
  if (Error_frame)
   begin
    if (Error_Flag_Counter < 4'd11)//changed from 3'd7
     begin
      if (Node_Error_Passive)
       Tx_CAN = 1'b1;
      else
       Tx_CAN = 1'b0;
     end
   end
 end
```

4 Test Cases

The proposed test facility was employed to test the CAN conformance of the custom created CAN soft core, written in Verilog. As the number of total number of test cases to consider in any single CAN conformance test plan is numerous, it is beyond the scope of

the current paper to present comprehensive test results; such test results are available in the form of technical report [18]. However, in this Section we will present two test cases that help highlight the main features of the proposed facility. Both tests were carried out successfully, and are described in the following two Sections.

4.1 Overload Frame Management

This test is a part of the Overload Frame Management class [2]. This test verifies that an IUT will be able to transmit a data frame starting with the identifier and without transmitting SOF, when detecting a dominant bit on the third bit of the intermission field. This test involves two instances of the IUT and the ARM7 Micro controller boards. The test will be setup using the following organization:

1. Both of the IUT's must be in default state ready for transmission or reception according to the setup sent by the Host Controller.
2. The IUT acting as the Transmitter is set to transmit two data frames as programmed in the Host processor.
3. The Receiver IUT will be set to request an Overload frame after reception of the first frame.
4. After the completion of the Overload Frame on the third bit of the Intermission field (Normally the Intermission field is a sequence of three Recessive bits) is set to dominant by the Fault injector node i.e. K.E.
5. The transmitter must not consider it as a bit error and shouldn't send a Dominant level SOF and consider the dominant bit of the Intermission field as the SOF.
6. Normal reception of the message should take place.

This test was successful with desired results as stated in the purpose of the test; the observation on the transmitter node from the Chipscope - shown in figure 3 is as follows:

1. Left of Marker 'T' The Tx_state flag is high indicating ongoing transmission, Receive_state_Data and ACK_DELIM indicating a successful transmission while the node is error active.
2. At Marker 'T' there is an error on the Receive_state_intermission field generating an overload frame with Overload Flag of six dominant and Overload delimiter of 8 bits as can be seen by the count of sample point.
3. After the Overload frame an intermission field signal can be seen at the Marker 'X'.
4. The third bit of intermission field is is a dominant bit as can be counted between Markers 'X' and 'O' the number of sample points is 2 and the third sample point is a dominant bit.
5. Just after the Marker 'O' we can see the Receive_State_ID [10:0] going high without any SOF. The Identifier first 4 bits are dominant as required by the Test case.

4.2 Suspended Transmission Field Test

The purpose of this test is to verify that a Passive state IUT acting as a transmitter doesn't transmit any frame before the end of a suspended Transmission field following an Error Frame.

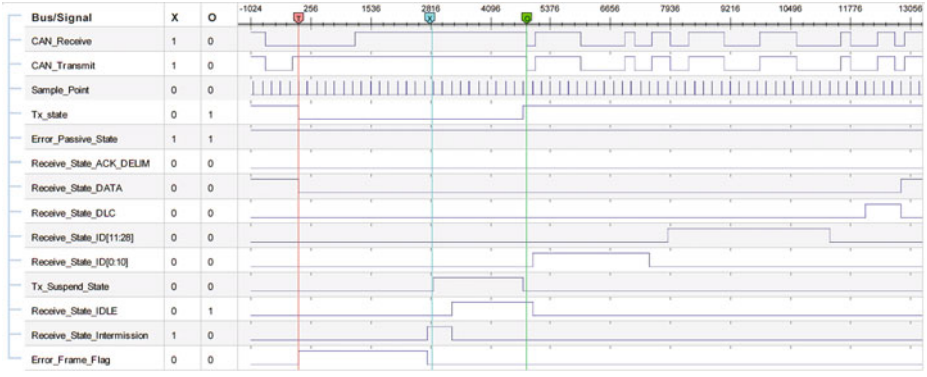


Fig. 4. Transmitter snapshot for suspended transmission test case

The setup for this test is to put the Transmitter IUT into Error Passive state and then introduce a bit error during transmission. The IUT will send a Passive Error Flag which is overwritten by Active Error Flag from participating CAN nodes, the IUT will receive 7 dominant bits as the Error delimiter. After the completion of the Error Flag, the IUT will send the pending data frame once the Suspended Transmission Field is over.

This test was successful with desired results as stated in the purpose of the test; the observation on the transmitter node from the Chipscope - shown in figure 4 is as follows:

1. Left of Marker 'T' The Tx_state Flag is high indicating on going transmission when a bit error set the node to send an Error_Frame, the node is Error Passive.
2. Between Marker 'X' and 'O' there is no transmission when the Tx_Suspend_state is high which is indicated by the TX_State low status although in case of error recovery the re-transmission of the message is carried soon after the intermission field.
3. As the Tx_Suspend_state is set to low after 8 bit times then the SOF is indicated by CAN_Receive and CAN_Transmit signals which changes to dominant state.
4. Rest of the Receive state signals indicates the re- transmission.

5 Comparative Study

This Section presents a cost and flexibility comparison between conventional CAN conformance testing hardware and software with the approach that has been discussed in this paper. The first observation is that the next facility does not require expensive CAN PC interface cards [24,25] which are normally required for CAN conformance testing [26]. These cards are used to capture CAN bus data to analyse the internal status of different registers and to log the events; these cards not only required the hardware but also specialized software [27] along with interface cables which can also add to the cost and complexity of the setup. In our proposed implementation we can analyse the internal state of CAN IUT directly using Chipscope, and also by using the Keil uVision 3 IDE. This IDE can run an online debug on the COTS ARM boards, which are significantly cheaper in comparison to specialised CAN interface Cards. The Keil IDE can display the status of all the CAN

registers hence providing us useful information in verification of the status of the CAN bus and the data being transmitted and received. In addition, there are several key advantages of our proposed test bed using Chipscope over hardware logic analyser systems:

1. The standard bench analysers doesn't show enough signals as required in case of CAN conformance as illustrated in section 4. There are Logic analyser systems which can show large number of signals simultaneously with large data widths [28] but there prices are 10 times more than Integrated Logic analyser.
2. Normal Bench analysers can show Mega samples [29,28], while the Chipscope is limited to a Sample width of 16K, we overcome this problem by using Digital clock Manager [30] which can divide or multiply the system clock by 'n' times, the board we used in our system can divide the system clock by 16 times hence we were able to capture 16 times more sample than on system clock which can easily capture 3 to 4 complete CAN messages in a single trigger.
3. Additional probes with wide numbers of I/O pins are required to interface with the Logic analysers while Chipscope can carry magnitude of these signals using a simple JTAG cable, although there are few solution like Agilent's FPGA trace port [31] which use a simple interface to analyse multiple signals but it also requires a specialized hardware and Chipscope pro tool.
4. Not only all I/O signals are accessible through Chipscope but also internal wires can be traced [32] which are really helpful in Conformance testing specially when setting up triggering conditions we have lot more options to setup a trigger condition for example in the test cases discussed it is really easy to setup a trigger condition to wait for an Error Frame flag signal goes to high to analyse an error condition, while for external Logic analysers only I/O signals are available.

6 Conclusions and Future Work

In this paper we have presented an approach to utilize Virtual I/Os and Integrated Logic Analysers to perform CAN conformance testing in accordance with the ISO standards. It has been shown that the facility is capable of performing the full range of test required and specially related to CAN bit timing tests conforming to the relevant CAN standard. In conclusion, this facility can be assembled and used for a fraction of the cost of a regular test facility for CAN conformance. A full list of the how each individual test may be implemented when using a facility such as this has been described in [18].

As a final note, it can be seen that test facility that has been described is not restricted to the CAN protocol, and with suitable modifications can be used to test conformance of many alternate network protocols, for example [33].

Acknowledgements. The work in this paper is supported by the award of a PhD scholarship to Sheikh Imran from the NWFP University of Engineering & Technology, Peshawar, Pakistan through the Higher Education Commission of Pakistan.

References

1. 11898:1993(E), I.: Road vehicles interchange of digital information controller area network (can) for high speed communication (November 1993)
2. ISO16845. Road vehicles- controller area network (can) - conformance test plan
3. ISO9646-1. Information technology- ISO - conformance testing methodology and frame work-part 1: General concepts
4. Carmes, E., Junier, C., Aussedat, F.: Can conformance: Methodology and tools, keynote speech. In: CAN in Automation Proceedings of 3rd iCC, Paris (October 1996)
5. Lawrenz, W., Kinowski, P., Kircher, G.: Can conformance testing-the developing iso standard and necessary extensions. In: Proceedings of International Truck and Bus Meeting and Exposition, Indianapolis, Indiana (November 1998)
6. Sheikh, I., Pont, M.J., Short, M.: Hardware implementation of a shared-clock scheduling protocol for can: A pilot study. In: 4th UK Embedded Forum, University of Southampton (September 2008)
7. Short, M., Pont, M.: Fault-tolerant time-triggered communication using can. IEEE Transactions on Industrial Informatics 3(2), 131–142 (2007)
8. Broster, I., Burns, A.: Timely use of the can protocol in critical hard real-time systems with faults. In: 13th Euromicro Conference on Real-Time Systems, pp. 95–102 (2001)
9. Pedreiras, P., Almeida, P.: Edf message scheduling on controller area network. Computing and Control Engineering 13(4), 163–170 (2002)
10. Kirschbaum, A., Renner, F.M., Wilmes, A., Glesner, M.: Rapid-prototyping of a can-bus controller: a case study. In: proceedings of Seventh IEEE International Workshop on Rapid System Prototyping, June 1996, pp. 146–151 (1996)
11. Winter, A., Bittruf, D., Tanurhan, Y., Muller-Glaser, K.D.: Rapid prototyping of a communication controller for the can bus. In: Proceedings of Seventh IEEE International Workshop on Rapid System Prototyping, pp. 152–157 (1996)
12. Nimsub, K., Dawi, K., Kyuhyung, C., Jinsang, K., Wonkyung, C.: Design and verification of a can controller for custom asic. In: CAN in Automation Proceedings of 10th iCC (2005)
13. Zarri, G., Colucci, F., Dupuis, F., Mariani, R., Pasquariello, M., Risaliti, G., Tibaldi, C.: On the verification of automotive protocols. In: Proceedings of Design, Automation and Test in Europe, March 2006, vol. 2, pp. 6–10 (2006)
14. Di Blasi, A., Colucci, F., Mariani, R.: Y-can platform: A re-usable platform for design, verification and validation of can-based systems on a chip. In: ETS 2003 Symposium (2003)
15. Ferreira, J., Oliveira, A., Fonseca, P., Fonseca, J.A.: An experiment to assess bit error rate in can. In: 3rd international workshop on real-time networks RTN 2004, Proceedings (June 2004)
16. Ayavoo, D., Pont, M.J., Short, M., Parker, S.: Two novel shared-clock scheduling algorithms for use with 'controller area network' and related protocols. Microprocess and Microsyst. 31(5), 326–334 (2007)
17. Sheikh, I., Short, M.: Improving information throughput in controller area networks: Implementing the dual-speed approach. In: Proceedings 8th International Workshop on Real-Time Networks, Dublin, Ireland (June 2009)
18. Sheikh, I., Short, M.: Can conformance testing-a new approach. In: Technical Report techreport ESL-09-01, ESL, Engineering Department, University of Leicester (February 2009)
19. Xilinx. Chipscope Pro Software and Cores (January 2007)
20. Oltu, O., Milea, P., Simion, A.: Testing of digital circuitry using xilinx chipscope logic analyzer. In: Proceedings International Semiconductor Conference, CAS 2005, vol. 2, pp. 471–474 (2005)
21. Xilinx. Ise foundation, http://www.xilinx.com/ise
22. Keil. uvision ide tool (2008), http://www.keil.com/uvision

23. IEEE. IEEE standard for verilog hardware description language. IEEE Std 1364-2005 (Revision of IEEE Std 1364-2001), 1–560 (2006)
24. National Instruments. 1 Port, High Speed CAN, USB Interface (2008)
25. Softing, A.G.: CAN/CANopen/DeviceNet Interface boards (2007)
26. Lawrenz, W., Kinowski, P., Kircher, G.: Can conformance testing - state of the art and test experience. In: Proceedings of 5th International CAN Conference, ICC 1998, San Jose, California (1998)
27. NI. Lab view 8.6 (2009)
28. Agilent. Series logic analysis system 16900 mainframes (2008)
29. Tektronics. Tla 5000b logic analyzers (2009)
30. Xilinx Inc. Using Digital Clock Managers (DCMs) in Spartan-3 FPGAs, Application Note (2003)
31. Agilent. Deep storage with xilinx chipscope pro and agilent technologies fpga trace port analyzer
32. Lee, T., Fan, Y., Yen, S., Tsai, C., Hsiao, R.: An integrated functional verification tool for fpga systems. In: Second International Conference on Innovative Computing, Information and Control, ICICIC 2007, p. 203 (September 2007)
33. Fuhrer, T., Muller, B., Dieterle, W., Hartwich, F.: Time-triggered communication on can (time-triggered can ttcan). In: Proceedings of iCC 2000, Amsterdam, The Netherlands (2000)

Design Parametrization for Dynamically Similar Delayed Teleoperation Systems

Ernesto Olguín-Díaz[1], Vicente Parra-Vega[1], Luis G. García-Valdovinos[2], and Vicente de Paul Garca-Alvizu[1]

[1] Robotics and Advanced Manufacturing Division, CINVESTAV-Saltillo, México
[2] Centro de Ingeniería y Desarrollo Industrial, CIDESI, Querétaro, México
{ernesto.olguin,vicente.parra}@cinvestav.edu.mx,
ggarcia@cidesi.mx

Abstract. Bilateral teleoperation system are prone to instability coming out from the time-delay introduced by the indeterministic communication channel. This problem has been subject of intensive research under the assumption of non-equal master-slave teleoperators, however, dynamical similarity of master-slave robots can always be induced by proper tuning of feedback gains in the controllers, based on the physical ratio of both robots When we consider a *linear* Dynamically Similar Teleoperation System (DSTS), there arises advantages, observed heuristically in several experiments [1] or analytically in [10]. In this paper, a simple yet important parametrization trade-off of feedback control gains and performance teleoperation parameters is presented that allows an easy way to achieve stability or even also passivity. The analysis is based on an an impedance control scheme reported in [2] when the time-delay in the communication channel is considered constant but unknown. Stability and passivity limits are obtained based on Llewellyn's and Raisbeck's criteria.

This result explicitly suggests clearly guidelines among key factors, such as time delay, desired velocities and feedback gains in terms of the scaling parameters, arises a clear advantage when dealing with dynamically similar systems. This explains why the transparency of the teleoperation system, also studied under the same control law in [10] is improved by augmenting/reducing the dynamic/kinematic scaling factor, for given desired frequency, time delay and feedback gains.

Keywords: Teleoperation, Haptics, Parametric design.

1 Introduction

A bilateral teleoperation system is composed of a master robot and a slave robot, with a human operator commanding the master robot in order to produce the desired position and/or contact force trajectories for the remotely located slave robot, depending whether the slave tool is in contact or not. In turns, the slave robot follows these trajectories and *touches* the environment with contact forces that have to be sent back to the master robot to induce the touching sensation to the operator. In this way, a force/force-position control law is implemented in the master/slave stations with a communication channel introducing delayed position and forces signals [4]. It is well known that the

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 143–155.
springerlink.com

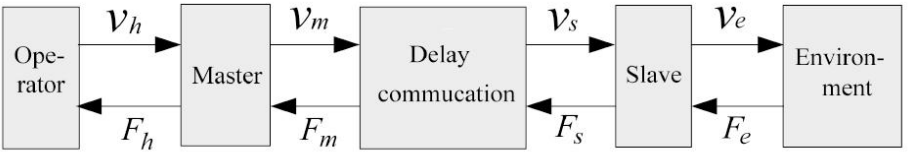


Fig. 1. Basic Bilateral Teleoperation Scheme

source of instability of bilateral teleoperation system appears because the time-delay introduced by the indeterministic communication channel is not passive [8]. Then, the limits to achieve human operator stable interaction with the slave robot, placed at a remote location, through the master robot, becomes an issue, in particular successful teleoperation requires a certain degree of transparency (Manipulation of the slave robot without any dynamics in between) and/or telepresence (Sensation of being physically in the remote environment). Limits of stable and passive interaction for DSTS subject to time-delay in the communication channel is studied in [10], and also advantages of dynamically similar

In this paper, motivated by the empirical observation that a DSTS are easier to handle and the previos results presented in [10], it is argued that dynamic and kinematic similarity introduce a clear trade-off of some of the aspects of telepresence and transparency, that not only depend on the simultaneous convergence of the position and force error between the master and slave robot but also on the subjective cognition of the human operator of being there, which in turns depends, among other aspects, on visual aids of the remote environment, kinesthetic coupling, the ability to deal with delayed signals and the man-machine interface.

Then it is reasonable to expect a simple trade-off of these aspects. Preliminary analytical results are presented in [10], based on an impedance control scheme [2], when the delay is considered unknown but constant.

1.1 Contribution and Organization

In this work the trade-off discussed in the previous section is presented in a parametrical way that allows better understanding of the stability/passive limits in the chosen set for the scaling factors and the imposed closed-loop impedance parameters.

Our basic hypothesis is that as long as the human perceives linear correlated variations in both teleoperated robots, he can improve the command of the closed-loop bilateral teleoperation system since spatial and temporal attributes of the visual remote location and kinesthetic coupling will vary linearly without distortion. So cognitively, the human can quickly learn to command the task with greater dexterity. Additionally in this paper we deal with unknown time delay so we design a novel controller to deal with unknown constant time delay [2], [1]. A computed-torque controller is employed in the master station and a computed torque second order sliding mode controller in the slave station is proposed to produce a desired impedance in closed loop. Then, stability theory is used to analyze the closed-loop stability and passivity properties and therefore the limits of human-teleoperation [10] scaling factors. The contribution of this work lays

on the fact that trade-off between these scaling gains can be achieved easily by looking at the stability/passivity limits when expressed in an appropriate parametric way.

Hence a review of DSTS with constant unknown delay is presented in Section 2 while controllers are explained in Section 3. Stability and passive analysis using Llewelyn [7], and Raisbeck [9] criteria, and their parametrization are presented in Section 4. The full relationship and trade-off of the system parameters that can be used for design purpose are given in Section 5. Final conclusions are given in Section 6.

2 Dynamically Similar Delayed Teleoperation System

A dynamically similar teleoperation system [3] has constant scaling factors which relate kinematic and dynamic parameters of the master and slave robots. This similarity between the systems is poorly understood so far, since there is not theoretical apparent evident advantage to work out with bilateral system, despite some analysis reported in [6], where the advantages has not been addressed properly in terms of explicit trade-offs of feedback gains, system parameters, desired trajectories and time delay. We surmise in this paper that DSTS improves significantly the ability of humans operating the master teleoperator to carry out efficiently teleoperation tasks at remote environments, when there is an unknown time delay involved in the communication channel.

Impedance control has been explored in [1] to enforce a desired impedance dynamics in closed loop in order to program arbitrarily the desired impedance parameters. This closed-loop linear dynamic allows to model the entire system as a 2-port network to relate the force and flows of input and output, respectively, by an impedance matrix or an hybrid matrix. This matrix can be used to describe the stability of the entire system using the Absolute Stability Theory [5], where a tight relationship between *output* scaling factors and impedance parameters can be found to give sufficient conditions on stability. However, when *dynamic* scaling factors are introduced, the master and slave robot dynamics are related by constants, either in kinematic and/or dynamic parameters, thus, a sort of advantages can emerge from this relation since a single feedback parameter appears. In this paper, we offer an analysis in terms of both the Llewelyn's criteria and Raisbeck's criteria, and verify its real time performance, which demonstrates a clear and intuitive trade off in terms of scaling factors of the DSTS.

2.1 Dynamically Similar System

Consider the dynamics of a linear teleoperation system consisting of two n-DoF manipulators decoupled systems as follows

$$M_m\ddot{x}_m + B_m\dot{x}_m + K_m x_m = F_{mc} + F_m \tag{1}$$

$$M_s\ddot{x}_s + B_s\dot{x}_s + K_s x_s = F_{sc} + F_s \tag{2}$$

where $\ddot{x}_i$ and $\dot{x}_i$ denote acceleration and velocity of the robot i,respectively; F_{mc}, F_{sc} are the control force inputs and F_m, F_s are external forces induced to the master and to the slave systems, respectively. The dynamic parameters M_i, B_i and K_i are the inertia, damping and stiffness, positive coefficients, of the systems; with $i = m$ for the master and $i = s$ for the slave.

Let $K_k > 0$ be the kinematic scaling factor of the slave configuration with respect to the master's, such that

$$x_s = K_k x_m \tag{3}$$

Then also holds $\dot{x}_s = K_k \dot{x}_s$ and $\ddot{x}_s = K_k \ddot{x}_m$.

Definition 1. *The system expressed by (1)-(2) is said* dynamically similar *after the coordination (3) if there exists a scalar $\bar{\zeta} > 0$ such that*

$$M_m \ddot{x}_m + B_m \dot{x}_m + K_m x_m = \frac{M_s \ddot{x}_s + B_s \dot{x}_s + K_s x_s}{\bar{\zeta}} \tag{4}$$

Thus, the dynamic parameters (M_i, B_i, K_i) of the master and those of the slave in (1)-(2) are linearly related by

$$X_s = \frac{\bar{\zeta}}{K} X_m = \zeta X_m, \qquad X_i = \begin{cases} M_i & \text{mass} \\ B_i & \text{damping} \\ K_i & \text{stiffness} \end{cases} \tag{5}$$

where the constant

$$\zeta \triangleq \frac{\bar{\zeta}}{K}$$

takes account of both kinematic and dynamic scalings.

The apparent advantage of this dynamic relationship has not been well explored in the context of teleoperation with unknown constant time delay, though there are a lot of heuristical intuition which leads us to conclude that these systems would allow greater kinesthetic coupling with greater manipulability dexterity as consequence. To this end, it is introduced an impedance control system [2] to enforce a 2-port closed-loop desired linear system.

3 Impedance Control Law

3.1 1 DoF Teleoperation System

Similarly to (1)-(2), let a 1 DoF master/slave teleoperation system be modeled as a mass-spring-damper system, where external master force F_m on the master is nothing but the human commanding force F_h and the external slave force force F_s stands as the environmental contact force F_e, then (1)-(2) becomes

$$M_m \ddot{x}_m + B_m \dot{x}_m + K_m x_m = F_{mc} + F_h \tag{6}$$

$$M_s \ddot{x}_s + B_s \dot{x}_s + K_s x_s = F_{sc} - F_e \tag{7}$$

The negative sign in F_e appears due to fact that this force is considered to be the force applied by the robot to the environment (in opposite direction as the force induced by the environment to the robot's end effector due to Newton's 3^{rd} Law), which is the same that is read by the force sensor in the robot and is available for force feedback.

3.2 Impedance Control Law for the Master

For completeness, the control law [2] is introduced here. Consider the following master controller

$$F_{mc} = -F_h + B_m\dot{x}_m + K_m x_m + \frac{M_m}{\bar{M}_m}(F_h - K_f F_e^{dy} - \bar{B}_m\dot{x}_m - \bar{K}_m x_m) \quad (8)$$

The control law (8) when applied to the open-loop system (6) gives rise to the following close-loop impedance equation for the master robot

$$\bar{M}_m\ddot{x}_m + \bar{B}_m\dot{x}_m + \bar{K}_m x_m = F_h - K_f F_e^{dy} \quad (9)$$

where the positive constants $\bar{M}_m, \bar{B}_m, \bar{K}_m$ are the desired values of inertia, damping and stiffness for the master robot. The constant gain K_f is the *force scaling factor* that makes the user (F_h) to feel a portion of the desired environmental contact force F_e^{dy}. The last is defined as the most recent real one in the slave, that can be read in the master, after the telecommunication transmission as

$$F_e^{dy} = F_e(t - T_s) \quad (10)$$

with T_s being the unknown but constant time-delay from the slave to the master station. That is, the *master* impedance control law enforces a desired impedance (9) in closed-loop, whose parameters are chosen by the user depending of a specific task, such that:

- If the slave robot *is not touching* the environment, $F_e^{dy} = 0$. Then (9) becomes a mass-spring-damper system driven solely by the human force F_h. Notice that in this case the controller is in *position impedance mode*.
- If the slave robot *does touch* the environment, $F_e^{dy} > 0$. Then (9) becomes a mass-spring-damper system driven by force error $F_h - K_f F_e^{dy}$. In this case, actuators in the master station makes the human to perceive a *scaled contact force* equal to $K_f F_e^{dy}$. Notice that when the slave is contact, the master control is in *force control mode*.

  In this mode the human could virtually recreate, cognitively, the surface of the object that the slave is touching according to this vector, through kinesthetic sensations and the visual information coming also from the slave station.

3.3 Impedance Control Law for the Slave

In order to achieve the desired effects in the master robot, the slave robot must be controlled in both impedance position and force control modes, according to the contact regime. Similarly to the master controller, the objective in the slave station is to impose a desired impedance to the slave robot

$$\bar{M}_s\ddot{\tilde{x}}_s + \bar{B}_s\dot{\tilde{x}}_s + \bar{K}_s\tilde{x}_s = -F_e \quad (11)$$

where positive $\bar{M}_s, \bar{B}_s, \bar{K}_s$ are the desired inertia, damping and stiffness for the slave robot, respectively. The position tracking error $\tilde{x}_s$ is expressed as follows

$$\tilde{x}_s = x_s - K_p x_m^{dy} \quad (12)$$

where the constant gain K_p is the *position scaling factor* that makes the slave to move a portion of the desired task space position, which in turn is defined as the most recent real one in the master, that can be read in the slave, after the telecomunication transmission as:

$$x_m^{dy} = x_m(t - T_m),$$

being T_m the unknown, but constant time-delay from the master to the slave station. Now, consider the following control law for the slave robot, presented in [2]:

$$F_{sc} = -\frac{M_s}{\bar{M}_s}\left(\bar{B}_s\dot{x}_s + \bar{K}_s x_s + F_e + K_i\sigma\right) + F_e + B_s\dot{x}_s - K_g\Omega$$
$$+M_s K_p \bar{M}_m^{-1}\left(F_h^{dy} - K_f F_e^{dy} - \bar{B}_m\dot{x}_m^{dy} - \bar{K}_m x_m^{dy}\right) \quad (13)$$

$$I_e = \bar{M}_s\ddot{x}_s + \bar{B}_s\dot{x}_s + \bar{K}_s x_s + F_e \quad (14)$$

$$\sigma = \int_0^t \text{sgn}\left\{I_e(\tau)\right\} d\tau \quad (15)$$

$$\Omega = \frac{1}{\bar{M}_m}\left(\int_0^t I_e(\tau)d\tau + \int_0^t\int_0^t \text{sgn}\left\{I_e(\tau)\right\} d\sigma d\tau\right) \quad (16)$$

where the desired force reference F_h^{dy} is defined as the delayed master's operator input force:

$$F_h^{dy} = F_h(t - T_m)$$

and F_e^{dy} as in (10).

The proposed sliding surface I_e in (14) is naturally proposed out of the desired close-loop (11), because it is desired that this function becomes an attractive convergent manifold. That is if (14) converge to zero, then (11) appears and the human would perceive the desired impedance at the robot and would be able to control the slave robot at will. Finally the high order sliding surface Ω is built as a function of the sliding surface I_e such that substitution of (13)-(16) into (7) gives rise to the closed-loop error equation for the slave robot:

$$\dot{\Omega} = -\beta\Omega \quad (17)$$

where $\beta = \frac{K_g}{\bar{M}_s} > 0$ and (17) is Lipschitz. Consequently, all closed-loop signals in the slave station are bounded, enforcing exponential convergence of $\Omega \to 0$. Therefore, this chain of implications means that a second order sliding mode is enforced, and a sliding mode arises, at $I_e = 0$, which means that (11) arises in finite-time.

Remarks:

1. Notice that the feedforward term F_e^{dy} in (13) allows control without delayed measurement of F_e. In any case, notice that F_e, being measured at this slave station is available at any time.
2. Notice that the gain K_g is a new control variable that weights the extended error variable Ω.
3. A closer analysis shows that the slave impedance control law enforces a desired impedance in closed-loop whose parameters are chosen by the user depending of a specific task, such that:

- when the slave robot *is not* touching the environment, $F_h^{dy} = 0, F_e = 0$ and (11) becomes an unforced mass-spring-damper system such that $\tilde{x}_s \rightarrow 0$ and the slave tracks the desired delayed position and velocities of the master. Notice that when the slave is not in contact, the slave control is in position impedance control mode.
- when the slave robot *is* touching the environment, $F_e > 0$ and (11) becomes a mass-spring-damper system driven by the slave contact force F_e. In this case, actuators in the slave station make that the slave robot maintains contact ($\mid \tilde{x}_s \mid > 0$) while F_e stays around F_h^{dy}. Notice that when the slave is in contact, the slave control is in impedance *force-position* control mode.

4 Delayed Teleoperation Stability

Last section presented the control law for each station, master and slave, and their individual stability analysis. However, the overall teleoperation stability system must be also addressed.

With the desired impedance imposed by the controllers (8) and (13)-(15), the closed-loop dynamics (9) and (16) can be modeled as a 2-port network. Transforming this dynamic into the frequency domain and doing some algebra, closed-loop system can be represented as

$$\begin{bmatrix} F_h \\ V_s \end{bmatrix} = \begin{bmatrix} h_{11}(s) & h_{12}(s) \\ h_{21}(s) & h_{22}(s) \end{bmatrix} \begin{bmatrix} V_m \\ -F_e \end{bmatrix} = H(s) \begin{bmatrix} V_m \\ -F_e \end{bmatrix} \tag{18}$$

where V_m and V_s are the velocity signals, in Laplace domain, of the master and slave respectively and $H(s)$ is the so called Hybrid Matrix. Using the dynamic similarity scales (5) the hybrid matrix is built out from elements depending on the master's desired impedance parameters as follows

$$H(s) = \begin{bmatrix} \frac{\bar{M}_m s^2 + \bar{B}_m s + \bar{K}_m}{s} & -K_f e^{-T_s s} \\ K_p e^{-T_m s} & \frac{1}{\zeta} \frac{s}{\bar{M}_m s^2 + \bar{B}_m s + \bar{K}_m} \end{bmatrix} \tag{19}$$

which is fundamental to carry out the implications of a unique dynamic scaling factor.

4.1 Absolute Stability

The absolute stability definition for a 2-port systems [2] is stated as:

Definition 2. ***Llewellyn's Absolute Stability Criteria for 2-port Systems****: A two-port system (18)-(19) is absolute stable if it does not exist a set of impedances for which the entire system become unstable. If the network is not absolutely stable, it is potentially unstable. By the conditions of the Llewellyn's criteria a 2-port network is absolutely stable if and only if*

1. *h_{11} and h_{22} have no poles in the right half plane*
2. *Any poles of h_{11} and h_{22} on the imaginary axis are simple with real and positives residues*

3. *For all real values of* ω
   (a) $\mathbb{R}\mathrm{e}\{h_{11}(j\omega)\} \geq 0$
   (b) $\mathbb{R}\mathrm{e}\{h_{22}(j\omega)\} \geq 0$
   (c) $2\mathbb{R}\mathrm{e}\{h_{11}(j\omega)\}\mathbb{R}\mathrm{e}\{h_{22}(j\omega)\} - \mathbb{R}\mathrm{e}\{h_{12}h_{21}(j\omega)\} - |h_{12}h_{21}(j\omega)| \geq 0$

Notice that since the human operator is physically holding the master robot with his hand, it is imperative to ensure stable behavior. Thus it is required to guarantee the fulfillment of previous Definition. To this end, when using (19), notice that conditions 1. and 2. are trivially satisfied with positive impedance parameters. For the third condition, with $s = j\omega$, a) and b) are also trivially satisfied while c) becomes:

Definition 3. ***The stability condition for delayed dynamically similar teleoperation systems****:*

$$K_p K_f \big(1 - \cos\{(T_m + T_s)\omega\}\big) \leq 2\nu(\omega)/\zeta \tag{20}$$

where the function $\nu(\omega)$ is defined as:

$$\nu(\omega) \triangleq \frac{(\bar{B}_m\omega)^2}{\big(\bar{K}_m - \bar{M}_m\omega^2\big)^2 + (\bar{B}_m\omega)^2} \tag{21}$$

with $0 < \nu(\omega) \leq 1$.

A conservative result, independent on the communication's time-delay channel, can be expressed using the bounds on the left-hand-side of condition (20):

$$0 \leq \big(\cos\{(T_m + T_s)\omega\} + 1\big)K_p K_f \leq 2K_p K_f$$

Then a delay-independent stability condition can be stated with the next expression:

$$\zeta K_p K_f \leq \nu(\omega) \tag{22}$$

4.2 Passivity Analysis

Passivity is a powerful criteria to analysis the energetic coupling of a closed loop system, a more conservative implication in comparison to Lyapunov stability criteria, however since the human operator is physically coupled with a typically mechanical system in closed-loop, it is important to analyze the passivity of the closed-loop system.

A two-port network is said to be passive if for all inputs of energy, the output energy is equal or less than the input energy. If the network is not passive, it is active. Raisbeck's passivity criterion is used to determine the passivity of the system.

Definition 4. ***Raisbeck Passivity Criteria for 2-port Systems****: It is said that a 2 port-network is passive if and only if*

1. *The elements* h_{ij} *of the hybrid matrix (19) have no poles in the right half plane*
2. *Any poles of the elements of the hybrid matrix on the imaginary axis are simple and their residues* r_{ij} *satisfy the following conditions, for all real values of* ω*,*
   (a) $r_{11} \geq 0$*,*

(b) $r_{22} \geq 0$,
(c) $r_{11}r_{22} - r_{12}r_{21} \geq 0$

3. *The elements* h_{ij} *of the hybrid matrix must satisfy the following condition:*

$$4\mathbb{R}\mathrm{e}\{h_{11}\}\mathbb{R}\mathrm{e}\{h_{22}\} - \left[\mathbb{R}\mathrm{e}\{h_{12}\} + \mathbb{R}\mathrm{e}\{h_{21}\}\right]^2 - \left[\mathrm{Im}\{h_{12}\} - \mathrm{Im}\{h_{21}\}\right]^2 \geq 0$$

In a similar way, the first two conditions are satisfied with positive impedance parameters, while the third is expressed as:

Definition 5. ***The passivity condition for delayed dynamically similar teleoperation systems:***

$$K_p^2 + K_f^2 + 2K_pK_f\cos\{(T_m - T_s)\omega\} \leq 4\nu(\omega)/\zeta \tag{23}$$

A conservative result, independent on the communication's time-delay channel, can be expressed using the bounds on the left-hand-side of condition (23):

$$(K_p - K_f)^2 \leq K_p^2 + K_f^2 + 2K_pK_f\cos\{(T_m - T_s)\omega\} \leq (K_p + K_f)^2$$

Then a delay-independent passivity condition can be stated with the next expression:

$$K_p + K_f \leq 2\sqrt{\frac{\nu(\omega)}{\zeta}} \tag{24}$$

5 Design Parametrization

5.1 The Bounding Function $\nu(\omega)$

The function ν, defined in (21), that is used in the stability and passivity conditions, express the bounds of those conditions in terms of a frequency ω. This frequency represents the spectrum of the transmitted signals for- and backwards in the communication channel. So at the end, this bounding function, and as consequence the stability and passivity conditions, depends also in the shape and richness of the transmitted signals for teleoperation.

It is always possible to filter the transmitted signal and cut-off noisy remanent frequencies beyond some allowed bandwidth. To select this parameter the bounding function ν can be rewritten in terms of only two impedance parameters instead of three. This is because the mass of the desired dynamics $\bar{M}_m$ is chosen to be very close to the actual mass of the system while the chosen parameters are the damping $\bar{B}_m$ and the rigidity $\bar{K}_m$. Then if close-loop natural frequency ω_n and the damping ratio ξ are chosen as the new parameters, defined as follows:

$$\omega_n = \sqrt{\frac{\bar{K}_m}{\bar{M}_m}}, \qquad \xi = \frac{\bar{B}_m}{2\sqrt{\bar{K}_m\bar{M}_m}} \tag{25}$$

and the frequency is normalized by the close-loop natural frequency, the bounded function is rewritten as:

$$\nu(\omega) = \frac{4\xi^2(\omega/\omega_n)^2}{\left(1 - (\omega/\omega_n)^2\right)^2 + 4\xi^2(\omega/\omega_n)^2} \tag{26}$$

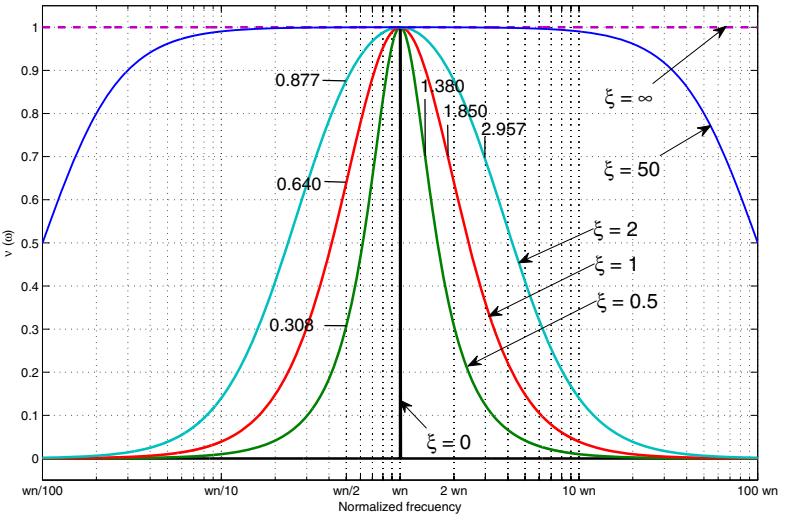


Fig. 2. Bounding function $\nu(\omega)$ for different values of damping ratio ξ, as function of the normalized frequency

Table 1. Bandwidth parameter $\alpha(\bar{\nu}, \xi)$

$\bar{\nu}$	$\xi = 1/2$	$\xi = 1/\sqrt{2}$	$\xi = 1$	$\xi = \sqrt{2}$	$\xi = 2$
0.1	3,303	4,467	6,162	8,602	12,083
0.2	2,414	3,146	4,236	5,828	8,123
0.3	2,022	2,552	3,353	4,541	6,270
0.4	1,785	2,189	2,806	3,732	5,095
0.5	1,618	1,932	2,414	3,146	4,236
0.6	1,488	1,732	2,107	2,682	3,548
0.7	**1,380**	1,565	**1,850**	2,289	**2,957**
0.8	1,281	1,414	1,618	1,932	2,414
0.9	1,180	1,263	1,387	1,577	1,869

$$\alpha(\bar{\nu}, \xi) = \sqrt{1 - 2\tilde{\nu}\xi^2 + 2\xi\sqrt{\tilde{\nu}(\tilde{\nu}\xi^2 - 1)}}$$

$$\frac{1}{\alpha(\bar{\nu}, \xi)} = \sqrt{1 - 2\tilde{\nu}\xi^2 - 2\xi\sqrt{\tilde{\nu}(\tilde{\nu}\xi^2 - 1)}}$$

$$\tilde{\nu} = 1 - \frac{1}{\bar{\nu}}$$

Figure 2 shows the shape of the bounding function ν for different values of damping ratio ξ. From (26) it can be calculated that the function gets its maximum at the natural frequency $\omega = \omega_n$ regardless of the damping ratio. The extreme solutions for the damping ratio, i.e. $\xi = 0$ (which yields $\nu = 0,\ \forall\omega \neq \omega_n$; with a singular value $\nu = 1$ at $\omega = \omega_n$) and $\xi = \infty$ (that yields to $\nu = 1,\ \forall\omega > 0$) are purely hypothetic cases. The first because there is no possible physical way to achieve a non dissipative mechanical system (there is always some kind of friction) for the closed-loop slave/master robotics systems. The second because it implies the use of a very large value of close-loop damping factor $\bar{B}_m$ that would render the closed-loop behavior extremely slow and energy consuming (and still not guarantying stability nor passivity). A more practical choice is that of picking a damping ratio in the vicinity of the critical value $\xi = 1$

(critically damped case). For these values, it can be seen in Figure 2 that there is an "acceptable" bandwidth:

$$\omega_b \triangleq \left[\frac{1}{\alpha}\omega_n, \alpha\omega_n\right],$$

centered at the chosen close-loop natural frequency ω_n. The size of this bandwidth, i.e. the value of α, depends on two factors. One is the chosen damping ratio ξ, but the other is an acceptable upper-bound $\bar{\nu}$ of the bounding function $\nu(\omega)$. Table 1 shows values of the bandwidth parameter $\alpha(\bar{\nu}, \xi)$ for given values of acceptable bounds $\bar{\nu}$ and damping ratio ξ. The bolded values in the table are also shown in Figure 2. On the other hand, Table 2 shows values of acceptable bounds of $\bar{\nu}(\alpha, \xi)$ for given values of the bandwidth parameter α and the damping ratio ξ. The bolded values are also shown in Figure 2.

Table 2. Upper bound $\bar{\nu}(\alpha, \xi)$

α	$\xi = 1/2$	$\xi = 1/\sqrt{2}$	$\xi = 1$	$\xi = \sqrt{2}$	$\xi = 2$
1.125	0, 947	0, 973	0, 986	0, 993	0, 997
1.25	0, 832	0, 908	0, 952	0, 975	0, 988
1.5	0, 590	0, 742	0, 852	0, 920	0, 958
2	**0, 308**	0, 471	**0, 640**	0, 780	**0, 877**
4	0, 066	0, 125	0, 221	0, 363	0, 532

$$\bar{\nu}(\alpha, \xi) = \frac{4\xi^2\alpha^2}{(1-\alpha^2)^2 + 4\xi^2\alpha^2}$$

$$\bar{\nu}(\alpha, \xi) = \frac{4\xi^2 \frac{1}{\alpha^2}}{\left(1-\frac{1}{\alpha^2}\right)^2 + 4\xi^2\frac{1}{\alpha^2}}$$

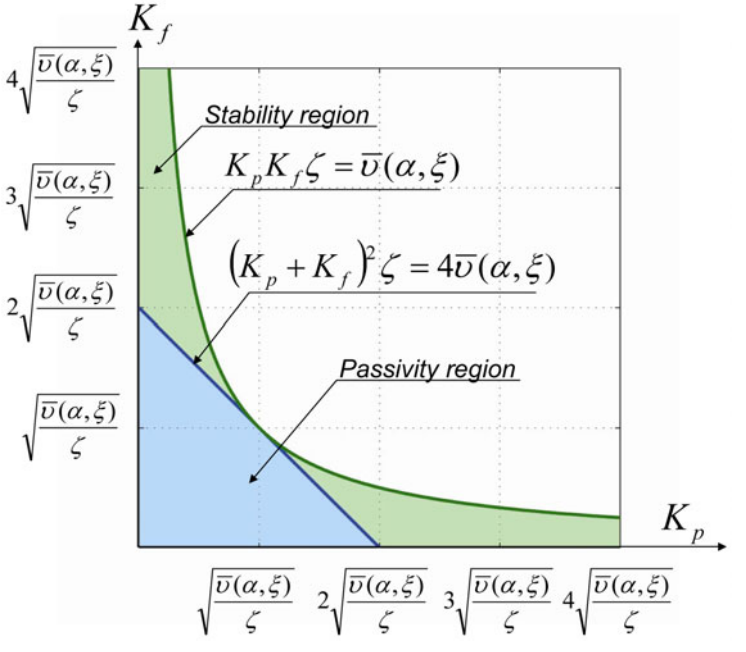


Fig. 3. Force/position scaling gains plot with stability and passive limits

5.2 Scaling Gains

Stability and passivity conditions expressed by (22) and (24) respectively can both be expressed in a phase plot of the scaling factors for position and force in the teleoperation scheme. This plot is shown in Figure 3.

The border curves in Fig. 3 are parametric dependant on the close-loop impedance desired parameters (mass, damping and stiffness) and hence by the induced natural frequency and damping ratio. Also they have dependance on the acceptable bandwidth which is parameterized also by upper bound $\bar{\nu}$ or frequency factor α, but more important they depend on the dynamic similarity scale factor. These borders defines the areas in which the values of the position and force scaling gains shall be placed in order to preserve stability or even more restrictively passivity. The borders are somewhat defined by the physical construction of the system by the dynamic similarity scaling factor but they can be replaced by changing the impedance parameters or the acceptable bandwidth transmission.

6 Conclusions

Using a novel impedance controller and advanced stability tools, precise conditions to guarantee stability, even in harsh conditions, is proposed for dynamically similar bilateral teleoperation robotic system. In this case, this system depends on a constant parameter, which relates explicitly and clearly a trade off between stability/passivity with scaling gains (position, force and dynamic similarity) and acceptable channel bandwidth transmission.

The controller enforces convergence in finite time due to the sliding surface, which is nothing but the impedance equation, thus the closed-loop system dynamics is entirely governed by the desired controlled equation. This yields useful boundaries to vary impedance, scaling parameters and frequency, in terms of the bounded time delay, which in turn allows to introduce a desired performance criteria in terms of surrounding physical conditions. This result seemingly allows to establish a simpler methodology to design dynamically similar teleoperators with desired forward and backward scaling gains, on a given desired performance in realistic conditions.

Therefore, the consequences of introducing a dynamic similar system in teleoperation, are:

1. scaling factors of position, force, and dynamical similarity ζ are critical for the absolute stability or the passivity performance of the system.
2. The introduction of a similarity factor ζ offers a simpler way to analyze the design and to tune the system performance.
3. The similarity factor ζ improves the design methodology of teleoperators based in impedance controllers. That is, there is a clear trade-off of all important parameters of the system: depending on impedance and the desired performance parameters(ω_n, ξ, $\bar{M}_m$, $\bar{B}_m$ and $\bar{K}_m$) and time delays, the scaling factors (position K_p, force K_f and dynamic similarity ζ) can be found.
4. From (20), the scaling factor ζ allows bigger margin on other parameters, thus the opportunity to improve the performance based on the physical structure of the teleoperation system.

5. Due to the fact that both master and slave impedance parameters are related by this factor, the whole set of parameters can be expressed in terms of each other, reducing the number of parameters implied in the design process, making it easier to establish a performance limit.

Then absolute stability is a more relaxed stability criteria than passivity. The passivity condition is necessary to assure a complete energetic stability performance of the closed loop system. Thus, we can choose the dynamic scaling factor in order to have a greater upper and lower bounds to vary the other factors of position and force, affecting also passivity. However the last can be chosen to set the performance parameters because it is of primary interest to maintain passivity since the human is physically holding the mechanical master robot.

References

1. Cho, H., Park, J.: Impedance controller design of internet-based teleoperation using absolute stability concept. In: IEEE Int.Conf. on Intelligent Robots and Systems, pp. 2256–2261 (2002)
2. Garcia-Valdovinos, L.G.: Teleoperadores Bilaterales No Lineales de Impedancia Convergente con Ambiente Semiestructurado. Phd. thesis, CINVESTAV (2006)
3. Goldfarb, M.: Similarity and invariance in scaled bilateral telemanipulation. ASME J. Dynam. Syst. Meas. Contr. 121, 79–87 (1999)
4. Hokayem, P., Spong, M.: Bilateral teleoperation: An historical survey. IEEE Control System Magazine 4, 30–34 (1984)
5. Hashtrudi-Zaad, K., Salcudean, S.E.: Analysis and evaluation of stability and performance robustness for teleoperation control architectures. In: IEEE Int. Conf. on Intelligent Robotics and Automation, pp. 3107–3113 (2000)
6. Li, P., Lee, D.: Passive bilateral feedforward control of linear dynamically similar teleoperated manipulators. IEEE Transactions on Robotic and Automation 19, 443–456 (2003)
7. Llewellyn, F.: Some fundamental properties of transmission systems. Proc. IRE 40, 271–283 (1952)
8. Deng, Q.W., Wei, Q., Li, Z.X.: Analysis of absolute stability for time-delay teleoperation systems. The International Journal of Automation and Computing, 203–207 (2007)
9. Raisbeck, G.: A definition of passive linear metworks in terms of time and energy. J. of Applied Physics 25 (1994)
10. Parra-Vega, V., García-Alvizo, V.P., Olguín-Díaz, E., García-Valdovinos, L.G.: Limits of human interaction in dynamical similar teleoperation, systems under unknown constant time-delay with impedance control. In: 6th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2009, pp. 102–109 (2009)

Part II

Robotics and Automation

Planning Image Trajectories for Visual Servoing via LMI-Based Optimization

Graziano Chesi

Department of Electrical and Electronic Engineering
University of Hong Kong, Pokfulam Road, Hong Kong
chesi@eee.hku.hk
http://www.eee.hku.hk/∼chesi

Abstract. Path-planning is a key technique in visual servoing, as it allows one to take into account various constraints and optimize a desired performance. This paper addresses the problem of planning image trajectories to be followed by using an image-based visual servoing controller. The proposed technique consists of parametrizing the camera trajectory in the 3D space via polynomials, and by imposing constraints satisfaction and performance optimization via the square matrix representation (SMR) and linear matrix inequalities (LMIs). Examples in the absence and in the presence of uncertainties illustrate and validate the proposed methodology.

1 Introduction

Visual servoing is an important area of robotics that concerns the closed-loop control of robotic system with visual feedback. The problem typically consists of steering a robot end-effector from an unknown initial location to an unknown desired location by using the visual information provided by a camera mounted on the robot end-effector. The camera is firstly located at a certain location, called desired location, and the image projections of some object points are recorded. Then, the camera is moved to another location of the robot workspace, from which the same object points are visible, and whose relative motion with respect to the desired location is unknown. The problem consists of reaching the desired location from this new location, which is called initial location. See for instance [1] and references therein.

The procedure just described, known as teaching-by-showing approach, has numerous applications, for instance in the industrial manufacture for the construction of complex components such as parts of a ship, where its function consists of allowing a robotic arm to grasp and position tools and objects. Other applications are in surveillance, where a mobile camera observes some areas of interest such as the entrance of a building in order to identify people, and in airplane alignment, where the system to be positioned is represented by the airplane that has to be aligned with respect to the runway in order to land. Also, the teaching-by-showing approach finds application in surgery, where an instrument is automatically guided to the organ to operate, in navigation, where a mobile robot has to explore a scene, and in dangerous environments such as nuclear stations and spatial missions, where humans should be replaced.

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 159–172.
springerlink.com

In last years, various methods have been developed for addressing the teaching-by-showing approach. Some of these methods have proposed the use of the camera pose as feedback information (known as position-based visual servoing, see e.g. [2]), definition of the feedback error in the image domain (known as image-based visual servoing, see e.g. [3]), use of both camera pose error and image error (known as $2\ 1/2$ D visual servoing, see e.g. [4]), partition of the degrees of freedoms [5], switching control for constraints satisfaction [6,7], generation of circular-like trajectories for ensuring global stability and minimizing the trajectory length [8], use of complex image features via image moments [9], global motion plan via navigation functions [10], enlargement of stability regions [11], and model-less control [12].

Path-planning strategies have also been proposed in order to take into account multiple constraints and optimize a desired performance. See for instance [13,14,15,16,17,18,19] and references therein. These methods generally adopt potential fields along a reference trajectory in order to fulfill the required constraints, in particular the potential fields do not affect the chosen reference trajectory wherever the constraints are not violated, while they make the camera deviating from this path wherever a constraint does not hold.

This paper addresses the problem of planning image trajectories to be followed by using an image-based visual servoing controller. The trajectories connecting the initial location to the desired one are parametrized through polynomials by estimating the initial camera pose and the position of the object points. Typical trajectory constraints and guaranteed level of performance are formulated in terms of positivity of certain polynomials, which is imposed by using the square matrix representation (SMR) and linear matrix inequalities (LMIs). Then, the image trajectory of the object points along the planned camera trajectory is tracked by using an image-based visual servoing controller. Examples in the absence and in the presence of uncertainties illustrate and validate the proposed methodology.

This paper extends our previous works [17] by exploiting LMIs for imposing constraints, and [18] by introducing alternative techniques for converting polynomial inequalities into LMIs.

The organization of the paper is as follows. Section 2 introduces the problem formulation and some preliminaries. Section 3 describes the proposed approach for planning image trajectories. Section 4 illustrates the simulation results. Lastly, Section 5 concludes the paper with some final remarks.

2 Preliminaries

In this section we introduce some preliminaries, namely the notation, problem formulation, and a tool for representing polynomials.

2.1 Notation and Problem Formulation

Let us start by introducing the notation adopted throughout the paper:

- $\mathbb{R}$: real numbers space;

- 0_n: $n \times 1$ null vector;
- I_n: $n \times n$ identity matrix;
- $\|v\|$: euclidean norm of vector v.

We consider a generic stereo vision system, where two cameras are observing a common set of object points in the scene. The symbols F^{ini} and F^{des} represent the frames of the camera in the initial and desired location respectively. These frames are expressed as

$$\begin{aligned} F^{ini} &= \{R^{ini}, t^{ini}\} \\ F^{des} &= \{R^{des}, t^{des}\} \end{aligned} \tag{1}$$

where $R^{ini}, R^{des} \in \mathbb{R}^{3\times3}$ are rotation matrices, and $t^{ini}, t^{des} \in \mathbb{R}^3$ are translation vectors. These quantities R^{ini}, R^{des}, t^{ini} and t^{des} are expressed with respect to an absolute frame, which is indicated by F^{abs}.

The observed object points project on the image plane of the camera in the initial and desired location onto the image points $p_1^{ini}, \ldots, p_n^{ini} \in \mathbb{R}^3$ (initial view) and $p_1^{des}, \ldots, p_n^{des} \in \mathbb{R}^3$ (desired view). These image points are expressed in homogeneous coordinates according to

$$p_i^{ini} = \begin{pmatrix} p_{i,1}^{ini} \\ p_{i,2}^{ini} \\ 1 \end{pmatrix}, \quad p_i^{des} = \begin{pmatrix} p_{i,1}^{des} \\ p_{i,2}^{des} \\ 1 \end{pmatrix}. \tag{2}$$

where $p_{i,1}^{ini}, p_{i,1}^{des} \in \mathbb{R}$ are the components on the x-axis of the image screen, while $p_{i,2}^{ini}, p_{i,2}^{des} \in \mathbb{R}$ are those on the y-axis. The projections p_i^{ini} and p_i^{des} are determined by the projective law

$$\begin{aligned} d_i^{ini} p_i^{ini} &= K R^{ini'} \left(q_i - t^{ini}\right) \\ d_i^{des} p_i^{des} &= K R^{des'} \left(q_i - t^{des}\right) \end{aligned} \tag{3}$$

where $d_i^{ini}, d_i^{des} \in \mathbb{R}$ are the depths of the ith point, $q_i \in \mathbb{R}^3$ is the ith point expressed with respect to F^{abs}, and $K \in \mathbb{R}^{3\times3}$ is the upper triangular matrix containing the intrinsic parameters of the camera.

The problem we consider in this paper consists of planning a trajectory from the initial location F^{ini} to the desired one F^{des} (which are unknown) by using the available estimates of:

1. the image projections $\hat{p}_1^{ini}, \hat{p}_1^{des}, \ldots, \hat{p}_n^{ini}, \hat{p}_n^{des}$;
2. and intrinsic parameters matrix $\hat{K}$.

This trajectory has to take into account physical constraints such as the limited field of view of the camera and the robot workspace, and has to optimize a desired performance.

In the sequel, we will indicate with $SO(3)$ the set of rotation matrices in $\mathbb{R}^{3\times3}$, and with $SE(3)$ the set of frames in the 3D space, where $SE(3) = SO(3) \times \mathbb{R}^3$.

2.2 Representation of Polynomials

Before proceeding, let us briefly introduced a tool for representing polynomials which will be exploited in the sequel. Let $p(x)$ be a polynomial of degree $2m$ in the variable

$x = (x_1, \ldots, x_n)' \in \mathbb{R}^n$, i.e.

$$p(x) = \sum_{\substack{i_1 + \ldots + i_n \leq 2m \\ i_1 \geq 0, \ldots, i_n \geq 0}} c_{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n} \tag{4}$$

for some coefficients $c_{i_1,\ldots,i_n} \in \mathbb{R}$. Then, $p(x)$ can be expressed as

$$p(x) = x^{\{m\}'} P(\alpha) x^{\{m\}} \tag{5}$$

where $x^{\{m\}}$ is any vector containing a base for the polynomials of degree m in x, and hence can be simply chosen as the set of monomials of degree less than or equal to m in x, for example via

$$x^{\{m\}} = (1, x_1, \ldots, x_n, x_1^2, x_1 x_2, \ldots, x_n^m)', \tag{6}$$

and

$$P(\alpha) = P + L(\alpha) \tag{7}$$

where $P = P'$ is a symmetric matrix such that

$$p(x) = x^{\{m\}'} P x^{\{m\}}, \tag{8}$$

while $L(\alpha)$ is a linear parametrization of the linear space

$$\mathcal{L}(n, m) = \{L = L' : \ x^{\{m\}'} L x^{\{m\}} = 0 \ \ \forall x\} \tag{9}$$

being α a vector of free parameters. The dimension of $x^{\{m\}}$ is given by

$$\sigma(n, m) = \frac{(n+m)!}{n!m!} \tag{10}$$

while the dimension of α (i.e., the dimension of $\mathcal{L}$) is

$$\tau(n, m) = \frac{1}{2}\sigma(n, m)(\sigma(n, m) + 1) - \sigma(n, 2m). \tag{11}$$

The representation in (5) was introduced in [20] with the name SMR in order to investigate positivity of polynomials via convex optimizations. Indeed, $p(x)$ is clearly positive if it is a sum of squares of polynomials, and this latter condition holds if and only if there exists α such that

$$P(\alpha) \geq 0 \tag{12}$$

which is an LMI. It turns out that, establishing whether an LMI admits a feasible solution or not, amounts to solving a convex optimization problem. The matrices P and $P(\alpha)$ are known as SMR matrices of $p(x)$, and can be computed via simple algorithms. See also [21,22,23].

3 Trajectory Planning

This section describes the proposed approach. Specifically, we first introduce the parametrization of the trajectory, we then describe its computation, and lastly we discuss the estimation of the initial camera pose and object points.

3.1 Trajectory Parametrization

Let us start by parameterizing the trajectory of the camera from the initial location to the desired one. This can be done by denoting the frame of the camera along the trajectory as

$$F(a) = \{R(a), t(a)\} \tag{13}$$

where $a \in [0,1]$ is the normalized trajectory abscise, $R(a) \in SO(3)$ is the rotation matrix of $F(a)$, and $t(a) \in \mathbb{R}^3$ is the translation vector. We choose the convention

$$\begin{array}{l} a = 0 \rightarrow F(a) = F^{ini} \\ a = 1 \rightarrow F(a) = F^{des}. \end{array} \tag{14}$$

The functions $R : [0,1] \rightarrow SO(3)$ and $t : [0,1] \rightarrow \mathbb{R}^3$ must satisfy the boundary conditions

$$\begin{array}{c} R(0) = \hat{R}^{ini},\ R(1) = \hat{R}^{des} \\ t(0) = \hat{t}^{ini},\quad t(1) = \hat{t}^{des} \end{array} \tag{15}$$

where $\hat{R}^{ini}$, $\hat{R}^{des}$, $\hat{t}^{ini}$ and $\hat{t}^{des}$ are the available estimates of R^{ini}, R^{des}, t^{ini} and t^{des} (the computation of these estimates will be addressed in Section 3.3). We adopt polynomials in order to parameterize $R(a)$ and $t(a)$. Specifically, we parameterize $t(a)$ according to

$$t(a) = \sum_{i=0}^{\delta} \check{t}_i a^i \tag{16}$$

where δ is an integer representing the chosen degree for $t(a)$, and $\check{t}_0, \ldots, \check{t}_\delta \in \mathbb{R}^3$ are vectors to be determined. Then, we parameterize $R(a)$ as

$$R(a) = \frac{E(r(a))}{\|r(a)\|^2} \tag{17}$$

where $E : \mathbb{R}^4 \rightarrow SO(3)$ is the parametrization of a rotation matrix via quaternions and $r : [0,1] \in \mathbb{R}^4$ is the parameter along the trajectory. In particular,

$$E(r) = \begin{pmatrix} r_1^2 - r_2^2 - r_3^2 + r_4^2 & 2(r_1 r_2 - r_3 r_4) & 2(r_1 r_3 + r_2 r_4) \\ 2(r_1 r_2 + r_3 r_4) & -r_1^2 + r_2^2 - r_3^2 + r_4^2 & 2(r_2 r_3 - r_1 r_4) \\ 2(r_1 r_3 - r_2 r_4) & 2(r_2 r_3 + r_1 r_4) & -r_1^2 - r_2^2 + r_3^2 + r_4^2 \end{pmatrix}. \tag{18}$$

It turns out that

$$E(r) \in SO(3)\ \ \forall r \in \mathbb{R}^4 \setminus \{0_4\}, \tag{19}$$

and moreover

$$\forall R \in SO(3)\ \exists \xi(R) \in \mathbb{R}^4 \setminus \{0_4\} :\ E(\xi(R)) = R. \tag{20}$$

A possible choice for $\xi(R)$ is given by

$$\xi(R) = \begin{pmatrix} \sin\dfrac{\theta}{2} u \\ \cos\dfrac{\theta}{2} \end{pmatrix} \tag{21}$$

where $\theta \in [0, \pi]$ and $u \in \mathbb{R}^3$, $\|u\| = 1$, are respectively the rotation angle and axis in the exponential coordinates of R, i.e.

$$R = e^{[\theta u]_\times}. \tag{22}$$

We parameterize $r(a)$ according to

$$r(a) = \sum_{i=0}^{\gamma} \check{r}_i a^i \tag{23}$$

where $\check{r}_0, \ldots, \check{r}_\gamma \in \mathbb{R}^4$ are vectors for some integer γ. The boundary conditions in (15) become

$$\begin{array}{ll} \check{r}_0 = \xi(\hat{R}^{ini}), & \sum_{i=0}^{\gamma} \check{r}_i = \xi(\hat{R}^{des}) \\ \check{t}_0 = \hat{t}^{ini}, & \sum_{i=0}^{\delta} \check{t}_i = \hat{t}^{des} \end{array} \tag{24}$$

which imply that $r(a)$ and $t(a)$ can be re-parameterized as

$$\begin{array}{l} r(a) = \left(\xi(\hat{R}^{des}) - \xi(\hat{R}^{ini}) - \sum_{i=1}^{\gamma-1} \bar{r}_i\right) a^\gamma + \sum_{i=1}^{\gamma-1} \bar{r}_i a^i + \xi(\hat{R}^{ini}) \\ t(a) = \left(\hat{t}^{des} - \hat{t}^{ini} - \sum_{i=1}^{\delta-1} \bar{t}_i\right) a^\delta + \sum_{i=1}^{\delta-1} \bar{t}_i a^i + \hat{t}^{ini} \end{array} \tag{25}$$

where $\bar{r}_1, \ldots, \bar{r}_{\gamma-1} \in \mathbb{R}^4$ and $\bar{t}_1, \ldots, \bar{t}_{\delta-1} \in \mathbb{R}^3$ are free vectors.

Let us observe that the derived parametrization can describe arbitrarily complicated trajectories, simply by selecting sufficiently large degrees γ and δ. Moreover, it is useful to observe that special cases such as straight lines are simply recovered by the choices

$$\begin{array}{c} \gamma = 1 \text{ (straight line in the domain of } E) \\ \delta = 1 \text{ (straight line in the translational space).} \end{array} \tag{26}$$

For ease of description we will assume $\gamma = 1$ in the following sections.

3.2 Trajectory Computation

In this section we address the problem of imposing constraints satisfaction and performance optimization on the parametrized trajectory via LMIs. Due to space limitation, we consider only two constraints, namely the visibility and the workspace constraints, and only one performance, namely minimizing the length of the trajectory.

Let us consider first the visibility constraint, and let us indicate with $p_i(a) = (p_{i,1}(a), p_{i,2}(a), 1)' \in \mathbb{R}^3$ the image projection of the ith object point along the trajectory. This point is visible whenever

$$p_{i,j}(a) \in (s_{i,1}, s_{i,2}) \ \ \forall i = 1, \ldots, n \ \forall j = 1, 2 \ \forall a \in [0, 1] \tag{27}$$

where $s_{1,1}, s_{1,2}, s_{2,1}, s_{2,2} \in \mathbb{R}$ are the screen limits. We estimate $p_i(a)$ via

$$p_i(a) = \frac{f_i(a)}{f_{i,3}(a)} + (1 - a)\left(\hat{p}_i^{ini} - \frac{f_i(0)}{f_{i,3}(0)}\right) + a\left(\hat{p}_i^{des} - \frac{f_i(1)}{f_{i,3}(1)}\right) \tag{28}$$

where $f_i(a) = (f_{i,1}(a), f_{i,2}(a), f_{i,3}(a))' \in \mathbb{R}^3$ is

$$f_i(a) = \hat{K}E(r(a))' \left(\hat{q}_i - t(a)\right) \tag{29}$$

and $\hat{q}_i \in \mathbb{R}^3$ is the estimate of the object point q_i (the computation of this estimate will be addressed in Section 3.3). Let us observe that this choice ensures $p_i(0) = \hat{p}_i^{ini}$ and $p_i(1) = \hat{p}_i^{des}$. We can rewrite $p_i(a)$ as

$$p_i(a) = \frac{1}{f_{i,3}(a)} \begin{pmatrix} g_{i,1}(a) \\ g_{i,2}(a) \\ f_{i,3}(a) \end{pmatrix} \tag{30}$$

where $g_{i,1}(a), g_{i,2}(a) \in \mathbb{R}$ are polynomials. Hence, visibility is ensured whenever

$$h(a) > 0 \;\; \forall h(a) \in \mathcal{H} \, \forall a \in [0,1] \tag{31}$$

where

$$\begin{aligned} \mathcal{H} = &\left\{ s_{j,k} f_{i,3}(a) - (-1)^k g_{i,j}(a) \; \forall i = 1, \ldots, n, \; j,k = 1,2 \right\} \\ &\cup \left\{ f_{i,3}(a) \; \forall i = 1, \ldots, n \right\}. \end{aligned} \tag{32}$$

For each polynomial $h(a)$ in $\mathcal{H}$, we introduce an auxiliary polynomial $u_h(a)$ of some degree, and we define

$$v_h(a) = h(a) - a(1-a)u_h(a). \tag{33}$$

We express these polynomials via the SMR as

$$\begin{aligned} u_h(a) &= y_h(a)' U_h y_h(a) \\ v_h(a) &= z_h(a)' V_h(\alpha_h) z_h(a) \end{aligned} \tag{34}$$

where $y_h(a), z_h(a)$ are vectors containing polynomial bases, and $U, V(\alpha_h)$ are symmetric SMR matrices (see Section 2.2 for details).

Then, let us consider the workspace constraint. A possible way to define the workspace constraint is via inequalities such as

$$d_i' \left(t(a_i) - o_i\right) > w_i \;\; \forall i = 1, \ldots, n_w \tag{35}$$

where $d_i \in \mathbb{R}^3$ is the direction along which the constraint is imposed, $a_i \in [0,1]$ specifies where the constraint is imposed on the trajectory, $o_i \in \mathbb{R}^3$ locates the constraint, $w_i \in \mathbb{R}$ specifies the minimum distance allowed from the point o_i along the direction d_i, and n_w is the number of constraints.

Lastly, let us consider the minimization of the length of the trajectory. A reasonable approximation that allows one to simply tackle this problem consists of minimizing any norm of the coefficients of the polynomial

$$s(a) = t(a) - \hat{t}^{ini} - \left(\hat{t}^{des} - \hat{t}^{ini}\right) a^k \tag{36}$$

where k in any integer satisfying $1 \leq k \leq \delta$. Indeed, let us observe that the polynomial $\hat{t}^{ini} + \left(\hat{t}^{des} - \hat{t}^{ini}\right) a^k$ describes a straight line from $\hat{t}^{ini}$ to $\hat{t}^{des}$ for any admissible

choice of k. Clearly, only the nonzero coefficients of $s(a)$ need to be considered, and we denote the scalar entries of these coefficients with $s_1, \ldots, s_{n_s}$ (observe in fact that $s(a)$ is a vector function).

The sought trajectory can be obtained by solving the optimization problem

$$z^* = \min z \ \text{ s.t. } \begin{cases} U_h > 0 \ \ \forall h(a) \in \mathcal{H} \\ V_h(\alpha_h) > 0 \ \ \forall h(a) \in \mathcal{H} \\ d_i' \left(t(a_i) - o_i \right) > w_i \ \ \forall i = 1, \ldots, n_w \\ z > s_i \ \ \forall i = 1, \ldots, n_s \\ z > -s_i \ \ \forall i = 1, \ldots, n_s \end{cases} \tag{37}$$

where $z \in \mathbb{R}$ is an additional variable. This problem belongs to the class of eigenvalue problems and is a convex optimization problem with LMI constraints. Once the problem has been solved, one can obtain the 3D trajectory from (25), and the sought image trajectory from (27)–(29). Observe that $z^* = 0$ implies that the 3D trajectory is a straight line.

3.3 Camera Pose and Scene Estimation

In the previous sections we have described how the trajectory of the camera can be parameterized and computed. In particular, the parametrization was based on the estimates $\hat{R}^{ini}$, $\hat{R}^{des}$, $\hat{t}^{ini}$ and $\hat{t}^{des}$ of the components of the initial and desired frames F^{ini} and F^{des}, while the computation was based on the estimates $\hat{q}_1, \ldots, \hat{q}_n$ of the object points $q_1, \ldots, q_n$. Here we describe some ways to obtain these estimates.

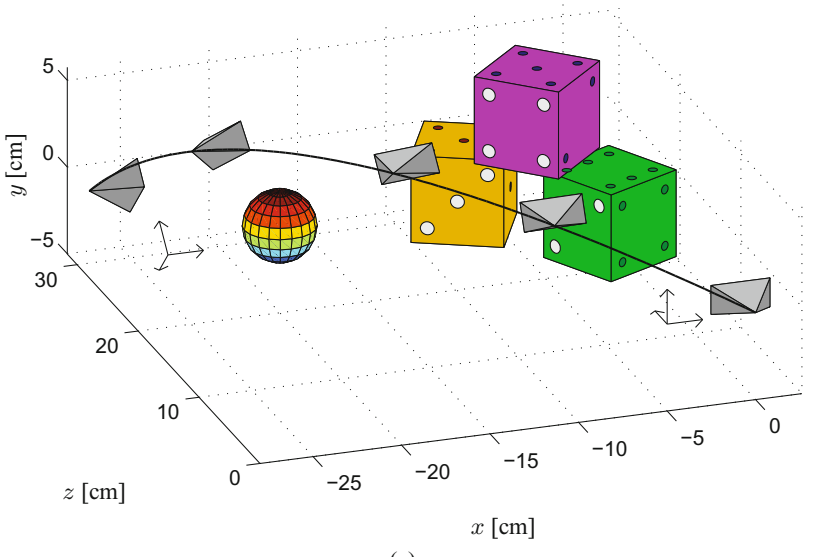
Given the estimates $\hat{p}_1^{ini}, \hat{p}_1^{des}, \ldots, \hat{p}_n^{ini}, \hat{p}_n^{des}$ of the image projections and $\hat{K}$ of the intrinsic parameters matrix, one can estimate the camera pose between F^{ini} and F^{des}, and hence R^{ini} and t^{ini} since F^{des} can be chosen without loss of generality equal to F^{abs}. This estimation can be done, for example, through the essential matrix or through the homography matrix, see for instance [24,25,26] and references therein.

Once that the estimates $\hat{R}^{ini}$ and $\hat{t}^{ini}$ have been found, one can compute the estimates $\hat{q}_1, \ldots, \hat{q}_n$ of the object points via a standard triangulation scheme, which amounts to solving a linear least-squares problem.

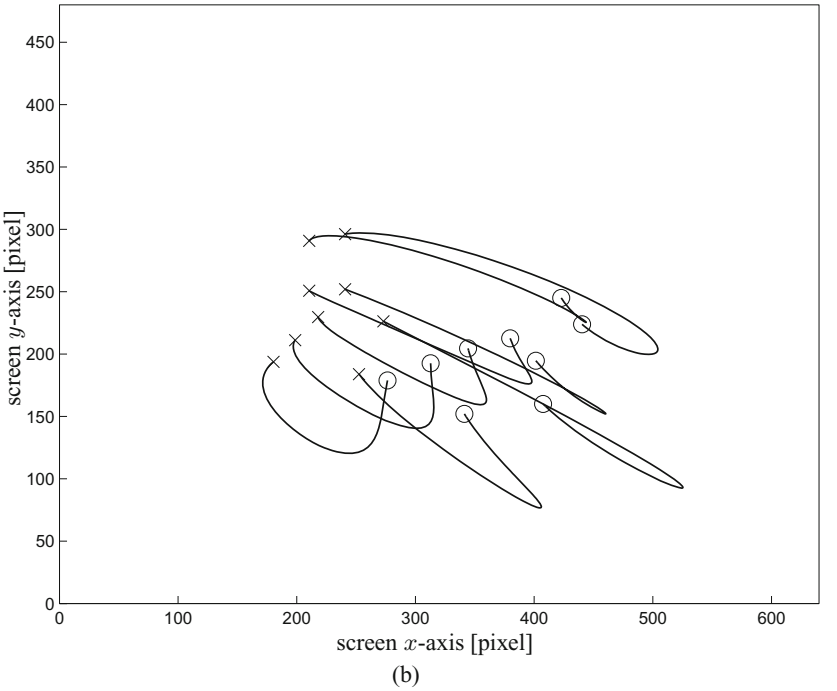
Let us observe that, if no additional information is available, the translation vector and the object points can be estimated only up to a scale factor. In this case, the workspace constraint has to be imposed in a normalized space. This problem does not exist if a CAD model of the object (or part of it) is available, since this allows to estimate the distance between the origins of F^{ini} and F^{des}.

4 Illustrative Example

In this section we present an illustrative example of the proposed approach. Let us consider the situation shown in Figure 1a, where a camera observes some object points (the centers of the nine large dots in the “2”, “3” and ”4” faces of the three dices) from the initial and desired locations (leftmost and rightmost cameras respectively).

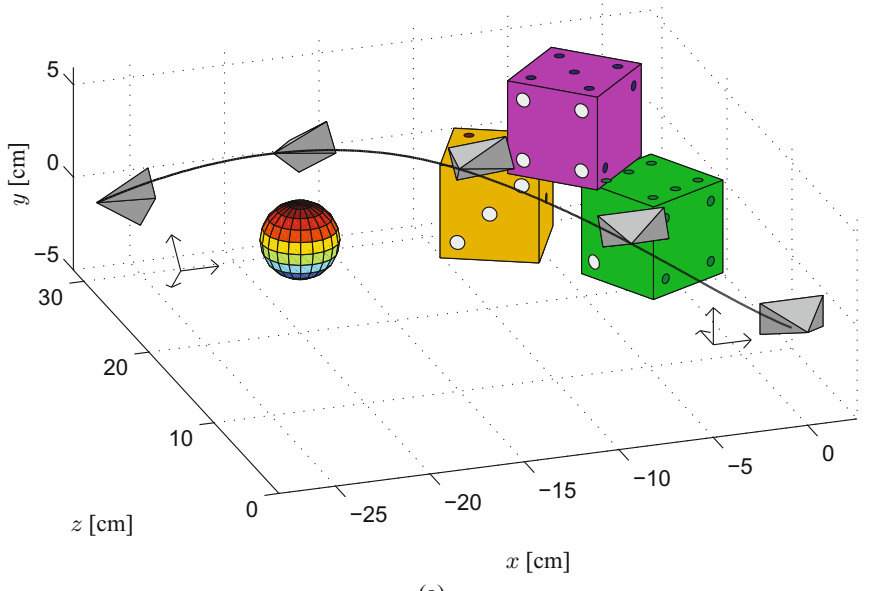


(a)

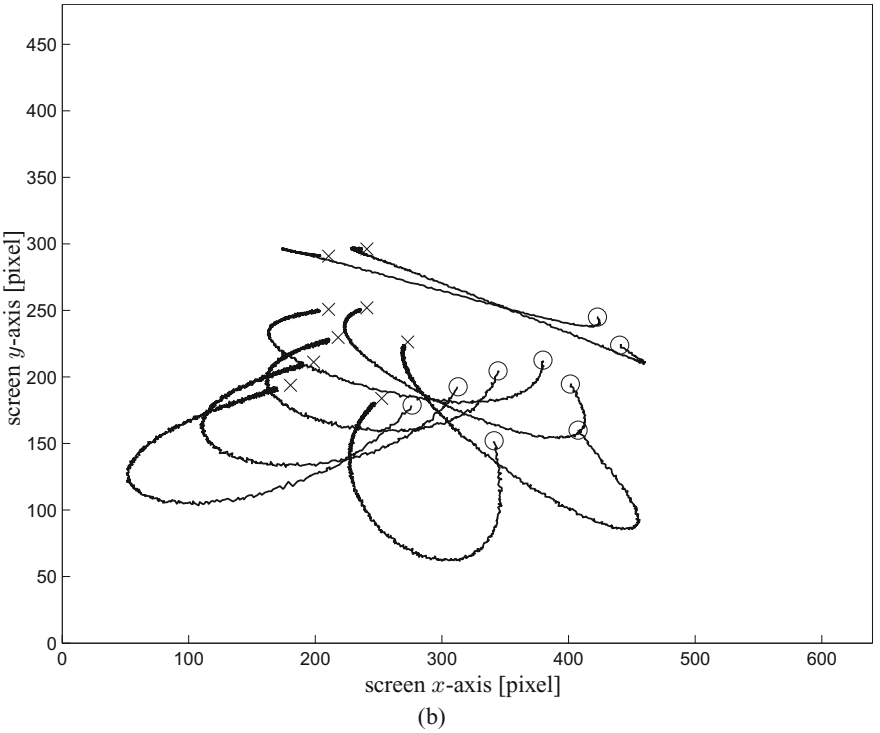


(b)

Fig. 1. (a) Initial frame F^{ini} (leftmost camera), desire frame F^{des} (rightmost camera), object points (centers of the nine large dots in the "2", "3" and "4" faces of the three dices), planned trajectory (solid line), and some intermediate locations of the camera along the planned trajectory. (b) Image projections of the object points in the initial view ("o" marks) and desired view ("x" marks), and image projection of the planned trajectory (solid line).

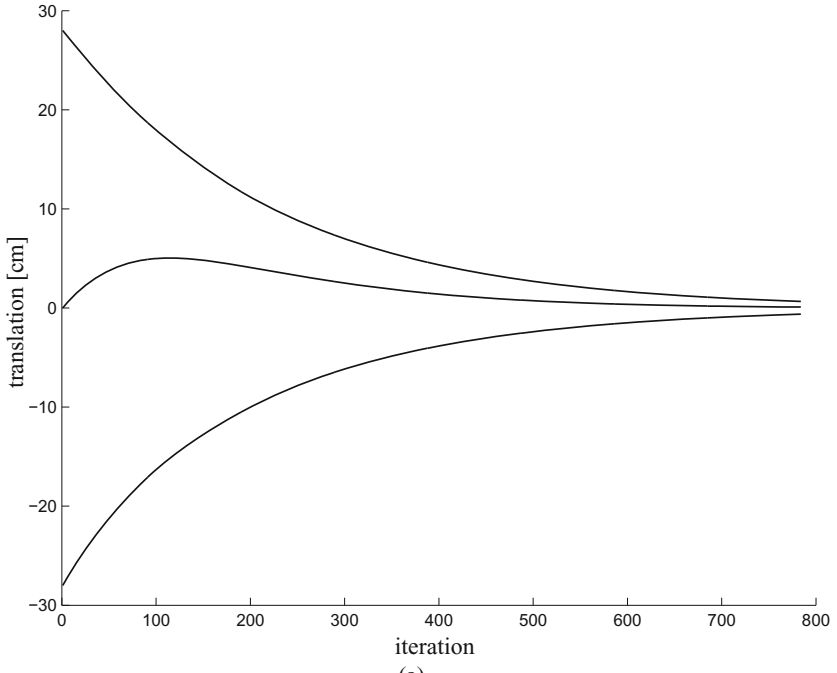


(a)



(b)

Fig. 2. Results obtained by planning the trajectory in the presence of image noise and calibration errors, and by tracking the planned trajectory with an image-based controller



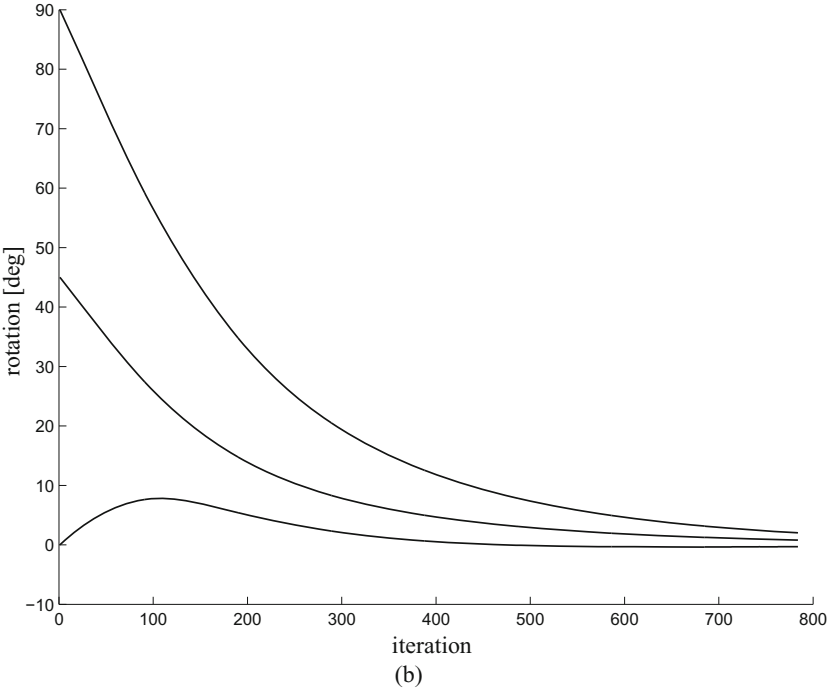


Fig. 3. Camera coordinates during the motion shown in Figure 2

Figure 1b shows the image projections of these points in the initial view ("o" marks) and desired view ("x" marks). The intrinsic parameters matrix is chosen as

$$K = \begin{pmatrix} 400 & 0 & 320 \\ 0 & 300 & 240 \\ 0 & 0 & 1 \end{pmatrix}. \tag{38}$$

The problem consists of planning a trajectory from the initial location to the desired one which ensures that the object points are kept inside the field of view of the camera and the camera does not collide with the sphere interposed between F^{ini} and F^{des} (which represents an obstacle to avoid). It is supposed that the position of the obstacle is known with respect to F^{des}, and that the distance between two points of the dices is known.

Let us use the proposed approach. We parameterize the trajectory as described in Section 3.1 with polynomials of degree two by estimating the camera pose between F^{ini} and F^{des} via the essential matrix. We build the set of polynomials $\mathcal{H}$, which impose the visibility constraint, and we compute the SMR matrices U_h and $V_h(\alpha_h)$ in (34). The workspace constraint is chosen by requiring that the trajectory must remain at a certain distance from the obstacle in three directions. Then, we solve the eigenvalue problem (37), and we build the 3D trajectory from (25) and the image trajectory from (27)–(29), which are shown in Figures 1a and 1b.

Now, in order to introduce typical uncertainties of real experiments, we corrupt the image projections of the object points by adding image noise with uniform distribution in $[-1, 1]$ pixels to each component. Moreover, we suppose that the camera is coarsely calibrated, in particular the available estimate of the intrinsic parameters matrix is

$$\hat{K} = \begin{pmatrix} 430 & 0 & 338 \\ 0 & 275 & 250 \\ 0 & 0 & 1 \end{pmatrix}. \tag{39}$$

We repeat the previous steps in the presence of these uncertainties, and then we track the planned trajectory by using the image-based controller proposed in [13]. Figures 2a and 2b show the obtained results: as we can see, the camera reaches the desired location by avoiding collisions with the obstacle in spite of the introduced uncertainties. Figures 3a and 3b show the camera coordinates during the motion.

5 Conclusions

This paper has proposed an approach for planning image trajectories in visual servoing based on the use of polynomials and convex optimization problems. It has been shown how constraints satisfaction and performance optimization can be imposed on the sought trajectory through LMIs by adopting the SMR. The planned image trajectory is then tracked by using an image-based visual servoing controller.

Acknowledgements. This work was supported by the Research Grants Council of Hong Kong under Grant HKU711208E.

References

1. Chesi, G., Hashimoto, K. (eds.): Visual Servoing via Advanced Numerical Methods. Springer, Heidelberg (2010)
2. Thuilot, B., Martinet, P., Cordesses, L., Gallice, J.: Position based visual servoing: keeping the object in the field of vision. In: IEEE Int. Conf. on Robotics and Automation, Washington, D.C, pp. 1624–1629 (2002)
3. Hashimoto, K., Kimoto, T., Ebine, T., Kimura, H.: Manipulator control with image-based visual servo. In: IEEE Int. Conf. on Robotics and Automation, pp. 2267–2272 (1991)
4. Malis, E., Chesi, G., Cipolla, R.: 2 1/2 D visual servoing with respect to planar contours having complex and unknown shapes. Int. Journal of Robotics Research 22(10), 841–853 (2003)
5. Corke, P.I., Hutchinson, S.: A new partitioned approach to image-based visual servo control. IEEE Trans. on Robotics and Automation 17(4), 507–515 (2001)
6. Chesi, G., Hashimoto, K., Prattichizzo, D., Vicino, A.: Keeping features in the field of view in eye-in-hand visual servoing: a switching approach. IEEE Trans. on Robotics 20(5), 908–913 (2004)
7. Lopez-Nicolas, G., Bhattacharya, S., Guerrero, J.J., Sagues, C., Hutchinson, S.: Switched homography-based visual control of differential drive vehicles with field-of-view constraints. In: IEEE Int. Conf. on Robotics and Automation, Rome, Italy, pp. 4238–4244 (2007)
8. Chesi, G., Vicino, A.: Visual servoing for large camera displacements. IEEE Trans. on Robotics 20(4), 724–735 (2004)
9. Tahri, O., Chaumette, F.: Point-based and region-based image moments for visual servoing of planar objects. IEEE Trans. on Robotics 21(6), 1116–1127 (2005)
10. Cowan, N.J., Chang, D.E.: Geometric visual servoing. IEEE Trans. on Robotics 21(6), 1128–1138 (2005)
11. Tarbouriech, S., Soueres, P., Gao, B.: A multicriteria image-based controller based on a mixed polytopic and norm-bounded representation of uncertainties. In: 44th IEEE Conf. on Decision and Control and European Control Conf., Seville, Spain, pp. 5385–5390 (2005)
12. Miura, K., Hashimoto, K., Inooka, H., Gangloff, J.A., de Mathelin, M.F.: Model-less visual servoing using modified simplex optimization. Journal Artificial Life and Robotics 2, 131–135 (2006)
13. Mezouar, Y., Chaumette, F.: Path planning for robust image-based control. IEEE Trans. on Robotics and Automation 18(4), 534–549 (2002)
14. Deng, L., Janabi-Sharifi, F., Wilson, W.J.: Hybrid motion control and planning strategy for visual servoing. IEEE Trans. on Industrial Electronics 52(4), 1024–1040 (2005)
15. Yao, Z., Gupta, K.: Path planning with general end-effector constraints. Robotics and Autonomous Systems 55(4), 316–327 (2007)
16. Chesi, G., Prattichizzo, D., Vicino, A.: Straight line path-planning in visual servoing. Journal of Dynamic Systems, Measurement and Control 129(4), 541–543 (2007)
17. Chesi, G., Hung, Y.S.: Global path-planning for constrained and optimal visual servoing. IEEE Trans. on Robotics 23(5), 1050–1060 (2007)
18. Chesi, G.: Visual servoing path-planning via homogeneous forms and LMI optimizations. IEEE Trans. on Robotics 25(2), 281–291 (2009)
19. Kazemi, M., Gupta, K., Mehran, M.: Global path planning for robust visual servoing in complex environments. In: IEEE Int. Conf. on Robotics and Automation, Kobe, Japan (2009)
20. Chesi, G., Tesi, A., Vicino, A., Genesio, R.: On convexification of some minimum distance problems. In: European Control Conf., Karlsruhe, Germany (1999)
21. Chesi, G., Garulli, A., Tesi, A., Vicino, A.: Solving quadratic distance problems: an LMI-based approach. IEEE Trans. on Automatic Control 48(2), 200–212 (2003)

22. Chesi, G.: On the gap between positive polynomials and SOS of polynomials. IEEE Trans. on Automatic Control 52(6), 1066–1072 (2007)
23. Chesi, G., Garulli, A., Tesi, A., Vicino, A.: Homogeneous Polynomial Forms for Robustness Analysis of Uncertain Systems. Springer, Heidelberg (2009)
24. Malis, E., Chaumette, F.: 2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. Int. Journal of Computer Vision 37(1), 79–97 (2000)
25. Chesi, G., Hashimoto, K.: A simple technique for improving camera displacement estimation in eye-in-hand visual servoing. IEEE Trans. on Pattern Analysis and Machine Intelligence 26(9), 1239–1242 (2004)
26. Chesi, G.: Camera displacement via constrained minimization of the algebraic error. IEEE Trans. on Pattern Analysis and Machine Intelligence 31(2), 370–375 (2009)

Evolutionnary Autonomous VGSTV Staircase Climbing

Jean-Luc Paillat, Phillipe Lucidarme, and Laurent Hardouin

Laboratoire d'Ingénierie des Systèmes Automatisés (LISA)
62 Avenue Notre Dame du Lac, Angers, France
jlpaillat@gmail.com,
{philippe.lucidarme,laurent.hardouin}@istia.univ-angers.fr

Abstract. This paper introduces an originally designed tracked robot. This robot belongs to the VGSTV (Variable Geometry Single Tracked Vehicle) category, which is actually a sub-group of Variable Geometry Vehicles well represented by the iRobot Packbot. Those robots have been used several times for search and rescue mission and seems to be a real asset because of their clearing capability. After a brief categorization, technical specification of our robot are presented and geometrical and dynamical models are computed in order to compare static and dynamic balance. Then, thanks to the results of the balance study, an autonomous staircase clearing controller based on artificial neural network is computed and tested. A general conclusion about possible improvements and future work ends the paper.

1 Introduction

UNMANNED GROUNDED VEHICLE (UGV) is a topically research field applied to a wide range of applications like for example exploration or missions in hostile environments. Research laboratories and robotics companies are currently working on the design of tele-operated and autonomous robots. According to [3] and [2] UGVs can be classified into three categories: Man-packable, Man-portable and Not man-portable.

The robots presented in this paper are classified in the man-packable and man-portable categories. In this class of robots, designers have to face the following dilemma: on one hand, build a small robot that can be easily carried and move into narrow environments. Unfortunately, it will generally result in poor obstacle clearing capability. On the other hand, build a bigger robot will increase its ability to surmount obstacles but will not enable the robot to go through narrow openings. The challenge is then to build a robot as small as possible with the higher obstacle clearing capability. Based on this observation the first part of this paper introduces the existing experimental and commercial robots and discusses about their clearing capabilities. The following of the paper describes an originally designed UGV (Fig. 3). This robot can be classified into the Variable Geometry Single Tracked Vehicle (VGSTV) category, i.e. it has the mechanical ability to modify its own shape according to the ground configuration. The design of our prototype is described in the third part with a short discussion about the technical choices (information can be found on the project website: http://www.istia.univ-angers.fr/LISA/B2P2/b2p2.html). The next section introduces the dynamic model of the robot. The first tele-operated experiments have shown that the

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 173–186.
springerlink.com

mass distribution is crucial to pass through large obstacles. Then, thanks to the results of the balance study, an autonomous staircase clearing controller based on artificial neural network is computed and tested. A general conclusion about possible improvements and future work ends the paper.

2 Existing UGVs

2.1 Wheeled and Tracked Vehicles with Fixed Shape

This category gathers non variable geometry robots (fig. 1a and b). Theoretically, this kind of vehicles are able to climb a maximum step twice less high than their wheel diameter. Therefore their dimensions are quite important to ensure a large clearing capability. This conception probably presents a high reliability [1] but those robots cannot be easily used in unstructured environments like after an earthquake [3].

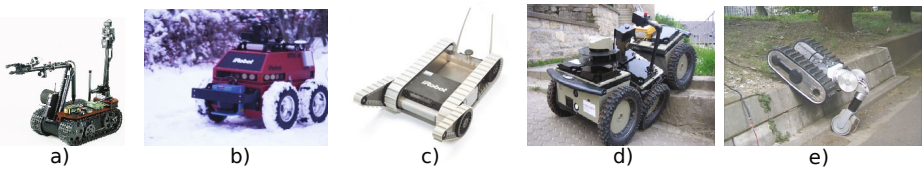


Fig. 1. a): Talon-Hazmat robot (Manufacturer: Foster-Miller), b): ATRV-Jr robot. Photo Courtesy of AASS, rebro University c): Packbot (manufacturer: IRobot), d): RobuROC 6 (Manufacturer: Robosoft), e): Helios VII.

2.2 Variable Geometry Vehicle

A solution to ensure a large clearing capability and to reduce the dimensions consists in developing tracked vehicles which are able to modify their geometry in order to move their center of mass and climb higher obstacles than their wheel's diameters.

The Packbot robot (Fig. 1c) is probably one of the most famous commercial VGTV (Variable Geometry Tracked Vehicle). This robot is equipped with tracks and two actuated tracked flippers (372 mm long). The flippers are used to step over the obstacles. The obstacle clearing capability of this kind of VGTV depends on the size of the flippers. For more information and a detailed survey on clearance capability of the Packbot the reader can consult [6].

The robuROC6 (Fig. 1d) is equipped with 46.8 cm diameters wheels and can clear steps until 25cm (more than half the diameter of the wheels). Joints between the axles make this performance possible. An other original system called Helios VII (Fig 1e) [7] is equipped with an arm ended by a passive wheel which is able to elevate the chassis along a curb.

2.3 Variable Geometry Single Tracked Robots

Actually, there is a subgroup in VGTV called Variable Geometry Single-Tracked Vehicles (VGSTV) [13]. It gathers robots equiped with as tracks as propulsion motors. In

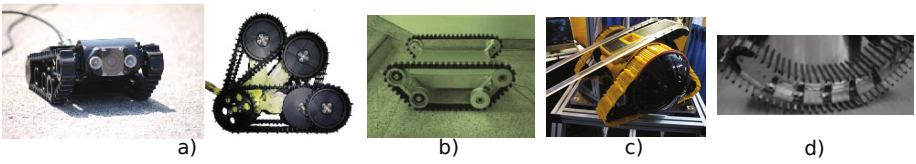


Fig. 2. a): Micro VGTV (manufacturer: Inuktun Ltd) b): VGSTV mechanism c): Viper robot (Manufacturer: Galileo) d): Rescue mobile track WORMY

most cases those robots are equipped with one or two tracks (one for each side). It can be divided into two groups:

- robots with deformable tracks,
- robots with non deformable tracks.

Non-deformable Tracks VGSTV. The most famous example is the Micro VGTV. Illustrations of a prototype manufactured by the company Inuktun are presented on Fig. 2a. This robot is based on an actuated chassis used to modify the shape of the robot. The right picture of Fig. 2a shows the superimposing configurations. The tracks are kept tightened by a passive mechanism. The robot is thus equipped with three motors: two for the propulsion and one for the chassis joint.

Non commercial vehicles exists in the literature as the VGSTV mechanism (Fig. 2b) which is dedicated to staircase clearing. It is composed of two tracks and two articulations which allow it to have many symmetrical configurations such as a rectangle, trapezoids, inverse trapezoids etc.

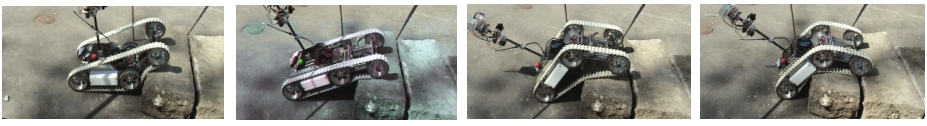
Many other VGTV architecture exist, for further information reader can consult [18], [15], [4], [7] and [13].

Deformable Tracks VGSTV. Some single tracked robots have the ability to modify the flexing of their tracks. Two examples presented on Fig. 2 c and d, are able to adapt their shape to obstacles [12]. However, even if the control of the robot seems easier with a flexible track than with a non flexible one, the mechanical conception could be more complicated.

According to the presented state of the art, for general purpose missions we beleive that the best compromise between design complexity, reliability, cost and clearing capabilities is the Variable geometry single tracked robots category. The next section will introduce and describe our prototype of VGSTV.

3 Prototype Description

The main interest of VGSTV (equipped with deformable tracks or not) is that it is practicable to overcome unexpected obstacles [13]. Indeed, thanks to the elastic property of the tracks the clearance of a rock in rough terrain will be more smoothly with a VGSTV (e. g. Fig 2) than with a VGTV (Fig. 1 c, d and e). On the Micro VGTV presented on



a) b) c) d)

Fig. 3. B2P2: Clearing of a curb

Fig. 2, the tension of the tracks is mechanically linked with the chassis joint so it is constant during the movement. Nevertheless, in some cases, less tense tracks could increase the clearance capability by increasing the adherence. An interesting study about this point was developed by [8] giving a VGSTV able to climb staircases where the tension of the tracks was mechanically managed as on the MicroVGTV (Fig. 2a). However, this system was equipped with a spring to allow the tracks to adapt their shape to the ground (depending on the strength of the spring).

The conception of our prototype is based on this previous work, but we decided to actuate the tension of the tracks. Indeed, by using two motors instead of one (Fig. 4) it is possible to increase the tracks adaptation to the ground developed by [8] and reach new configurations for the robot. As example, the solution proposed in this paper allows our robot to adopt classical postures of VGSTV (Fig. 3(a), 3(b) and 3(c)), but also other interesting positions. On Fig. 3 B2P2 is clearing a curb of 30 cm height with tense tracks. The position of the robot on Fig 3(c) can also be obtained with the Micro VGTV, but it is a non-safety position and B2P2 is close to topple over. On Fig. 3(d) the tracks have just been released. They take the shape of the curb and it can be cleared safely. This last configuration outlines the interest of using an active system instead of a passive one. Consequently, although our prototype (Fig. 2b) belong to the VGSTV category and have not deformable tracks, it has the ability to adapt them to the ground (as deformable ones).

Besides, even if there is a risk of the tracks coming off, loosening the tracks may be an efficient mean of increase the surface in contact with the floor in rough terrain and then to improve the clearing capability of the structure. By the way, the risk could decreased by using sensor based systems to control the tension of the tracks or by modifying the mechanical structure of the robot (adding some kind of cramps on the tracks or using a guide to get back the tracks before it comes off).

3.1 Mechanical Description

This UGV is equipped with four motors. Fig. 4 presents the integration of the motors in the robot. Motors 1 and 2 are dedicated to the propulsion (tracks).

The actuated front part is composed of motors 3 and 4:

- Motor 3 actuates the rotational joint, it allows the rotation of the front part around the second axle.
- Motor 4 actuates a driving screw, it controls the distance between the second axle and the third one.

To keep the tension of the tracks the trajectory of the third axle is given by an ellipse defined by two seats located on the first and the second axle.

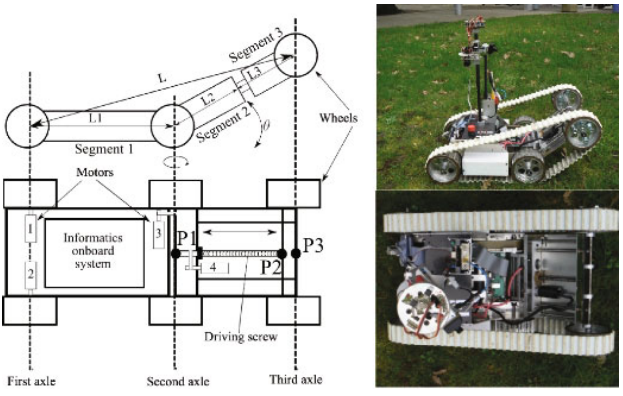


Fig. 4. Overview of the mechanical structure, side and top view of the real robot

$$L + (L_2 + L_3) = K \tag{1}$$

where the lengths L, L_2 and L_3 are referenced on Fig. 4. K is a constant parameter depending on the length of the tracks, L_3 evolves in order to achieve equality (1) and is linked to the angle θ in the following manner:

$$L_3 = \frac{L_1^2 - K^2}{2(L_1 cos(\pi - \mid \theta \mid) - K)} - L_2 \tag{2}$$

4 Dynamic Model

This section deals with the dynamic model of the robot which is based on the geometric model (Fig. 5) detailed on [17].

According to this model, the robot motion in a 3D frame (R_0) is described by the vector q of the 8 joints variables:

$$q = [q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8]^T$$

The dynamic model of a mechanical system establishes a relation between the effort applied on the system and its coordinates, generalized speeds and accelerations ([5] and [10]). In this section, the following notations are used :

- j describes the joints from 1 to 8,
- i describes the segments from 1 to 3 (referenced on Fig 4),
- n and m describes indexes from 1 to 8.

4.1 The Dynamic Equations

The general dynamic equations of a mechanical system is:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q_j}} - \frac{\partial L}{\partial q_j} = Q_j + T_j \tag{3}$$

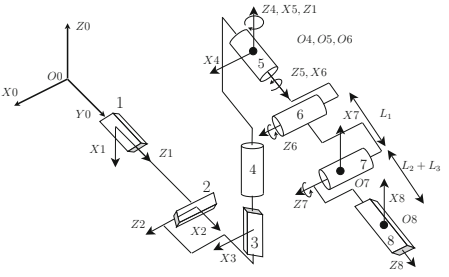


Fig. 5. B2P2's geometric model. Joints 1,2 and 3 represents the robot position. Joints 4, 5 and 6 symbolize respectively the yaw, roll and pitch, 7 and 8 are the actuated joints.

- L is the Lagrangien of the system. It is composed of rigid segments, so there is no potential energy. Although the Lagrangien corresponds to the kinetic energy.
- q_j is the j^{th} joint variable of the system.
- Q_j is the gravity's torque applied to the j^{th} joint of the system.
- T_j is the external force's torque applied to the j^{th} joint of the system.

The kinetic energy is given by:

$$K = \sum_{i=1}^{n} \frac{1}{2} m_i v_i^T v_i + \frac{1}{2} w_i^T I_i w_i. \tag{4}$$

- m_i is the mass of the i^{th} element of the model,
- v_i is the linear speed of the i^{th} element's center of gravity,
- w_i is the angular speed of the i^{th} element's center of gravity,
- I_i is the matrix of inertia of the i^{th} element of the system.

In order to have homogeneous equations, w_i is defined in the same frame as I_i ; it allows to formulate v_i and w_i according to q:

$$v_i = J_{v_i}(q)\dot{q} \tag{5}$$

$$w_i = R_{0j}^T J_{w_i}(q)\dot{q} \tag{6}$$

where J_{v_i} and J_{w_i} are two matrices and R_{0j} is the transport matrix between the frame R_0 and the frame j linked to the segment i.

The kinetic energy formula is:

$$K = \frac{1}{2}\dot{q}^T \sum_i [m_i J_{v_i}(q)^T J_{v_i}(q) + J_{w_i}^T(q) R_{0j} I_i R_{0j}^T J_{w_i}(q)]\dot{q} \tag{7}$$

which can be rewritten as:

$$K = \frac{1}{2}\dot{q}^T D(q)\dot{q} \tag{8}$$

by developing the previous formula, we obtain:

$$K = \frac{1}{2} \sum_{m,n} d_{m,n}(q) \dot{q}_m \dot{q}_n \tag{9}$$

where $d_{m,n}(q)$ is the m, n^{th} element of the matrix $D(q)$.

The gravity's torque is given by:

$$Q_j = \sum_i g m_i \frac{\partial G^0_{zi}}{\partial q_j}. \tag{10}$$

- G^0_{zi} is the z coordinate of the CoG of the i^{th} segment's computed in the base frame (R_0),
- g is the gravity acceleration.

Vector T (defined in (3)) is composed of the external forces' torque. For the robot presented here, there is no consideration of external forces, so the T vector only describes the motorized torques. Joints 1, 4, 7 and 8 are motorized, so the vector T is given by those four parameters. T_1 and T_4 are computed from the torques of motors 1 and 2 while T_7 and T_8 are deduced from motors 3 and 4.

The Euler-Lagrange equations can be written as:

$$\sum_m d_{jm}(q) \ddot{q}_m + \sum_{n,m} c_{nmj}(q) \dot{q}_n \dot{q}_m = Q_j + T_j \tag{11}$$

$$c_{nmj} = \frac{1}{2} [\frac{\partial d_{jm}}{\partial q_n} + \frac{\partial d_{jn}}{\partial q_m} - \frac{\partial d_{nm}}{\partial q_j}] \tag{12}$$

which is classically written as:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} = Q + T \tag{13}$$

where $D(q)$ represents the matrix of inertia and $C(q, \dot{q})$ the centrifuge-coriolis matrix where X_{jm}, the jm^{th} element of this matrix, is defined as:

$$X_{jm} = \sum_n c_{nmj} \dot{q}_n.$$

Finally, the J_{vi} and J_{wi} matrix considered in (5) and (6) have to be computed.

4.2 J_{vi} and J_{wi} Matrix Formulation

The matrix which links articular speed and general speed of a segment is computed from the linear and angular speeds formulas. The goal is to find a matrix for each segment. They are composed of 8 vectors (one for each joint of the model).

The computation consists in formulating in the base frame, the speed $(V_{P_i}(j-1, j)^{R_0})$ of a point P_i given by a motion of the joint q_j attached to the frame j according to the frame $j-1$. Those parameters can be deduced from the law of composition

speeds and the Denavitt Hartenberg (DH) formalism used for the geometric model [17]. Indeed, the general formulation is simplified by the geometric model. Only one degree of freedom (DoF) links two frames using the DH model and this DoF is a revolute or a prismatic joint. Moreover, the Z axis is always the rotation or translation axis, so the angular and linear speeds are given by four cases:

– The angular speed of a point for a revolute joint:

$$w_P(j-1,j)^{R_0} = R_{0,j} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{q_j}. \tag{14}$$

– The linear speed of a point for a revolute joint:

$$\begin{aligned} v_P(j-1,j)^{R_0} &= V_{O_j}^{R_{j-1}} + V_P^{R_j} + w_j \wedge O_j P^{R_j} \\ &= \quad \dot{q_j} R_{0,j} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \wedge R_{0,j} P_j. \end{aligned} \tag{15}$$

– The angular speed of a point for a prismatic joint:

$$w_P(j-1,j) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{16}$$

– The linear speed of a point for a prismatic joint:

$$v_P(j-1,j)^{R_0} = R_{0,j} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{q_j} \tag{17}$$

where P_j is the P point's coordinates in R_j.

Thus, the matrix of a segment i is formulated by computing speeds for each joints as:

$$\begin{bmatrix} v_i \\ w_i \end{bmatrix} = J(q)\dot{q} = [J_{1,i}(q), J_{2,i}(q), ...J_{8,i}(q)]\, \dot{q} \tag{18}$$

where $J_{j,i}(q)$ is a vector which links the speed of the i^{th} segment according to the j^{th} joint. The first segment is not affected by the motion of joints 7 and 8 while the second is not affected by joint 8, therefore $J_{7,1}(q)$, $J_{8,1}(q)$ and $J_{8,2}(q)$ are represented by a null vector.

5 Balance Criterion

The balance criterion used here are the ZMP (Zero Moment Point), widely used for the stability of humanoid robots and the Center of Gravity (CoG). Previous theoretical works and experiments have proved the ZMP efficiency [19]. It consists in keeping the

point on the ground at which the moment generated by the reaction forces has no component around x and y axis ([11] and [9]) in the support polygon of the robot. When the ZMP is at the border of the support polygon the robot is teetering. Unlike the ground projection of the center of gravity, it takes into account the robot's inertia.

The purpose of the following is to defined the coordinates of this point in any frame of the model according to the configuration of the robot. The definition can be implemented into the Newton equations to obtain those coordinates. In any point of the model: $M_0 = M_z + OZ \wedge R$ (M_0 and M_z define respectively the moment generated by the reaction force R at the points 0 and z).

According to the previous definition, there is no moment generated by reaction forces at the Zero Moment Point. Consequently, if Z defines the ZMP coordinates $M_0 = OZ \wedge R$. This formulation can be implemented into the Newton equations as:

$$\delta_0 = M_0 + \boldsymbol{OG} \wedge \boldsymbol{P} + \boldsymbol{OG} \wedge \boldsymbol{F_i} \tag{19}$$

where P is the gravity force, G is the robot's center of gravity and F_i is the inertial force (the first Newton's law gives $F_i = -m\ddot{G}$). According to the ZMP definition, the equation (19) can be formulated as:

$$\delta_0 = \boldsymbol{OZ} \wedge \boldsymbol{R} + \boldsymbol{OG} \wedge \boldsymbol{P} + \boldsymbol{OG} \wedge \boldsymbol{F_i} \tag{20}$$

$$\begin{cases} \delta_{0x} = Z_y R_z + G_y P_z - G_z P_y - G_z F i_y \\ \delta_{0y} = -Z_x R_z + G_z P_x - G_x P_z \end{cases} \tag{21}$$

$$\begin{cases} Z_y = \frac{\delta_{0x} - G_y P_z + G_z P_y + G_z F i_y}{R_z} \\ Z_x = \frac{-\delta_{0y} + G_z P_x - G_x P_z}{R_z}. \end{cases} \tag{22}$$

Also, it is possible to compute the position of the ZMP as a function of q (δ_0 depends on the matrix $D(q)$).

Assuming the ground knowledge, the ZMP computation gives a criterion to determinate the stability of the platform.

6 Results

This section presents the numerical computation of the criterion in the case of the clearance of a staircase (staircase set of 15 cm risers and 28 cm runs) with an average speed of 0.13 $m.s^{-1}$. The robot is equipped with a 2-axis inclination sensor that provides rolling and pitching. Vector q entries are measured using encoders on each actuated axis of the robot. Data have been stored during the experiments and the models (CoG and ZMP) have been computed off-line. This computation does not take into account the tracks' weight which is negligible in regard with the robot's weight. Fig. 6 presents the evolution of the ZMP (left) and the difference between those two criterion (right) during all the clearance. P_1 P_2 and P_3 represents the z-coordinates in the frame R_5 (Fig. 5) of three points of the robot which localization are noticed on Fig. 4.

This experiment allows us to validate the presented model and confirms the computation of the ZMP criterion. However, as it is shown on Fig. 6, the average difference

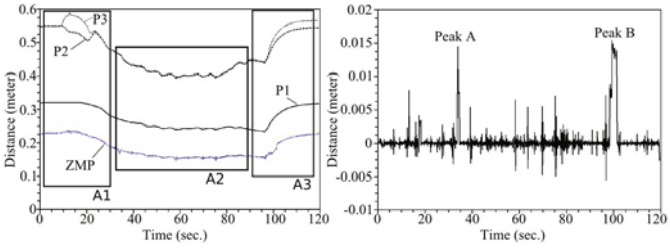


Fig. 6. Experiment's results. The left chart represents the evolution of the ZMP and the right one, the difference between CoG and ZMP.

between the ZMP and the COG is insignificant (about 0.21%). Moreover, the two peaks (A and B) on the Fig. 6 are not due to the dynamics of the system but to measurement errors. As the acceleration is measured with the encoders (linked to the motor shaft), when the tracks slip, the measurement is erroneous. The ZMP is computationally more expensive, needs more sensor measurements and the difference with the CoG is negligible. For these reasons, we conclude that the CoG seems well suited for this kind of experiments. Anyway, in the case where fast obstacle clearance may be necessary, the CoG may not longer be considered and the ZMP must be used instead.

7 Autonomous Stair Climbing Controller

The purpose of the following is to present a reactive autonomous staircase climbing controller. This study was performed through a home made C++ software which simulates the behavior of a VGTV in staircase climbing situations. The simulator is based on the model of our prototype B2P2. Giving the reduced impact of dynamics effects on the robot as explained previously, the CoG is considered as balance criterion in the simulations. The system presented here consists in controlling the elevation of the front part (e.g. the shape of the robot and implicitly the position of CoG).Currently, the tension of the tracks is controlled by a man-programmed algorithm that automatically adapts the shape of the robot according to ground.

7.1 Entries of the System

The system has to be able to react differently in regards with the climbing stage. So, entries have to differ according to the stage (first step, middle steps, final step). The chassis inclination measurement allows to check out the first stage (if the ground is plane) but does not help to conclude about the two others stages (Fig. 7). On the other hand, a distance sensor could check out the last stage as it is shown on Fig. 7. So considering that the robot is always parallel to the step in front of it, the inclination and distance sensors could be sufficient to achieve an autonomous staircase climbing.

7.2 Controller

As explained by [16] genetic algorithms can be used to train and optimize control systems. Typically, Artificial Neural Networks (ANNs) are often used as control systems

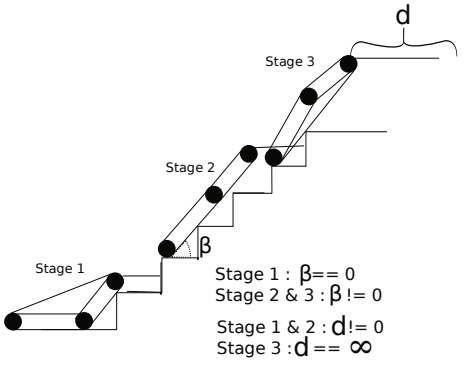


Fig. 7. Stage decomposition during climbing

for obstacle avoidance or grasping tasks ([14]) because such controller can approximate a wide range of mathematical functions. An ANN is composed of several units linked by weighted connections w_{ij}. Each unit i has entries x_j and one output y_i which is a function $\sigma()$ of $\sum_j^N (w_{ij}.x_j)$ where w_{ij} corresponds to the weight of the link between two neurones. Theoretically, neurones are organized in layers to perform a neural network. This network can be feed-forward (signal travel from inputs units forward to output units) or recurrent (there may be feedback connections from neuron in upper layer or in the same layer). As introduced, in the case of autonomous staircase climbing with a VGSTV as B2P2, the system have to control the elevation of the front part according to the position of the robot in the staircase. Otherwise, the output of the network is the elevation of the front part. Consequently the neural network architecture chosen is a feed-forward network with one hidden layer (Fig. 8) that must be addressed to approximate non linear functions. Indeed, a recurrent network does not seem useful, because a variation of the output (elevation angle) includes a variation of the network entries (IR distance sensors and inclination sensor).

7.3 Evolutionnary Training

In the controller, all the parameters are known except the 15 synaptic weights (w_i) which are deduced by an evolutionary algorithm based on a classical genetic approach. As the structure of the network has been fixed, only those weights have to be optimized. Consequently, a chromosome is only composed of the weights W_i.

Table 1. Chromosome description

w_1	w_2	...	w_i	...	w_{14}	w_{15}

Selective Reproduction. Each generation is composed of 200 sets of chromosomes composed of fifteen parameters randomly selected into $[-1, 1]$. After each generation, a selective reproduction is performed in order to compute the next generation. Here this

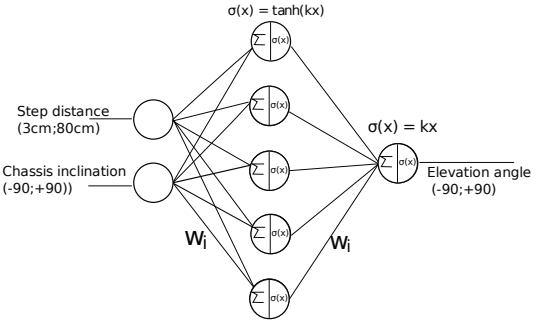


Fig. 8. Neural Network Model

selection is roulette wheel based that allow the best individual to be statistically selected more frequently. Otherwise:

$$p_i = \frac{f_i}{\sum_i^N f_i}$$

Where p_i is the probability of selection for the individual i, f_i stands for the fitness of the individual i and N represents the number of individual in the previous generation.

A selection process is performed for each chromosome element (e. g. fifteen times) to compute the next generation. However, the best individual of each generation is duplicated for the next generation without selective reproduction. Additionally, a mutation process is performed by replacing by a random value in the range $[-1, 1]$ a randomly selected gene (chromosome element) for 10% of the new generation to prevent premature convergence.

Fitness Function. As the goal is to climb staircases, the fitness function must be linked to the number of steps cleared. As this parameter is discrete, we combined it with another parameter that minimizes the energy (directly linked to the elevation part movement). Moreover, minimizing the energy provides smooth trajectory of the robot. It is expressed as follow:

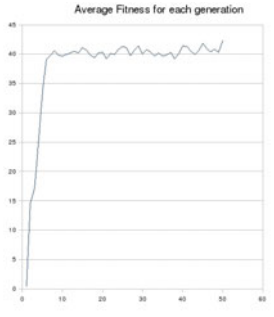
$$f_i = \frac{N_{steps}}{E}$$

Where f_i is the fitness of the individual i, N_{steps} stands for the number of steps cleared by individual i and E represents parameter depending on the average speed of the front part (V) as:

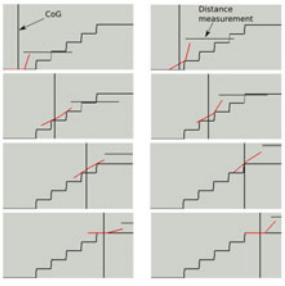
$$E = \begin{cases} 1 & \text{if } V < \text{threshold} \\ V & \text{otherwise} \end{cases}$$

7.4 Results

The simulation was performed with a classical staircase sets of five steps of 15 cm risers and 25 cm runs. Fifty generations was tested in order to have a characteristic convergence (Fig. 9(a)). Simulation results are shown on Fig. 9(b).



(a) Generation convergence



(b) Staircase climbing simulation results

Step length (cm) \ Step height (cm)	15	20	25	30
15	100%	100%	100%	100%
20	60%	80%	100%	100%

(c) Controller performance table

Fig. 9. Results

In order to evaluate how generic are the produced results, the best individual has to get over several kind of staircases as described in table 9(c). The percentage of cleared steps is indicated according to the length and the height of the steps. Note that the 25 cm is the maximum step height that the prototype is able to clear in tele-operated staircase clearance.

The controller seems to be able to perform an autonomous staircase clearing in regard with the quality of information given by the environment.

8 Conclusions

This paper presents a way to climb staircases autonomously with classical VGTVs by using their ability to modify their geometry in order to adapt themselves to the ground. It consists in a neural network which compute the elevation of the robot's front part in respect to its inclination and the distance between it and the next step. However, it was necessary to study the behavior of our robot in order to validate robot's model in context of a staircase clearance. Those simulation results have been implemented on our prototype B2P2 ; A video of autonomous staircase climbing can be found at http://www.istia.univ-angers.fr/LISA/B2P2/b2p2.html. Some improvement should be done to increase the reliability but those first result seems promising. Our future works focuses on a survey about control of VGSTV equipped with n-degrees of freedom using interval analysis in order to maximize the adaptability of such robots.

References

1. Carlson, J., Murphy, R.R.: Reliability analysis of mobile robots. International Conferenee on Robotics and Automation (September 2003)
2. Carlson, J., Murphy, R.R.: How ugvs physically fail in the field. IEEE Transactions on Robotics 21(3), 423–437 (2005)
3. Casper, J., Murphy, R.R.: Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. IEEE Transactions on Systems, Man, and Cybernetics 33(3), 367–384 (2003)

4. Clement, G., Villedieu, E.: Variable geometry track vehicle. Us patent (1987)
5. Craig, J.J.: Introduction to Robotics Mechanics and control, 2nd edn. Silma (1989)
6. Frost, T., Norman, C., Pratt, S., Yamauchi, B.: Derived performance metrics and measurements compared to field experience for the packbot. In: Proceedings of the 2002 PerMIS Workshop (2002)
7. Guarnieri, M., Debenest, P., Inoh, T., Fukushima, E., Hirose, S.: Development of helios vii: an arm-equipped tracked vehicle for search and rescue operations. IEEE/RSJ Int. Conference on Intelligent Robots (2004)
8. Iwamoto, T., Yamamoto, H.: Variable configuration track laying vehicle, us patent (March 1983)
9. 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: Proceedings of the 2002 IEEE Internationnal Conference on Robotics ans Automation (September 2003)
10. Khalil, W., Dombre, E.: Modeling, Identification and Control of Robots, 2nd edn. Kogan Page Science (2004)
11. Kim, J., Chung, W.K., Youm, Y., Lee, B.H.: Real time zmp compensation method using null motion for mobile manipulators. In: Proceedings of the 2002 IEEE Internationnal Conference on Robotics ans Automation (May 2002)
12. Kinugasa, T., Otani, Y., Haji, T., Yoshida, K., Osuka, K., Amano, H.: A proposal of flexible mono-tread mobile track. In: Internationnal Conference on Intelligent Robots and Systems (September 2008)
13. Sung Kyun, L., Dong Il, P., Yoon Keun, K., Byung-Soo, K., Sang-Won, J.: Variable geometry single-tracked mechanism for a rescue robot. In: IEEE International Workshop on Safety, Security and Rescue Robotics, pp. 111–115 (2005)
14. Lucidarme, P.: Evolution of a mobile robot's neurocontroller on the grasping task. In: Internationnal Conference on Informatics in Control, Automation and Robotics (2008)
15. Misawa, R.: Stair-climbing crawler transporter. Us patent (1997)
16. Nolfi, S., Floreano, D.: Evolutionnary Robotics. MIT Press, Cambridge (2000)
17. Paillat, J.L., Lucidarme, P., Hardouin, L.: Variable geometry tracked vehicle, description, model and behavior. In: Conference Mecatronics (May 2008)
18. Vincent, I., Trentini, M.: Shape-shifting tracked robotic vehicle for complex terrain navigation: Characteristics and architecture. In: Technical memorandum, Defence R and D Canada (December 2007)
19. Vukobratovic, M., Borovac, B.: Zero-momment point - thirty five years of its life. Internationnal journal of humanoid Robotics 1(1), 157–173 (2004)

Dynamic Walking on Uneven Terrains with Passivity-Based Bipedal Robots

Qining Wang[1,2], Yan Huang[1,2], Jinying Zhu[1,2], Baojun Chen[1,2], and Long Wang[1]

[1] Intelligent Control Laboratory, College of Engineering
Peking University, Beijing 100871, China
[2] Laboratory of Intelligent Systems and Human Engineering, College of Engineering
Peking University, Beijing 100871, China

Abstract. In this paper, we present an approach for passivity-based bipedal robots to achieve stable dynamic walking on uneven terrain. A powered two-dimensional seven-link walking model with flat feet and compliant ankles has been proposed to analyze and simulate the walking dynamics. We further describe a Particle Swarm Optimization based method, which uses optimized hip actuation and ankle compliance as control parameters of bipedal walking. Satisfactory results of simulations and real robot experiments show that the passivity-based walker can achieve stable bipedal walking with larger ground disturbance by the proposed method in view of stability and efficiency.

Keywords: Passive dynamic walking, Bipedal robots, Uneven terrains, Stability, Modeling.

1 Introduction

Stability guaranteed dynamic bipedal walking is one of the keys but also one of the more challenging components of humanoid robot design. Several actively controlled bipedal robots are able to deal with such dynamic stability when walk on irregular surface [22], step over obstacles [6] and climb stairs [11]. However, to increase the autonomy of the robot, the locomotion efficiency, which is far from that of human motion, has to be improved.

As one of the possible explanations for the efficiency of the human gait, passive dynamic walking [10] showed that a mechanism with two legs can be constructed so as to descend a gentle slope with no actuation and no active control. Several studies reported that these kinds of walking machines work with reasonable stability over a range of slopes (e.g. [10], [1]) and on level ground with kinds of actuation added (e.g. [2], [21]) show a remarkable resemblance to the human gait. In spite of having high energetic efficiency, passivity-based walkers have limits to achieve adaptive locomotion on rough terrain, which is one of the most important advantages of the legged robots. In addition, these walkers are sensitive to the initial conditions of walking.

To overcome such disadvantages, several studies proposed quasi-passive dynamic walking methods, which implement simple control schemes and actuators supplementarily to handle ground disturbances. For example, [13] described a reinforcement learning based controller and showed that the walker with such controller can maximally

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 187–199.
springerlink.com

overcome $-10mm$ ground disturbance. [15] presented a robot with a minimal number of degrees of freedom which is still capable of stable dynamic walking even on level ground and even up a small slope. [17] designed a learning controller for a two-dimensional biped model with two rigid legs and curved feet to walk on uneven surface that monotonically increases from $12mm$ to $40mm$ with a $2mm$ interval. In these studies, passivity-based walkers are often modeled with point feet or round feet. And the control parameter only includes hip actuation. None of them analyzed the stability or adaptability in quasi-passive dynamic walking with flat feet and ankle joint compliance which is more close to human motion. In fact, flat feet can offer the advantage of distributing the energy loss per step over two collisions, at the heel and at the toe [7], [12]. Moreover, experiments on human subjects and robot prototypes revealed that the tendon of the muscle in ankle joint is one mechanism that favors locomotor economy [4], [18]. It is predictable that by controlling and optimizing the hip actuation and ankle compliance, the passivity-based bipedal walker may achieve adaptive bipedal locomotion with larger disturbance on uneven terrain.

In this paper, we investigate how to control passivity-based bipedal walkers to achieve stable walking with terrain adaptability. A powered two-dimensional seven-link walking model with flat feet and ankle compliance has been proposed to analyze and simulate the walking dynamics. We hypothesized that the nervous system that controls human locomotion may use optimized hip actuation and joint compliance to achieve stable bipedal walking on irregular terrain. Actually, hip actuation will cause the walker to change locomotive patterns. Furthermore, we use both simulations and real robot prototype to explore the performance of the passivity-based walkers when walk on uneven terrain by utilizing a biologically inspired optimization based controller, which is adaptive and capable of selecting proper hip actuation and ankle compliance in view of walking stability and efficiency.

This paper is organized as follows. Section 2 describes the walking dynamics of the biped with flat feet and ankle compliance. Section 3 presents the optimization based walking control method. Experimental results are shown in section 4. We conclude in Section 5.

2 Passivity-Based Bipedal Walker

Our model extended the Simplest Walking Model [5] with the addition of hip joint actuation, upper body, flat feet and linear spring based compliance on ankle joints, aiming for adaptive bipedal locomotion with optimization based controller. The model consists of an upper body (point mass added stick) that rotates around the hip joint, a point mass representing the pelvis, two legs with knee joints and ankle joints, and two mass added flat feet (see Fig. 1).

The mass of each leg is simplified as point masses added on the Center of Mass (CoM) of the shank and the thigh respectively. Similar to [19], a kinematic coupling has been used in the proposed model to keep the body midway between the two legs. In addition, our model added compliance in ankle joints. Specifically, the ankle joints are modeled as passive joints that are constrained by linear springs. The model is implemented in MATLAB, using the parameter values shown in Table 1, which are derived from the mechanical prototype.

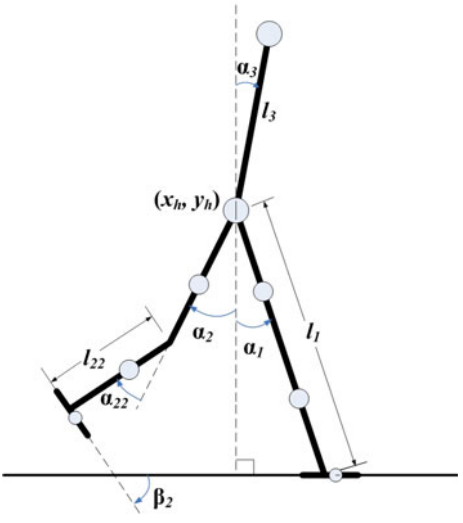


Fig. 1. Two-dimensional seven-link passive dynamic walking model. Mechanical energy consumption of level ground walking is compensated by applying a hip torque. The global coordinates of the hip joint is notated as (x_h, y_h). α_1 and α_2 are the angles between each leg and the vertical axis in sagittal plane respectively. The knee joints and ankle joints are all passive joints. To simulate human ankle compliance, the ankle joints of the model are constrained by linear springs.

The passive walker travels forward on level ground. The stance leg keeps contact with the ground while the swing leg pivots about the constraint hip. To compensate the mechanical energy consumption, similar to [8], we added a hip torque P between the swing leg and the stance leg. When the flat foot strikes the ground, there are two impulses, "heel-strike" and "foot-strike", representative of the initial impact of the heel and the following impact as the whole foot contacts the ground.

The walking model can be defined by the rectangular coordinates x which can be described by the x-coordinates and y-coordinates of the mass points and the corresponding angles. The walker can also be described by the generalized coordinates q. The springs on the ankles constrain the foot vertical to the shank when no heel-strike or foot-strike has occurred.

2.1 Constraint Equations

The constraint equations ξ_1 and ξ_2 used to detect heel contact with ground are defined as follows:

$$\begin{aligned} \xi_1 &= \begin{bmatrix} x_h + l_1 \sin\alpha_1 - l_{f1}\cos\alpha_1 - x_{f1} \\ y_h - l_1 \cos\alpha_1 - l_{f1}\sin\alpha_1 \end{bmatrix} \\ \xi_2 &= \begin{bmatrix} x_h + l_2 \sin\alpha_2 - l_{f2}\cos\alpha_2 - x_{f2} \\ y_h - l_2 \cos\alpha_2 - l_{f2}\sin\alpha_2 \end{bmatrix} \end{aligned} \tag{1}$$

Table 1. Default Parameter Values for the dynamic walking model and the following simulations

Parameter	Description	Value
m_1, m_2	leg mass	$1.12kg$
m_3	upper body mass	$0.81kg$
m_4	hip mass	$15.03kg$
m_{1t}, m_{2t}	thigh mass	$0.56kg$
m_{1s}, m_{2s}	shank mass	$0.56kg$
m_{f1}, m_{f2}	foot mass	$2.05kg$
l_1, l_2	leg length	$0.7m$
l_{11}, l_{22}	shank length	$0.35m$
l_{f1}, l_{f2}	foot length	$0.15m$
l_b	upper body length	$0.5m$
l_w	body width	$0.15m$
c_b	CoM of upper body	$0.2m$
c_1, c_2	CoM of leg	$0.2m$
c_{11}, c_{22}	CoM of shank	$0.2m$
k	ankle stiffness	$8.65Nm/rad$
P	hip torque	$0.38Nm$

where x_{f1} and x_{f1} are the global x-coordinates of the latest strike. When the flat foot completely contacts the ground, the constraint equations ξ_3 and ξ_4 used to maintain foot contact with ground are defined as follows:

$$\begin{aligned}\xi_3 &= \begin{bmatrix} x_h + l_1 \sin\alpha_1 - x_{f1} \\ y_h - l_1 \cos\alpha_1 \end{bmatrix} \\ \xi_4 &= \begin{bmatrix} x_h + l_2 \sin\alpha_2 - x_{f2} \\ y_h - l_2 \cos\alpha_2 \end{bmatrix}\end{aligned} \tag{2}$$

If only the heel contacts the ground, the constraint equations ξ_5 and ξ_6 during the period before foot-strike are defined as follows:

$$\begin{aligned}\xi_5 &= (x_h + l_1 \sin\alpha_1 - x_{f1})^2 + (y_h - l_1 \cos\alpha_1)^2 - l_{f1}^2 \\ \xi_6 &= (x_h + l_2 \sin\alpha_2 - x_2)^2 + (y_h - l_2 \cos\alpha_2)^2 - l_{f2}^2\end{aligned} \tag{3}$$

Note that ξ_5 and ξ_6 are similar to constraining the ankle joint that connects shank and foot to move in a circular orbit with heel as the center and distance between heel and ankle joint as the radius.

Similar to [19], a reduced mass matrix M_r is introduced, which is defined as follows:

$$[M_r] = [T]^T [M_g] [T] \tag{4}$$

where the Jacobian $T = \frac{\partial x}{\partial q}$. Here x is the global coordinates of the six pointmasses (stance shank with foot, swing shank with foot, hip, stance thigh, swing thigh, body), while M_g is the mass matrix in global coordinates. Matrix Ξ_i transfers the independent generalized coordinates q into the constraint equation ξ_i, where $i = 1, 2, \ldots, 6$

$$\Xi_i = \frac{d\xi_i}{dq} \tag{5}$$

Consequently, matrix $\tilde{\Xi}_i$ is defined as follows:

$$\tilde{\Xi}_i = \frac{\partial(\Xi_i \dot{q})}{\partial q}\dot{q} \tag{6}$$

2.2 Single Support Phase

Suppose that leg 1 (l_1) is the stance leg, while leg 2 (l_2) is the swing leg. In the beginning of the single support phase, the knee joint is locked (keep the shank and the thigh in a straight line). The Equation of Motion (EoM) is described as follows:

$$\begin{bmatrix} M_r & \Xi_3^T \\ \Xi_3 & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ F_c \end{bmatrix} = \begin{bmatrix} F_r \\ -\tilde{\Xi}_3 \end{bmatrix} \tag{7}$$

where F_r is the external force, while F_c is the foot contact force. Here, the external force F_r is used to compensate the mechanical energy consumption of level ground walking, which defined as follows:

$$\{F_r\} = [T]^T \left(\{F\} - [M_g]\{\ddot{x}\}\right)$$

where F is the external force in global coordinates, including gravity, hip actuation, and torque in compliant ankle joints. Then when the swing leg is swung forward, the knee joint releases the shank.

2.3 Heel-Strike Phase

In this phase, leg 1 (l_1) is still the stance leg, while leg 2 (l_2) is the swing leg. The heel contacts the ground (heel-strike occurs). The EoM of the model changes to:

$$\begin{bmatrix} M_r & \Xi_6^T \\ \Xi_6 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ F_c \end{bmatrix} = \begin{bmatrix} M_r \dot{q}^- \\ -e\Xi_6 \dot{q}^- \end{bmatrix} \tag{8}$$

After the heel-strike, the foot rotates around the ankle joint. The EoM of the model is:

$$\begin{bmatrix} M_r & \Xi_3^T & \Xi_6^T \\ \Xi_3 & 0 & 0 \\ \Xi_6 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ F_{c1} \\ F_{c2} \end{bmatrix} = \begin{bmatrix} F_r \\ -\tilde{\Xi}_3 \\ -\tilde{\Xi}_6 \end{bmatrix} \tag{9}$$

Note that the constraint equations guarantee that the stance leg maintains foot contact with ground and the swing leg maintains heel contact with ground. In addition, the spring constant k in the compliant ankle joints should not be too big. Otherwise, the stance leg will lose contact with ground. In this phase, since the foot rotates around the ankle joint, the force generated by the springs on the swing leg should be considered as external force. Thus, in this event, the mass matrix M would not include the point mass of the swing foot.

2.4 Toe-Strike Impulse

The proposed walking model with flat feet introduces a toe-strike impulse in addition to the heel-strike collision. The EoM of the model is:

$$\begin{bmatrix} M_r & \Xi_4^T \\ \Xi_4 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ F_c \end{bmatrix} = \begin{bmatrix} M_r \dot{q}^- \\ -e\Xi_4 \dot{q}^- \end{bmatrix} \tag{10}$$

Note that in this phase, we consider that the ankle joint of the swing leg is constrained to move in a circular orbit with toe as the center and distance between toe and ankle joint as the radius. The force generated by the spring on the swing leg should be considered as external force, which can also be considered as the constraint force of the circular orbit. After the toe-strike, one step ends.

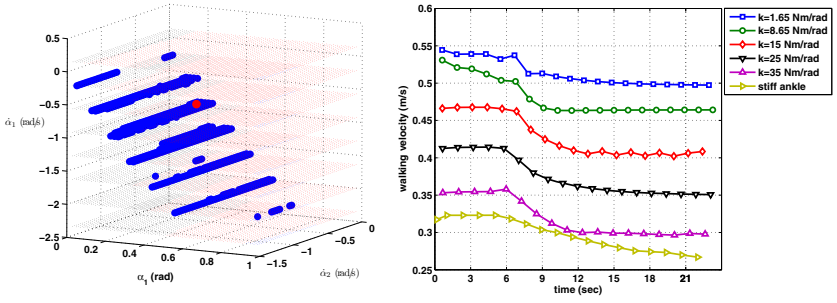
2.5 Effects of Hip Actuation and Ankle Compliance

By application of the cell mapping method that has been used in the several studies of passive dynamic walking (e.g. [19], [21]), we found that the model performs well in the concept of global stability. The allowable errors can be much larger than the results obtained in [19]. This can be inspected by the evaluation of the basin of attraction as shown in Fig. 2(a), which is the complete set of initial conditions that eventually result in cyclic walking motion. One can find that cyclic walking with initial conditions in Table 1, emerges even if the initial step is nearly fourfold as large, e.g. $\{\alpha_1(0) = 0.8, \dot{\alpha}_1(0) = -2.3, \dot{\alpha}_2(0) = -0.8\}$. It indicates that passive dynamic walking with flat feet and ankle compliance may play better on rough terrain with ground disturbance.

It has been examined that optimized hip actuation can improve the stability of the passive walker [9]. In addition, one can use the hip actuation as the control parameter to achieve stable walking on uneven terrain [13]-[16]. In our model mentioned above, ankle compliance k can also be used as a control parameter to affect the walking gait. As shown in Fig. 2(b), under the same change of hip actuation, different ankle compliance reveals different responses in walking velocity transition. It indicates that more compliance results in more visible sensitivity to the change of hip torque. According to the analysis of basins of attraction with different k [18], we find that a relatively small k will lead to more stable points. However, more compliance in ankle joints may result in often falling backward during walking. Thus, optimized ankle compliance may result in a more stable bipedal walking that allows larger disturbance.

3 Optimization Based Walking Control

In order to optimize the hip actuation and ankle compliance which affect walking gait as analyzed above, Particle Swarm Optimization (PSO) has been chosen with a focus lying on quickly finding suitable results, in view of time-consuming and adaptivity of the gait. In the realization of the PSO algorithm, a swarm of N particles is constructed inside a D-dimensional real valued solution space, where each position can be a potential solution for the optimization problem. The position of each particle is denoted as X_i $(0 < i < N)$. Each particle has a velocity parameter V_i $(0 < i < N)$. It specifies



(a) (b)

Fig. 2. (a) Basin of attraction of the passive dynamic walking model with flat feet and compliant ankles. The blue layers of points represent horizontal slices of a three-dimensional region of initial conditions that eventually result in the cyclic walking motion. The fixed point is indicated with a red point, which is above one of the sample slices. (b) Results of actively changing walking speed with the same hip actuation and different ankle compliance (k varies).

that the length and the direction of X_i should be modified during iteration. A fitness value attached to each location represents how well the location suits the optimization problem. The fitness value can be calculated by a fitness function of the optimization.

In this study, we used adaptive PSO with changing inertia weight. The update equation for velocity with inertial weight is described as follows:

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 r_{1d}^{k}(pbest_{id}^{k} - x_{id}^{k}) + c_2 r_{2d}^{k}(gbest_{d}^{k} - x_{id}^{k}) \tag{11}$$

where w is the inertia weight. v_{id}^{k} is one component of V_i (d donates the component number) at iteration k. Similarly, x_{id}^{k} is one component of X_i at iteration k. $pbest_i$ $(0 < i < N)$ and $gbest$ are the personal best position and the global best position at each iteration respectively. c_1 and c_2 are acceleration factors. r_1 and r_2 are random numbers uniformly distributed between 0 and 1. Note that each component of the velocity has new random numbers. In order to prevent particles from flying outside the searching space, the amplitude of the velocity is constrained inside a spectrum $[-v_d^{max}, +v_d^{max}]$.

3.1 Fitness Function and Optimization Process

For a specific passive walker, the mechanical parameters (length and mass distribution) are constant. To control passive dynamic walking on uneven terrain, we focus the control parameters on hip actuation P and ankle compliance k. Then the two-dimensional parameter space is $(P,\ k)$. Here a set of parameters stands for a particle of PSO. Since the walker will be optimized with integration of stability and efficiency, the fitness function is defined as follows:

$$\sigma = \sigma_s + \gamma\sigma_e \tag{12}$$

where σ_s and σ_e are the fitness value to assess the stability and efficiency of each set of parameters respectively. γ is the tuning factor to change the importance of the two characteristics.

There are several methods to evaluate the stability of the passive dynamic walking. In this study, the stability will be quantified by the modulus of the Jacobian matrix of the mapping function as defined in [19]. Here, we notate the maximal eigenvalue as λ_m, which represents the decreasing speed of the deviation. The stability grades varies for different sets of parameters even they all have a stable fixed point. The smaller the λ_m is, the faster the deviation decreases and the more stable the walker is. The similar conclusion can be obtained when all sets of parameters only have an unstable fixed point. The larger the λ_m is, the more far from the stable state the walker is. Then we define the fitness function of stability as the follows:

$$\sigma_s = \frac{1}{\lambda_m} \tag{13}$$

Similar to [2] and [19], the energetic efficiency of walking can be evaluated by the specific resistance as follows:

$$\Theta = \frac{E}{MgL} \tag{14}$$

where E is the cost of energy. In this study, the energy cost is generated only by the hip torque. M is the total mass of the model. g is the acceleration of gravity. L is the length of distance the robot passed. Then the fitness function can be defined as follows:

$$\sigma_e = \frac{1}{\Theta} = \frac{Mgl}{E} \tag{15}$$

From (12), (13) and (15), we can obtain the whole expression of the fitness function:

$$\sigma = \frac{1}{\lambda_m} + \gamma \frac{Mgl}{E} \tag{16}$$

Additionally, to evaluate the walking motion after overcoming ground disturbance, we introduce another expression of fitness function. We use further walking distance d instead of the fitness efficiency σ_e. Then the fitness function can be rewritten as:

$$\sigma = \frac{1}{\lambda_m} + \gamma d \tag{17}$$

3.2 Gait Controller with Optimized Parameters

After analyzing the effects of hip actuation and ankle compliance in the stability and adaptability of the passive dynamic walker, we select P and k as the gait control input. The output of the optimization simulator $P_o(t)$ and $k_o(t)$ are added to the current actual hip actuation $P_a(t)$ and ankle compliance $k_a(t)$ (see Fig. 3). This results in extra hip torque to move the swing leg more forward and prevent tripping. The purpose of the failure-detection block in Fig. 3 is to monitor in simulations the foot contact and the knee locking in order to detect whether walking failed. A failure means that the robot fell either backward or forward or that it started running (both feet leave contact with ground). There is a active control counter module in the diagram. It is used to count the times of applying active control (increasing or decreasing P or k) during one continuous walking. The output of this module make the simulator change P or k. With the

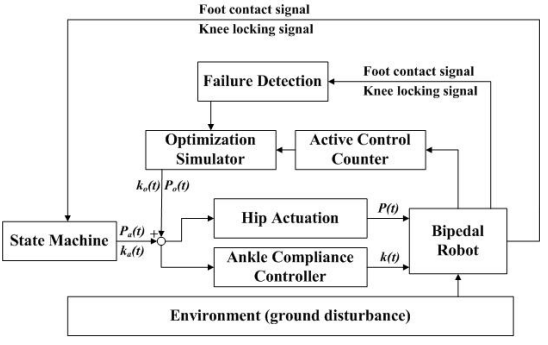


Fig. 3. The control scheme to overcome ground disturbance with optimized hip torque and ankle compliance

dynamic model that adequately describing the real robot, an adaptive optimizing control scheme can be done without manually set to teach the robot and without the robot damaging itself.

4 Experimental Results

All the simulation experiments used the dynamic model mentioned in Section II which was implemented in MATLAB, using the parameter values shown in Table 1. The numerical integration of the second order differential EoMs uses the Runge-Kutta method, which is similar to the simulation methods mentioned in [18,19].

4.1 Parameter Optimization

Based on the adaptive PSO with proper inertial weight mentioned above, we optimized the hip torque (hip actuation) and ankle compliance to achieve adaptive walking with maximal allowable ground disturbance of the model with parameter values in Table 1. The testing scenario is a floor with one step down. The height of the step is gradually chosen from the range from $1mm$ to $20mm$. The initial particles which represent the parameter set of P and k are randomly selected from the corresponding points in 2(a) that will finally achieve stable walking. During the walking simulation of the dynamic model, the ground disturbance gradually increase. The optimization process evaluates the maximum ground disturbance of the dynamic model with certain P and k. During the walking, the selected P and k keep constant to overcome gradually increased ground disturbance. The optimization finally record the maximum ground disturbance in each generation. The fitness function is (16). Fig. 12 shows the results. It is clear that by applying the adaptive PSO, the optimization process can quickly find the optimal parameter set. It also indicated that with optimized hip actuation and ankle torque, the passive dynamic walker can achieve stable walking with no active control even if there is $-11mm$ ground disturbance.

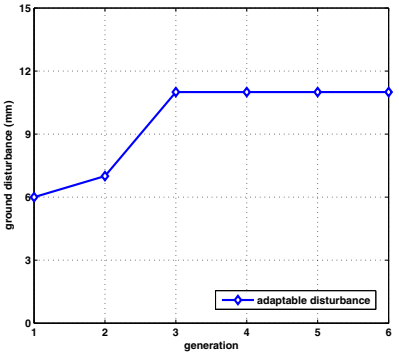
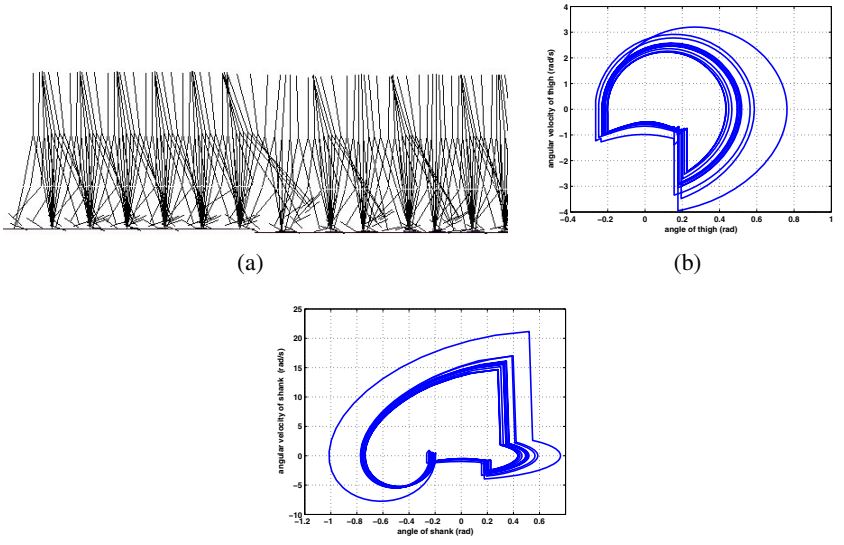


Fig. 4. Maximum ground disturbance with optimized hip torque and ankle compliance that keep unchanged during stable walking

4.2 Walking on Uneven Terrain with Control

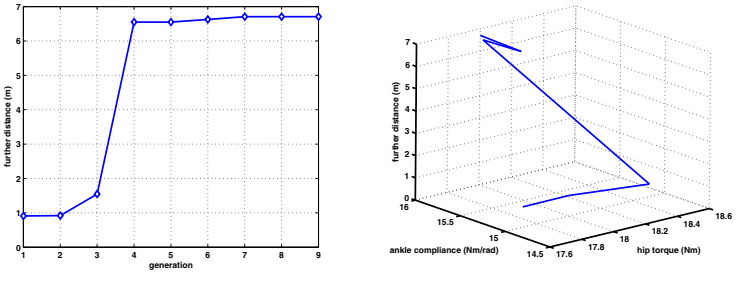
Starting from the optimized P and k with no active control during walking simulations, we add the control scheme shown in Fig. 3 to the walking motion. A preset ground disturbance occurs at known time during the level walking. The active control counter determines the times of active control to change P and k. In this simulation, the ground disturbance varies from $-15mm$ to $-25mm$. The fitness function of the optimization is (17). The counter first makes the optimization simulator to actively change P and k once when ground disturbance occurs. There was no optimized set of P and k that can overcome the $-25mm$ step. Then the counter makes the simulator to actively change P and k twice when ground disturbance occurs. The walker successfully achieved stable walking with $-25mm$ disturbance (see stick diagram shown in Fig. 5(a)). Fig. 5(b) and (c) show the trajectories of hip and knee during the adaptive walking. Note that the cyclic walking was initially actuated by a relatively small hip torque. After one time of active control (varying P and k), the hip torque increased to move the swing leg more forward and prevent tripping. Since there was a second time of active control, the trajectories of the swing thigh and the swing shank transited to bigger limit cycles. Such optimized P and k finally stabilized the walking motion after a step down occurred.

Fig. 6 shows the optimization process of the hip actuation and ankle compliance during the walking with two times of active control. We set that if the walker can walk stably for enough time after the ground disturbance, the walking motion is adaptive on the uneven ground. Specifically, the distance is the product of walking speed times 25 seconds. Fig. 6(a) demonstrates the results of further distance after ground disturbance each generation. After four generations, the walker can find optimized parameters to overcome the step. Fig. 6(b) shows the process of selecting P and k during the optimization. The initial hip torque is $5.5130Nm$ and ankle compliance is $12.2218Nm/rad$. The two times of variance of P and k are ($18.0000Nm$, $16.0000Nm/rad$ and ($5.5000Nm$, $15.4228Nm/rad$) respectively. Though there is no complex learning algorithm in the control scheme, the walker can perform better terrain adaptability comparing with other simulation results (e.g. [13], [17]).



(a) (b) (c)

Fig. 5. Adaptive locomotion with active control on uneven terrain. This result is obtained every 10 frames during a continuous walking. (a) is the stick diagram. (b) and (c) are the angular trajectories of the thigh and shank respectively.



(a) (b)

Fig. 6. Optimization of hip actuation and ankle compliance. Both P and k vary during the continuous walking. (a) is the results of further distance after ground disturbance each generation. (b) is the results of selecting hip actuation and ankle compliance during optimization.

4.3 Real Robot Experiments

To study natural and energy-efficient bipedal walking, we designed and constructed a bipedal robot prototype , $1.2m$ in height and $20kg$ in weight. With the bisecting hip mechanism similar to [21], the prototype has five Degrees of Freedom (DOFs). Two commercial motors are used in the hip joints to perform hip actuation. Each leg consists of a thigh and a shank interconnected through a passive knee joint that has a locking mechanism. Springs are installed between the foot and the plate that is pushing the leg up while it is rotated around the ankle. To prevent foot-scuffing, we modified the foot

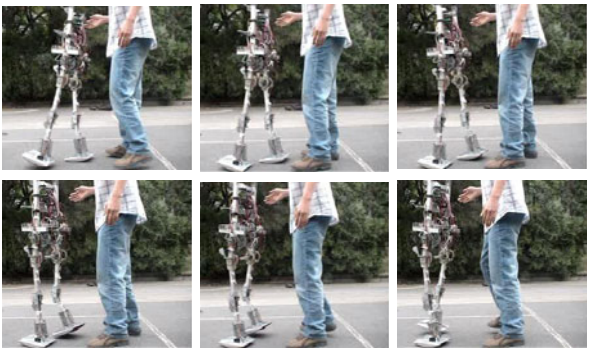


Fig. 7. A sequence of photos captured during autonomous walking of the robot prototype on natural ground

with arc in the front and the back-end. Specific mechanical parameters are shown in Table 1. In the experiment, by using the proposed method, the robot tried to walk on natural ground outdoor. The natural ground is not a strict continuous level floor, where random irregularity of the ground and slight slippery occurred.

Fig. 7 shows the result. The robot can achieve three-dimensional stable walking on natural ground with more than 10 steps. Comparing to the results of terrain adaptability of other real robot experiments (e.g. two-dimensional walkers [20], [16]), the successful three-dimensional walking of the robot prototype shows that the quasi-passive dynamic walker with optimized hip actuation and ankle compliance can perform stable walking with larger ground disturbance.

5 Conclusions

In this paper, we have investigated how to control passivity-based bipedal walkers to achieve stable locomotion with terrain adaptability. Satisfactory results of simulations and real robot experiments indicated that having the fixed mechanical parameters during walking, the passivity-based walker can walk on uneven terrain with larger ground disturbance by optimized hip actuation and ankle compliance in view of walking stability and efficiency. In the future, more real robot experiments will be continued to overcome more complex ground disturbance.

Acknowledgements. The authors would like to thank M. Wisse for sharing the simulation files of the simplest walking model. Dongjiao Lv contributed in the stick diagrams that visualize walking of the model. This work was supported by the 985 Project of Peking University (No. 3J0865600).

References

1. Collins, S., Wisse, M., Ruina, A.: A three-dimensional passive-dynamic walking robot with two legs and knees. Int. J. Robot. Res. 20, 607–615 (2001)
2. Collins, S., Ruina, A., Tedrake, R., Wisse, M.: Efficient bipedal robots based on passive-dynamic walkers. Science 307, 1082–1085 (2005)

3. Eberhart, R., Kennedy, J.: Particle swarm optimization. In: Proceedings of the IEEE Conference on Neural Network, pp. 1942–1948 (1995)
4. Fukunaga, T., Kubo, K., Kawakami, Y., Fukashiro, S., Kanehisa, H., Maganaris, C.N.: In vivo behaviour of human muscle tendon during walking. Proc. Biol. Sci. 268, 229–233 (2001)
5. Garcia, M., Chatterjee, A., Ruina, A., Coleman, M.: The simplest walking model: stability, complexity, and scaling. ASME J. Biomech. Eng. 120, 281–288 (1998)
6. Guan, Y., Neo, E.S., Yokoi, K., Tanie, K.: Stepping over obstacles with humanoid robots. IEEE Trans. on Robot. 22(5), 958–973 (2006)
7. Kwan, M., Hubbard, M.: Optimal foot shape for a passive dynamic biped. J. Theor. Biol. 248, 331–339 (2007)
8. Kuo, A.D.: Energetics of actively powered locomotion using the simplest walking model. ASME J. Biomech. Eng. 124, 113–120 (2002)
9. Kurz, M.J., Stergiou, N.: An artificial neural network that utilizes hip joint actuations to control bifurcations and chaos in a passive dynamic bipedal walking model. Biol. Cybern. 93, 213–221 (2005)
10. McGeer, T.: Passive dynamic walking. Int. J. Robot. Res. 9, 68–82 (1990)
11. Michel, P., Chestnutt, J., Kagami, S., Nishiwaki, K., Kuffner, J., Kanade, T.: GPU-accelerated real-time 3D tracking for humanoid locomotion and stair climbing. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 463–469 (2007)
12. Ruina, A., Bertram, J.E.A., Srinivasan, M.: A collisional model of the energetic cost of support work qualitatively explains leg sequencing in walking and galloping, pseudo-elastic leg behavior in running and the walk-to-run transition. J. Theor. Biol. 237(2), 170–192 (2005)
13. Schuitema, E., Hobbelen, D., Jonker, P., Wisse, M., Karssen, J.: Using a controller based on reinforcement learning for a passive dynamic walking robot. In: Proceedings of the IEEE-RAS International Conference on Humanoid Robots, pp. 232–237 (2005)
14. Smith, R., Rattanaprasert, U., O'Dwyer, N.: Coordination of the ankle joint complex during walking. Hum. Movement Sci. 20, 447–460 (2001)
15. Tedrake, R., Zhang, T.W., Fong, M.F., Seung, H.S.: Actuating a Simple 3D Passive Dynamic Walker. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 4656–4661 (2004)
16. Ueno, T., Nakamura, Y., Takuma, T., Hosoda, K., Shibata, T., Ishii, S.: Fast and stable learning of quasi-passive dynamic walking by an unstable biped robot based on off-policy natural actor-critic. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5226–5231 (2006)
17. Wang, S., Braaksma, J., Babuška R., Hobbelen D.: Reinforcement learning control for biped robot walking on uneven surfaces. In: Proceedings of the International Joint Conference on Neural Networks, pp. 4173–4178 (2006)
18. Wang, Q., Huang, Y., Wang, L.: Passive dynamic walking with flat feet and ankle compliance. Robotica 28(3), 413–425 (2010)
19. Wisse, M., Schwab, A.L., van der Helm, F.C.T.: Passive dynamic walking model with upper body. Robotica 22, 681–688 (2004)
20. Wisse, M., Schwab, A.L., van der Linde, R.Q., van der Helm, F.C.T.: How to keep from falling forward: elementary swing leg action for passive dynamic walkers. IEEE Trans. Robot. 21(3), 393–401 (2005)
21. Wisse, M., Feliksdal, G., van Frankenhuyzen, J., Moyer, B.: Passive-based walking robot - Denise, a simple, efficient, and lightweight biped. IEEE Robot. Autom. Mag. 14(2), 52–62 (2007)
22. Yamaguchi, J., Takanishi, A.: Development of a leg part of a humanoid robot - development of a biped walking robot adapting to the humans' normal living floor. Autonomous Robots 4(4), 369–385 (1997)

Decentralized Robust Collision Avoidance Based on Receding Horizon Planning and Potential Field for Multi-Robots Systems

Michael Defoort[1], Arnaud Doniec[2], and Noury Bouraqadi[2]

[1] Univ Lille Nord de France, F-59000 Lille, CNRS, FRE 3304, F-59313 Valenciennes, UVHC LAMIH, F-59313 Valenciennes, France
michael.defoort@univ-valenciennes.fr

[2] Ecole de Mines de Douai, 941 rue Charles Bourseul, BP 10838, 59508 Douai, France
arnaud.doniec@ensm-douai.fr, noury.bouraqadi@ensm-douai.fr

Abstract. This paper deals with the decentralized navigation problem for multi-robots systems. The proposed planning scheme consists of two parallel processes. A decentralized receding horizon motion planner, in which each robot computes its planned trajectory, enables the coordination between cooperative robots. A fast navigation controller, based on artificial potential fields and sliding mode control technique, is used to simultaneously track the planned trajectory while avoiding collision with unexpected entities. Simulation studies are provided in order to show the effectiveness of the proposed approach.

Keywords: Collision avoidance, Decentralized path planning, Multi-robots system, Potential field, Sliding mode control.

1 Introduction

The research effort in multi-robots systems (MRS) relies on the fact that multiple robots have the possibility to perform a mission more efficiently than a single robot. Among all the topics of study in this field, the issue of conflict resolution becomes an increasingly important point. Many cooperative tasks such as surveillance, search, rescue or area data acquisition need the robots to autonomously navigate without collision.

Solving conflicts in MRS consists in introducing some coordination mechanisms in order to give a coherence between the robot acts [1]. For motion planning, three coordination mechanisms are identified:

- *the coordination by adjustment*, where each robot adapts its behavior to achieve a common objective [2]. However, most of the planning algorithms are centralized, which often limit their applicability in real systems.
- *the coordination by leadership (or supervision)* where a hierarchical relationship exists between robots [3]. Such an approach is easy to implement [4]. However, due to the lack of an explicit feedback from the followers to the leader, the collision avoidance cannot be guaranteed if followers are perturbed (during obstacle avoidance for instance). Another disadvantage is that the leader is a single point of failure.

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 201–215.
springerlink.com

– *the standardization* where procedures are predefined to solve some particular interaction cases [5]. While this approach can lead to straightforward proofs, it also tends to be less flexible with respect to changing conditions.

Here, the problem of interest is the decentralized navigation for autonomous robots through a coordination by adjustment. Each vehicle is modeled as an unicycle with a limited sensing range in order to capture the essential properties of a wide range of vehicles. They are dynamically decoupled but have common constraints that make some conflicts. Indeed, each robot has to avoid collision with the other entities. Furthermore, the proposed framework allows moving (and static) obstacles to be avoided since they can be modeled as non cooperative entities.

Motion planning consists in generating a collision-free trajectory from the initial to the final desired positions for a robot. Since the environment is partially known and further explored in real time, the computation of complete trajectories from start until finish must be avoided. Therefore, the trajectories have to be computed gradually over time while the mission unfolds. It can be accomplished using an online receding horizon planner [6], in which partial trajectories from an initial state toward the goal are computed by solving optimal control problems over a limited horizon [7].

Two strategies for motion planning in MRS are the centralized and decentralized (distributed) approaches. Although the centralized one has been used in different studies (see [8] for instance), its computation time which scales exponentially with the number of robots, its communication requirement and its lack of security make it prohibitive. To overcome these limitations, one can use a distributed strategy which results in behaviors closed to what is obtained with a centralized approach.

Recently, some decentralized algorithms are based on a receding horizon approach. In [9], a distributed solution is provided for the rigid formation stabilization problem. In [10], the decentralized receding horizon planner is solved using mixed-integer linear programming (MILP). Every robot is allotted a time slot to compute its own dynamically feasible collision-free trajectory. This scheme creates a queueing order of non-conflicting groups of vehicles, where each group updates their trajectory sequentially and communicates the solution to its neighbors. In [11], a decentralized algorithm based on a coordination by adjustment is proposed to solve the navigation problem for MRS. However, the large amount of information exchanged between robots and the addition of several constraints make this strategy prohibitive when the number of vehicles increases.

One of other collision avoidance algorithm is potential field method, where an artificial potential function treats each robot as a charged particle that repels all the other entities [12]. However, most of them are only based on relative position information and do not consider coordination between cooperative robots.

In this paper, we proposed a practical decentralized scheme, based on a coordination by adjustment, for real time navigation of large-scale MRS. As illustrated in Fig. 1, the scheme consists of two parallel processes:

– a distributed receding horizon planner, in which each robot computes its own planned trajectory locally, for the coordination between cooperative robots,

– a reactive approach, which combines artificial potential fields and sliding mode control technique, for simultaneously tracking the planned trajectory while avoiding collision with unexpected entities (i.e. non cooperative entities).

The main advantages of the proposed strategy, especially for large-scale MRS, are the small amount of information exchanged between cooperative robots and the robustness.

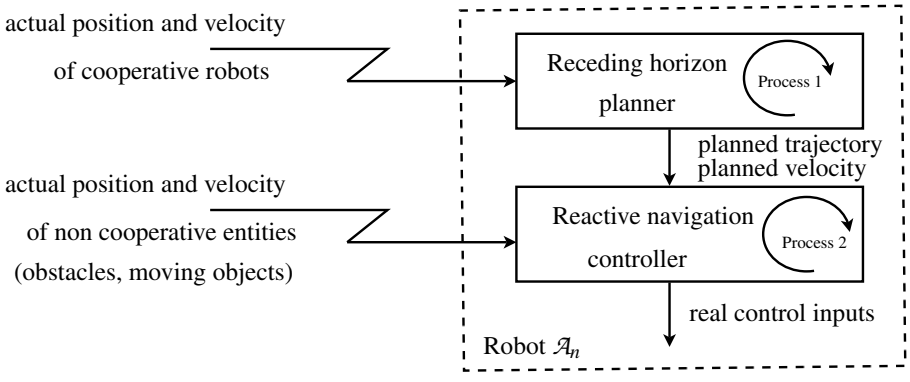


Fig. 1. Proposed navigation algorithm

The outline of this paper is as follows. In Section 2, the problem setup is described. In Section 3, the navigation algorithm is presented. Finally, numerical results illustrate the effectiveness of the strategy.

2 Problem Statement

2.1 Dynamic Model of the Robots

Each robot $\mathcal{A}_n$ ($n \in (1, \ldots, N)$ with $N \in \mathbb{N}$), shown in Fig. 2, is of unicycle-type. Its two fixed driving wheels of radius r_n, separated by $2\rho_n$, are independently controlled by two actuators (DC motors) and the passive wheel prevents the robot from tipping over as it moves on a plane. The centre of mass C_n, whose coordinates are (x_n, y_n), is located at the intersection of a straight line passing through the middle of the robot and the axis of the two driving wheels. Its configuration is given by:

$$\boldsymbol{\eta}_n = [x_n, y_n, \theta_n]^T$$

where (x_n, y_n) is the position of its mass center C_n and θ_n is its orientation in the global frame. The kinematics of the robot is shown under the nonholonomic constraints [13]. The pure rolling and nonslipping nonholonomic conditions are described by:

$$[-\sin\theta_n, \cos\theta_n, 0]\, \dot{\boldsymbol{\eta}}_n = 0$$

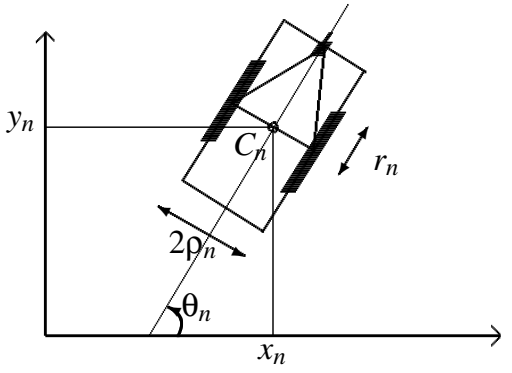


Fig. 2. Unicycle-type robot

The dynamic model of robot $\mathcal{A}_n$ is given as in [14]:

$$\dot{\boldsymbol{\eta}}_n = J(\boldsymbol{\eta}_n)\boldsymbol{z}_n \tag{1}$$

$$M_n\dot{\boldsymbol{z}}_n + D_n\boldsymbol{z}_n = \boldsymbol{\tau}_n \tag{2}$$

where

- M_n is a symmetric positive definite inertia matrix
- D_n is a symmetric damping matrix
- the transformation matrix $J(\boldsymbol{\eta}_n)$ is

$$J(\boldsymbol{\eta}_n) = \frac{r_n}{2}\begin{bmatrix}\cos\theta_n & \cos\theta_n \\ \sin\theta_n & \sin\theta_n \\ \rho_n^{-1} & -\rho_n^{-1}\end{bmatrix} \tag{3}$$

- $\boldsymbol{z}_n = [z_n^r, z_n^l]^T$ where z_n^r, z_n^l are the angular velocities of the right and left wheels. The relationship between $\boldsymbol{z}_n$ and the linear and angular velocities, denoted v_n, w_n, is

$$\begin{bmatrix}z_n^r \\ z_n^l\end{bmatrix} = B_n\begin{bmatrix}v_n \\ w_n\end{bmatrix} \text{ with } B_n = \frac{1}{r_n}\begin{bmatrix}1 & \rho_n \\ 1 & -\rho_n\end{bmatrix} \tag{4}$$

- $\boldsymbol{\tau}_n = [\tau_n^r, \tau_n^l]^T$ where τ_n^r,τ_n^l are the control torques applied to the wheels of the robot

Remark 1. *System (1)-(2) is flat (see [15] for details about flatness) since all system variables can be differentially parameterized by x_n, y_n as well as a finite number of their time derivatives. For instance, θ_n, v_n and w_n can be expressed as*

$$\theta_n = \arctan\frac{\dot{y}_n}{\dot{x}_n},\ v_n = \sqrt{\dot{x}_n^2 + \dot{y}_n^2},\ w_n = \frac{\ddot{y}_n\dot{x}_n - \ddot{x}_n\dot{y}_n}{\dot{x}_n^2 + \dot{y}_n^2}$$

Remark 2. *Speed $\boldsymbol{u}_n = [\dot{x}_n, \dot{y}_n]^T$ of $\mathcal{A}_n$ is restricted to lie in a closed interval $\mathcal{S}_n$*

$$\mathcal{S}_n = \left\{\, \boldsymbol{u}_n \in \mathbb{R}^2 \mid \|\boldsymbol{u}_n\| \leq u_{n,max} \,\right\} \tag{5}$$

2.2 Assumptions and Control Objective

Assumption 1. *The following assumptions are made:*

- $\mathcal{A}_n$ *knows its position* $\boldsymbol{p}_n = [x_n, y_n]^T$ *and its velocity* $\boldsymbol{u}_n = [\dot{x}_n, \dot{y}_n]^T$
- $\mathcal{A}_n$ *has a physical safety area, which is centered at* C_n *with a radius* R_n*, and has a circular communication area which is also centered at* C_n *with a radius* $\bar{R}_n$*. Note that* $\bar{R}_n$ *is strictly larger than* $R_n + R_j$*,* $j \in (1, \ldots, N), j \neq n$
- $\mathcal{A}_n$ *broadcasts* $(\boldsymbol{p}_n, \boldsymbol{u}_n)$ *and receives* $(\boldsymbol{p}_j, \boldsymbol{u}_j)$ *broadcasted by other cooperative robots* $\mathcal{A}_j$*, in its communication area*
- $\mathcal{A}_n$ *can compute the relative position and velocity* $(\boldsymbol{p}_{obs_i}, \boldsymbol{u}_{obs_i})$ *of non cooperative entities within a given sensing range*
- *At the initial time* $t_{ini} \geq 0$*, each robot starts at a location outside of the safety areas of other entities.*

The objective is to find the control inputs $\boldsymbol{\tau}_n$ for each robot $\mathcal{A}_n$ such that, under Assumption 1,

- $\mathcal{A}_n$ is stabilized toward its desired point $\boldsymbol{p}_{n,des}$, i.e.

$$\lim_{t \to \infty} \|\boldsymbol{p}_n(t) - \boldsymbol{p}_{n,des}\| = 0 \tag{6}$$

- collisions are avoided
- all computations are done on board in a decentralized cooperative way

Remark 3. *It should be noted that for collision avoidance, one can distinguish two kinds of entities, i.e.*

- *cooperative robots which are involved in a detected potential collision*
- *non cooperative entities which cannot cooperate in the collision avoidance process. They represent the moving objects and static obstacles.*

3 Distributed Algorithm

In order to solve the multi-robots navigation problem, a decentralized algorithm combining two parallel processes is proposed. First, a receding horizon planner, in which each robot computes its own planned trajectory locally, achieves middle-term objectives, i.e. coordination between cooperative robots which are involved in a detected potential collision. Then, a reactive navigation controller is proposed to fulfill short-term objectives, i.e. trajectory tracking while taking into account non cooperative entities.

3.1 Conflicts and Collisions

Definition 1. *(conflict) A conflict occurs between two cooperative robots* $\mathcal{A}_n$ *and* $\mathcal{A}_j$ *at time* $t_k \in \mathbb{R}^+$ *if they are not in collision at* t_k*, but at some future time, a collision may occur.*

The following proposition, based on the well-known concept of velocity obstacle [16], is useful to check the presence of conflicts.

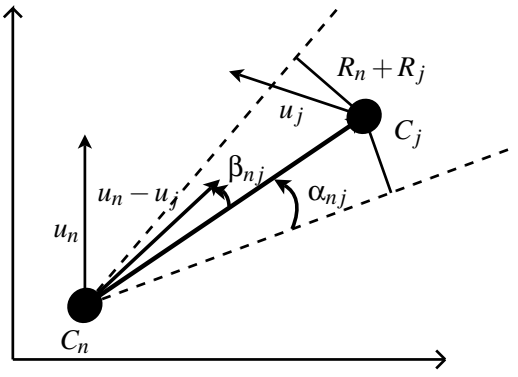


Fig. 3. Velocity obstacle concept

Proposition 1. *Let us define for each pair $(\mathcal{A}_n, \mathcal{A}_j)$, the following variables depicted in Fig. 3:*

$$\begin{aligned} \beta_{nj}(t_k) &= \arg(\boldsymbol{u}_n(t_k) - \boldsymbol{u}_j(t_k)) - \arg(\boldsymbol{p}_j(t_k) - \boldsymbol{p}_n(t_k)) \\ \alpha_{nj}(t_k) &= \arcsin\left(\frac{R_n + R_j}{\|\boldsymbol{p}_j(t_k) - \boldsymbol{p}_n(t_k)\|}\right) \end{aligned} \tag{7}$$

A necessary and sufficient condition for no conflict between $\mathcal{A}_n$ and $\mathcal{A}_j$ at t_k is:

$$|\beta_{nj}(t_k)| \geq \alpha_{nj}(t_k) \tag{8}$$

Definition 2. *(conflict subset) For each $\mathcal{A}_n$, the conflict subset $\mathcal{N}_n(t_k)$ at time $t_k \in \mathbb{R}^+$ is the set of all cooperative robots which are in the communication area of $\mathcal{A}_n$ and in conflict with $\mathcal{A}_n$.*

3.2 Receding Horizon Planner

In this section, we introduce a decentralized receding horizon planning scheme used to achieve middle-term objectives, based on [17,11]. The purpose of the distributed receding horizon planner is to decompose the overall problem into a family of simple receding horizon planning problems which are implemented in each robot $\mathcal{A}_n$. In every problem, the same planning horizon $T_p \in \mathbb{R}^+$ and update period $T_c \in \mathbb{R}^+$ ($T_c < T_p$) are used. The receding horizon updates are:

$$t_k = t_{ini} + (k-1)T_c, \quad k \in \mathbb{N} \tag{9}$$

Remark 4. *During the initialization step, that is to say before robots move, we denote $t_0 = t_{ini}$.*

At each update t_k, robots in conflict exchange information about each others (position, velocity, etc.) to predict their possible trajectories, denoted by $\widehat{\boldsymbol{p}}_j(t, t_k)$ and velocities $\widehat{\boldsymbol{u}}_j(t, t_k)$ (for all $\mathcal{A}_j$ belonging to the conflict subset). It is similar as what it is done

while driving a car. These trajectories are obtained without taking the collision avoidance constraint into account. Therefore, by design, the anticipated trajectory is the same in every receding horizon planning problem in which it occurs. At last, in parallel, every robot $\mathcal{A}_n$ computes its own planned trajectory $\boldsymbol{p}_n^*(t,t_k)$ and planned velocity $\boldsymbol{u}_n^*(t,t_k)$, over the planning horizon T_p, in order to integrate the collision avoidance between cooperative robots. From the planned trajectory and velocity associated to the planning horizon T_p, only the part which corresponds to the update horizon T_c is stored. At time t_{k+1}, a new optimization is solved over a shifted planning horizon.

Remark 5. *Note that the first argument of $\boldsymbol{p}_n^*$, $\widehat{\boldsymbol{p}}_n$, $\boldsymbol{u}_n^*$ and $\widehat{\boldsymbol{u}}_n$ denotes time. The second argument is only added to distinguish at which receding horizon update the trajectory and velocity are computed.*

The collection of distributed receding horizon planning problems is formally defined, for each robot $\mathcal{A}_n$, as follows:

Problem: At each update t_k, for each robot $\mathcal{A}_n$:
Given: the actual positions $\boldsymbol{p}_n(t_k)$, $\boldsymbol{p}_j(t_k)$ and the actual velocities $\boldsymbol{u}_n(t_k)$, $\boldsymbol{u}_j(t_k)$ of robot $\mathcal{A}_n$ and robots $\mathcal{A}_j$ belonging to $\mathcal{N}_n(t_k)$, respectively.
Find: the planned trajectory and velocity pairs $(\boldsymbol{p}_n^*(t,t_k), \boldsymbol{u}_n^*(t,t_k)), \forall t \in [t_{k+1}, t_{k+1}+T_p]$, that minimizes:

$$\int_{t_{k+1}}^{t_{k+1}+T_p} \left(a_n \sum_{j \in \mathbb{N} \mid \mathcal{A}_j \in \mathcal{N}_n(t_k)} \widehat{U}_{nj,rep}(t) + \|\boldsymbol{p}_n^*(t,t_k) - \widehat{\boldsymbol{p}}_n(t,t_k)\| \right) dt \tag{10}$$

subject to the following constraints: $\forall i \in \{i \in \mathbb{N} \mid \mathcal{A}_i \in \mathcal{N}_n(t_k) \cup \{\mathcal{A}_n\}\}$,

$$\begin{cases} \widehat{\boldsymbol{p}}_i(t_k,t_k) = \boldsymbol{p}_i(t_k) & \widehat{\boldsymbol{u}}_i(t_k,t_k) = \boldsymbol{u}_i(t_k) \\ \widehat{\boldsymbol{p}}_i(t_{fin},t_k) = \boldsymbol{p}_{i,des} & \widehat{\boldsymbol{u}}_i(t_{fin},t_k) = \boldsymbol{u}_{i,des} \\ \\ \widehat{\boldsymbol{u}}_i(t,t_k) \in \mathcal{S}_i & , \forall t \geq t_k \end{cases} \tag{11}$$

and

$$\begin{cases} \boldsymbol{p}_n^*(t_{k+1},t_k) = \boldsymbol{p}_n^*(t_{k+1},t_{k-1}) \\ \boldsymbol{u}_n^*(t_{k+1},t_k) = \boldsymbol{u}_n^*(t_{k+1},t_{k-1}) \\ \\ \boldsymbol{u}_n^*(t,t_k) \in \mathcal{S}_n & , \forall t \in [t_{k+1}, t_{k+1}+T_p] \end{cases} \tag{12}$$

where

$$\begin{aligned} \widehat{U}_{nj,rep}(t) &= \begin{cases} 0 & \text{if } \widehat{\rho}_{nj}(t) \geq b_n \\ \frac{1}{2}\left(\frac{1}{\widehat{\rho}_{nj}(t)} - \frac{1}{b_n}\right)^2 & \text{else} \end{cases} \\ \widehat{\rho}_{nj}(t) &= \|\boldsymbol{p}_n^*(t,t_k) - \widehat{\boldsymbol{p}}_j(t,t_k)\| - (R_n + R_j) \end{aligned} \tag{13}$$

a_n and b_n are strictly positive factors which can vary among robots to reflect differences in aggressiveness ($a_n < 1$, $b_n \ll 1$) and shyness ($a_n > 1$, $b_n \gg 1$).

It could be highlighted that the anticipated trajectories are computed without taking the collision avoidance constraint into account (see Eq. (11)). That is why, to integrate the path planning with local collision avoidance, the cost function (10) should be appropriately chosen.

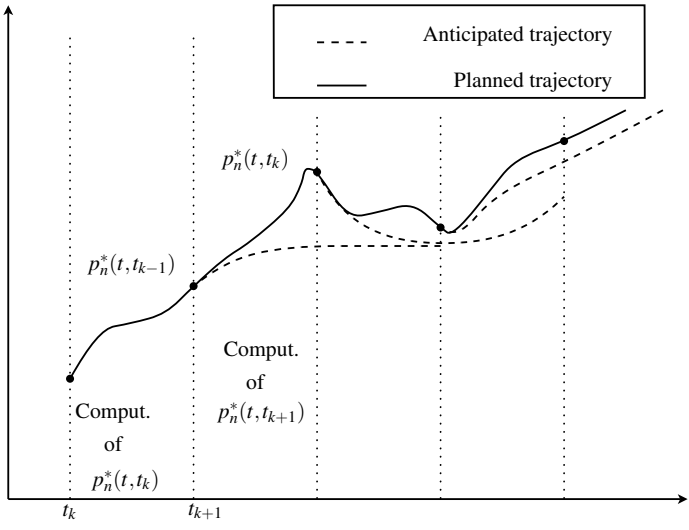


Fig. 4. Implementation of the receding horizon planner

One can note that the first part of cost (10) is designed to enforce the collision avoidance between cooperative robots. This term, based on artificial potential field [12], is designed such that it equals to infinity when a collision between robots occurs and decreases according to the relative distance between robots.

The second part of cost (10), i.e. the term $\|\boldsymbol{p}_n^*(t,t_k) - \widehat{\boldsymbol{p}}_n(t,t_k)\|$, is a way of penalizing the deviation of the optimal planned trajectory $\boldsymbol{p}_n^*(t,t_k)$ from the anticipated trajectory $\widehat{\boldsymbol{p}}_n(t,t_k)$, which is the trajectory that other robots rely on. In previous work, this term was incorporated into the decentralized receding horizon planner as a constraint [11]. The formulation presented here is an improvement over this past formulation, since the penalty yields an optimization problem that is much easier to solve.

Remark 6. *One can note that constraints (12) which guarantee the continuity of the planned trajectory and velocity need $\boldsymbol{p}_n^*(t_{k+1},t_{k-1})$ and $\boldsymbol{u}_n^*(t_{k+1},t_{k-1})$ computed in the previous step. Therefore, the proposed planner is not able to reject external disturbances or inherent discrepancies between the model and the real process. However, it takes the real time constraint into account. Indeed, each robot has a limited time to plan its trajectory. The time allocated to make its decision depends on its perception sensors, its computation delays and is less than the update period T_c (see Fig. 4).*

The discussed claim for robustness in trajectory tracking will be achieved hereafter.

Remark 7. *A compromise must be done between reactivity and computation time. Indeed, the planning horizon must be sufficiently small in order to have good enough results in terms of computation time. However, it must be higher than the update period to guarantee enough reactivity.*

Remark 8. *To numerically solve the above problems, a nonlinear trajectory generation algorithm [7] is applied. It is based on finding trajectory curves in a lower dimensional space and parameterizing these curves by B-splines. A constrained feasible sequential quadratic optimization algorithm is used to find the B-splines coefficients that optimize the performance objective while respecting the constraints.*

3.3 Reactive Navigation Controller

Hereafter, a reactive approach, which combines artificial potential fields and sliding mode control technique, for simultaneously tracking the planned trajectory while avoiding collision with unexpected entities (i.e. non cooperative entities), is proposed.

Since the robot dynamics (1)-(2) is of strict feedback systems (see [18] for details about strict feedback systems) with respect to the robot linear and angular velocities (i.e. v_n and w_n), a backstepping procedure is used to design the control input $\boldsymbol{\tau}_n$. That is why the control design is divided into two main steps.

Step 1 based on Artificial Potential Fields. Let us introduce the following notations:

$$\begin{cases} \theta_{ne} = \theta_n - \gamma_{\theta_n} \\ v_{ne} = v_n - \gamma_{v_n} \end{cases} \tag{14}$$

where γ_{θ_n} and γ_{v_n} are auxiliary variables used to avoid collisions. Replacing expressions (14) into the first two equations of (1) and using (4) yield:

$$\dot{\boldsymbol{p}}_n = \begin{bmatrix} \cos\gamma_{\theta_n} \\ \sin\gamma_{\theta_n} \end{bmatrix} \gamma_{v_n} + \Delta_{1n} + \Delta_{2n} \tag{15}$$

with $\Delta_{1n} = \gamma_{v_n} \begin{bmatrix} (\cos\theta_{ne} - 1)\cos\gamma_{\theta_n} - \sin\theta_{ne}\sin\gamma_{\theta_n} \\ \sin\theta_{ne}\cos\gamma_{\theta_n} + (\cos\theta_{ne} - 1)\sin\gamma_{\theta_n} \end{bmatrix}$ and $\Delta_{2n} = v_{ne} \begin{bmatrix} \cos\theta_n \\ \sin\theta_n \end{bmatrix}$.

The objective is to design the auxiliary variables γ_{v_n} and γ_{θ_n} such that robot $\mathcal{A}_n$ robustly tracks its planned trajectory $\boldsymbol{p}_n^*$ while avoiding unexpected collisions. Here, artificial potential functions are used in order to design an attractive force between the robot and its planned trajectory and a repulsive force to avoid collisions.

In conventional potential field method [12], the planned robot velocity $\boldsymbol{u}_n^*$ is assumed to be zero and the obstacle velocity $\boldsymbol{u}_{obs_i}$ is not considered. However, to make robot $\mathcal{A}_n$ track the planned trajectory among moving obstacles, velocities $\boldsymbol{u}_n^*$ and $\boldsymbol{u}_{obs_i}$ play key roles. This issue will be addressed by extending the results given in [19]. Let us consider the conventional potential function [12]:

$$U_n = U_{n,att} + U_{n,rep} \tag{16}$$

where $U_{n,att}$ and $U_{n,rep}$ are, respectively, the attractive potential defined to track the planned trajectory $\boldsymbol{p}_n^*$ and the repulsive potential related to collision avoidance, specified as follows:

– The attractive potential is designed such that it puts penalty on the tracking error and is equal to zero when the robot is at its desired position, i.e.

$$U_{n,att} = \frac{1}{2} \left\| \boldsymbol{p}_n - \boldsymbol{p}_n^* \right\|^2 \tag{17}$$

– The repulsive potential is designed such that it equals to infinity when a collision occurs with $\mathcal{A}_n$ and decreases according to the relative distance between $\mathcal{A}_n$ and an obstacle, i.e.

$$U_{n,rep} = c_n \sum_i U_{ni,rep} \tag{18}$$

with

$$U_{ni,rep} = \begin{cases} 0 & \text{if } \rho_{ni} \geq d_n \\ \frac{1}{2}\left(\frac{1}{\rho_{ni}} - \frac{1}{d_n}\right)^2 & \text{else} \end{cases} \tag{19}$$

where ρ_{ni} is the minimum distance between robot $\mathcal{A}_n$ and the obstacle i. c_n and d_n are strictly positive factors which have similar properties as a_n and b_n.

Proposition 2. *If the errors θ_{ne} and v_{ne} are asymptotically stable, $\mathcal{A}_n$ robustly tracks its planned trajectory $\boldsymbol{p}_n^*$ while avoiding collisions using the auxiliary variables:*

$$\gamma_{v_n} = \left[(\|\boldsymbol{u}_n^*\|\cos(\theta_n^* - \psi_n) - c_n \textstyle\sum_i \xi_{ni}\|\boldsymbol{u}_{obs_i}\|\cos(\theta_{obs_i} - \psi_{ni}) + \|\boldsymbol{p}_n - \boldsymbol{p}_n^*\|)^2 + \|\boldsymbol{u}_n^*\|^2 \sin^2(\theta_n^* - \bar{\psi}_n)\right]^{0.5}$$

$$\gamma_{\theta_n} = \begin{cases} \bar{\psi}_n + \arcsin\left(\frac{\|\boldsymbol{u}_n^*\|\sin(\theta_n^* - \bar{\psi}_n)}{\gamma_{v_n}}\right) & \text{if } \gamma_{v_n} \neq 0 \\ \theta_n^* & \text{if } \gamma_{v_n} = 0 \end{cases} \tag{20}$$

with

$$\begin{aligned} \theta_n^* &= \arg(\boldsymbol{u}_n^*) \\ \theta_{obs_i} &= \arg(\boldsymbol{u}_{obs_i}) \\ \psi_n &= \arg(\boldsymbol{p}_n^* - \boldsymbol{p}_n) \\ \psi_{ni} &= \arg(\boldsymbol{p}_{obs_i} - \boldsymbol{p}_n) \\ \bar{\psi}_n &= \arctan\left(\frac{\sin\psi_n - c_n\sum_i \xi_{ni}\sin\psi_{ni}}{\cos\psi_n - c_n\sum_i \xi_{ni}\cos\psi_{ni}}\right) \\ \xi_{ni} &= \begin{cases} 0 & \text{if } \rho_{ni} \geq d_n \\ \left(\frac{1}{\rho_{ni}} - \frac{1}{d_n}\right)\frac{1}{(\rho_{ni})^2}\frac{1}{\|\boldsymbol{p}_n - \boldsymbol{p}_n^*\|} & \text{else} \end{cases} \end{aligned}$$

Proof. Let us differentiate U_n with respect to time in equation (16), i.e.:

$$\dot{U}_n = \dot{U}_{n,att} + \dot{U}_{n,rep} \tag{21}$$

Substituting (20) into (21) yields after some geometric manipulations:

$$\begin{aligned} \dot{U}_n &= \|\boldsymbol{p}_n^* - \boldsymbol{p}_n\|(\|\boldsymbol{u}_n^*\|\cos(\theta_n^* - \psi_n) - \gamma_{v_n}\cos(\gamma_{\theta_n} - \psi_n) - \\ &\quad c_n\textstyle\sum_i \xi_{ni}(\|\boldsymbol{u}_{obs_i}\|\cos(\theta_{obs_i} - \psi_{ni}) + \gamma_{v_n}\cos(\gamma_{\theta_n} - \psi_{ni}))) \\ &\quad + \left((\boldsymbol{p}_n - \boldsymbol{p}_n^*)^T - c_n\textstyle\sum_i \xi_{ni}\frac{\|\boldsymbol{p}_n - \boldsymbol{p}_n^*\|}{\|\boldsymbol{p}_n - \boldsymbol{p}_{obs_i}\|}(\boldsymbol{p}_n - \boldsymbol{p}_{obs_i})^T\right)(\Delta_{1n} + \Delta_{2n}) \\ &= \|\boldsymbol{p}_n^* - \boldsymbol{p}_n\|(\|\boldsymbol{u}_n^*\|\cos(\theta_n^* - \psi_n) - \gamma_{v_n}\cos(\gamma_{\theta_n} - \bar{\psi}_n) \\ &\quad - c_n\textstyle\sum_i \xi_{ni}(\|\boldsymbol{u}_{obs_i}\|\cos(\theta_{obs_i} - \psi_{ni}))) \\ &\quad + \left((\boldsymbol{p}_n - \boldsymbol{p}_n^*)^T - c_n\textstyle\sum_i \xi_{ni}\frac{\|\boldsymbol{p}_n - \boldsymbol{p}_n^*\|}{\|\boldsymbol{p}_n - \boldsymbol{p}_{obs_i}\|}(\boldsymbol{p}_n - \boldsymbol{p}_{obs_i})^T\right)(\Delta_{1n} + \Delta_{2n}) \\ &= \|\boldsymbol{p}_n^* - \boldsymbol{p}_n\|(\|\boldsymbol{u}_n^*\|\cos(\theta_n^* - \psi_n) - \sqrt{\gamma_{v_n}^2 - \|\boldsymbol{u}_n^*\|^2\sin^2(\theta_n^* - \bar{\psi}_n)} - \\ &\quad c_n\textstyle\sum_i \xi_{ni}(\|\boldsymbol{u}_{obs_i}\|\cos(\theta_{obs_i} - \psi_{ni}))) \\ &\quad + \left((\boldsymbol{p}_n - \boldsymbol{p}_n^*)^T - c_n\textstyle\sum_i \xi_{ni}\frac{\|\boldsymbol{p}_n - \boldsymbol{p}_n^*\|}{\|\boldsymbol{p}_n - \boldsymbol{p}_{obs_i}\|}(\boldsymbol{p}_n - \boldsymbol{p}_{obs_i})^T\right)(\Delta_{1n} + \Delta_{2n}) \end{aligned}$$

Assuming that the errors θ_{ne} and v_{ne} are asymptotically stable (i.e. $\Delta_{1n} = \Delta_{2n} = 0$), one can get from (20):

$$\dot{U}_n \leq -\|\boldsymbol{p}_n^* - \boldsymbol{p}_n\|^2 \tag{22}$$

Since $U_n \geq 0$ and $\dot{U}_n \leq 0$, U_n is bounded. That is why $\mathcal{A}_n$ robustly tracks its planned trajectory $\boldsymbol{p}_n^*$ while avoiding collisions. □

Step 2 based on Sliding Mode Technique. Now, the objective is to force the motion of robot $\mathcal{A}_n$ such that the errors θ_{ne} and v_{ne} are asymptotically stable. The proposed strategy is based on the so-called second order sliding mode control (SMC) approach. The SMC methodology [20] is chosen because it is a robust technique to control nonlinear systems operating under uncertainty conditions [21]. Furthermore, second order SMC can reduce the chattering phenomenon (high frequency vibrations of the controlled system which degrade the performances). Indeed, instead of influencing the first sliding variable time derivative, the signum function acts on its second time derivative. This method can also achieve a better convergence accuracy with respect to discrete sampling time than conventional SMC (see [23] for a survey).

Let us apply to system (1)-(2) the following preliminary feedback:

$$\bar{\boldsymbol{\tau}}_n = (M_n B_n)^{-1} (\boldsymbol{\tau}_n - D_n \boldsymbol{z}_n) \tag{23}$$

where $\bar{\boldsymbol{\tau}}_n = [\bar{\tau}_{1n}, \bar{\tau}_{2n}]^T$ is the auxiliary control input. Thus, system (1)-(2) can be expressed as follows:

$$\dot{\boldsymbol{\eta}}_n = J(\boldsymbol{\eta}_n) B_n \begin{bmatrix} v_n \\ w_n \end{bmatrix} \tag{24}$$

$$\begin{bmatrix} \dot{v}_n \\ \dot{w}_n \end{bmatrix} = \bar{\boldsymbol{\tau}}_n \tag{25}$$

Since the relative degree of system (24)-(25) with respect to the sliding variable v_{ne} is only one, a dynamic extension is done before designing the control (see [22] for further details). Thus, an integrator chain is added on the input variable $\bar{\tau}_{1n}$.

There are several algorithms to ensure the finite time stabilization of the sliding variables θ_{ne} and v_{ne} towards the origin. Among them, the sampled twisting algorithm [23] has been developed for systems with relative degree two. This algorithm provides good convergence accuracy and robustness properties. It does not require the knowledge of the time derivative of the sliding variables and takes into account some practical constraints such as the sampling of the measurement and the control.

Proposition 3. *Consider system (1)-(2). The errors θ_{ne} and v_{ne} are stable in finite time under the nonlinear controller defined in (23) where*

$$\begin{aligned} \dot{\bar{\tau}}_{1n} &= \begin{cases} -\lambda_{1,M} \operatorname{sign}(\theta_{ne}) \ \textit{if} \ \theta_{ne} \Delta_{\theta_{ne}} > 0 \\ -\lambda_{1,m} \operatorname{sign}(\theta_{ne}) \ \textit{if} \ \theta_{ne} \Delta_{\theta_{ne}} \leq 0 \end{cases} \\ \bar{\tau}_{2n} &= \begin{cases} -\lambda_{2,M} \operatorname{sign}(v_{ne}) \ \textit{if} \ v_{ne} \Delta_{v_{ne}} > 0 \\ -\lambda_{2,m} \operatorname{sign}(v_{ne}) \ \textit{if} \ v_{ne} \Delta_{v_{ne}} \leq 0 \end{cases} \end{aligned} \tag{26}$$

with

$$\Delta_{\theta_{ne}} = \begin{cases} 0 & \text{if } k = 0 \\ \theta_{ne}(kT_s) - \theta_{ne}((k-1)T_s) & \text{else} \end{cases}$$
$$\Delta_{v_{ne}} = \begin{cases} 0 & \text{if } k = 0 \\ v_{ne}(kT_s) - v_{ne}((k-1)T_s) & \text{else} \end{cases} \tag{27}$$

T_s is the sampling period, $k \in \mathbb{N}$ is related to the time of the process and $\lambda_{i,m}$, $\lambda_{i,M}$, $i = 1, 2$ are positive constants high enough to enforce the sliding motion.

Proof. It can be shown that this controller ensures a finite time convergence of the trajectories onto the manifold $\{v_{ne} = \dot{v}_{ne} = 0\}$ and $\left\{\theta_{ne} = \dot{\theta}_{ne} = 0\right\}$. Hence, the application of the control input (26) results in the robust finite time stabilization of θ_{ne} and v_{ne}. □

Remark 9. *We would like to emphasize that although not explicitly considered here the procedure based on sliding mode control guarantees proper behavior even in the presence of uncertainties in the mass and inertia of the robots and additive disturbances to the linear and angular velocities which constitute very realistic assumptions.*

Once the sliding mode occurs on all the surfaces (which happens in finite time), based on Proposition 2, the global control objectives, defined in Section 2.2, are fulfilled.

Some specific advantages of the proposed decentralized algorithm are enumerated below:

- robustness with respect to uncertainties and disturbances (sliding mode controller),
- reactivity (potential field functions),
- low communication bandwidth, i.e. small amount of information is locally exchanged,
- reduction of deadlocks due to local minima in potential field (anticipation and coordination mechanism through the receding horizon planner).

4 Simulation Results

This section demonstrates the performance of the proposed decentralized algorithm. The following simulations showcase two different scenarios for which the environment is partially known (i.e. the range of sensors of each robot is of radius $1.5m$). The main parameters of the robots are: $\forall n$, $R_n = 0.25m$, $\bar{R}_n = 4m$ and $u_{n,max} = 1m/s$. For the decentralized algorithm, the following parameters are used: $T_p = 3s$, $T_c = 0.5s$, $a_n = c_n = 1$, $b_n = 2$, $d_n = 0.5$, $\lambda_{1,M} = \lambda_{2,M} = 10$, $\lambda_{1,m} = \lambda_{2,m} = 1$ and $T_s = 0.01s$.

4.1 Scenario 1: Crossing

In this scenario, there are four robots ($N = 4$) starting at $\boldsymbol{p}_1(t_{ini}) = [5, 0]^T$, $\boldsymbol{p}_2(t_{ini}) = [15, 0]^T$, $\boldsymbol{p}_3(t_{ini}) = [10, 5]^T$ and $\boldsymbol{p}_4(t_{ini}) = [10, -5]^T$ respectively, with velocities equal to zero. These robots must cross each other in order to reach their desired configuration $\boldsymbol{p}_{1,des} = [15, 0]^T$, $\boldsymbol{p}_{2,des} = [5, 0]^T$, $\boldsymbol{p}_{3,des} = [10, -5]^T$ and $\boldsymbol{p}_{4,des} = [10, 5]^T$. One can note that this problem is not trivial due to its symmetry properties.

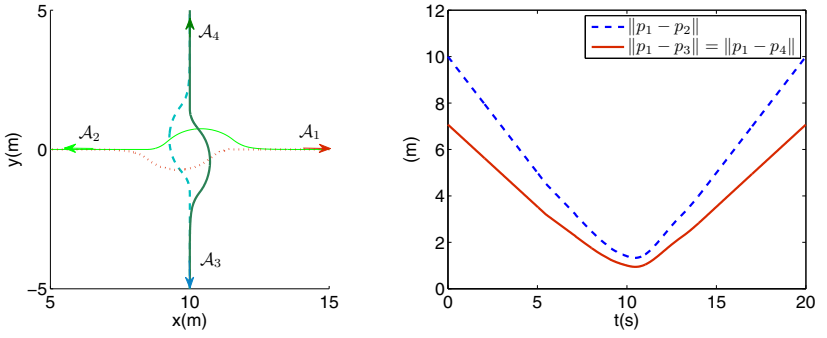


Fig. 5. Four vehicles simulation: (a) Robot trajectories. (b) Relative distances between $\mathcal{A}_1$ and other robots.

The simulation results are given in Fig. 5. One can see that each robot modifies its trajectory in order to avoid collision. Figure 5(b) depicts the evolution of the distance between robots. Since it is higher than $0.5m$, the collision avoidance is guaranteed.

4.2 Scenario 2: Reconfiguration with Collision Avoidance

In this scenario, a swarm of five robots ($N = 5$) reconfigures its geometric shape (from "linear" to "triangular") while avoiding collisions with obstacles. The proposed

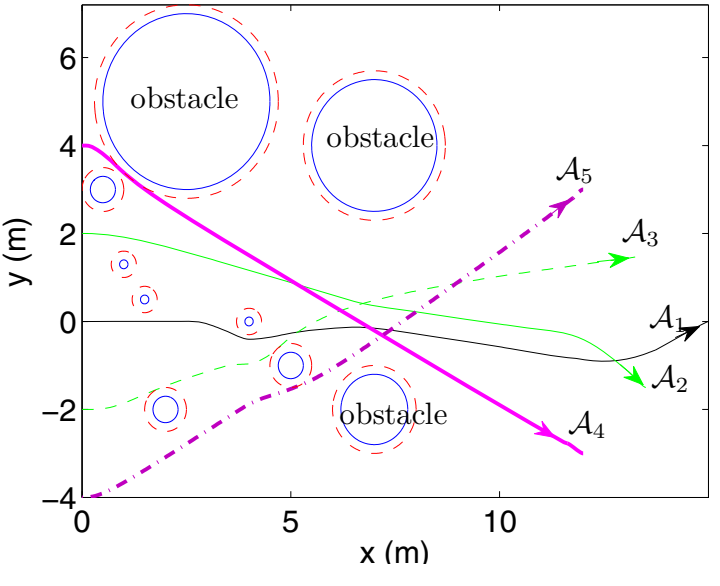


Fig. 6. Collision avoidance of five robots

decentralized controller has only a limited knowledge of the obstacles (initially unknown). It simply keeps the robots spaced out using the proposed potential field technique. The five robots make decisions in order to avoid collisions. One can note that the number of potential conflicts is high.

One can see in Fig. 6 that under the proposed decentralized algorithm, the robots meet the objective defined in Section 2.2. Note that the radius of obstacles is increased by 0.25m (dotted lines around obstacles) to take the size of robots into account.

5 Conclusions

A new distributed strategy for the navigation of multiple autonomous robots is presented. The proposed scheme combines a decentralized receding horizon motion planner to satisfy middle-term objectives (coordination between cooperative robots) with a fast navigation controller based on artificial potential fields and sliding mode control technique to satisfy short-term objectives (collision avoidance and trajectory tracking). The fact that there is no leader increases the security and the robustness of the missions. Simulation studies are provided in order to show the effectiveness of the proposed approach.

In the future, it is planned to design real time observers to estimate the relative velocities between robots.

References

1. Kuchar, J., Yang, L.: A review of conflict detection and resolution modeling methods. IEEE T. Intell. Transp. 1, 179–189 (2000)
2. Tomlin, C., Pappas, G., Sastry, S.: Conflict resolution for air traffic management: a study in multiagent hybrid systems. IEEE T. Automat. Contr. 43, 509–521 (1998)
3. Das, A., Fierro, R., Kumar, V., Ostrowski, J., Spetzer, J., Taylor, C.: A vision-based formation control framework. IEEE T. Robotic Autom. 18, 813–825 (2002)
4. Defoort, M., Floquet, T., Kokosy, A., Perruquetti, W.: Sliding mode formation control for cooperative autonomous mobile robots. IEEE T. Ind. Elect. 55, 3944–3953 (2008)
5. Pallatino, L., Scordio, V., Bicchi, A., Frazoli, E.: Decentralized cooperative policy for conflict resolution in multivehicle systems. IEEE T. Robot. 23, 1170–1183 (2007)
6. Mayne, D., Rawlings, J., Rao, C., Scokaert, P.: Constrained model predicitive control: Stability and optimality. Automatica 36, 789–814 (2000)
7. Defoort, M., Palos, J., Kokosy, A., Floquet, T., Perruquetti, W.: Performance based reactive navigation for nonholonomic mobile robots. Robotica 27, 281–290 (2009)
8. Dunbar, W., Murray, R.: Model predictive control of coordinated multi-vehicle formation. In: IEEE Conf. on Decision and Control (2002)
9. Dunbar, W., Murray, R.M.: Distributed receding horizon control for multi-vehicle formation stabilization. Automatica 42, 549–558 (2006)
10. Kuwata, Y., Richards, A., Schouwenaraars, T., How, J.: Decentralized robust receding horizon control. In: American Control Conf. (2006)
11. Defoort, M., Kokosy, A., Floquet, T., Perruquetti, W., Palos, J.: Motion planning for cooperative unicycle-type mobile robots with limited sensing ranges: A distributed receding horizon approach. Robotics and autonomous systems 57, 1094–1106 (2009)
12. Latombe, J.: Robot Motion Planning. Kluwer Academic Publishers, Norwell (1991)

13. Canudas-De-Wit, C., Sordalen, O.: Exponential stabilization of mobile robots with nonholonomic constraints. IEEE T. Automat. Contr. 37, 1791–1797 (1992)
14. Do, K., Jiang, Z., Pan, J.: A global output-feedback controller for simultaneous tracking and stabilization of unicycle-type mobile robots. IEEE T. Automat. Contr. 20(3), 589–594 (2004)
15. Fliess, M., Levine, J., Martin, P., Rouchon, P.: Flatness and defect of nonlinear systems: introductory theory and examples. Int. J. Control 61(6), 1327–1361 (1995)
16. Fiorini, P., Shiller, Z.: Motion planning in dynamic environments using velocity obstacles. Int. J. Robot. Res. 17(7), 760–772 (1998)
17. Keviczky, T., Borrelli, F., Balas, G.J.: Decentralized receding horizon control for large scale dynamically decoupled systems. Automatica 42(12), 2105–2115 (2006)
18. Krstic, M., Kanellakopoulos, I., Kokotovic, P.: Nonlinear and adaptative control design. Wiley, N.Y (1995)
19. Huang, L.: Velocity planning for a mobile robot to track a moving target - a potential field approach. Robot. Auton. Syst. 57(1), 55–63 (2009)
20. Utkin, V., Guldner, J., Shi, J.: Sliding Modes Control in Electromechanical Systems. Taylor and Francis, New York (1999)
21. Defoort, M., Floquet, T., Kokosy, A., Perruquetti, W.: A novel higher order sliding mode control scheme. Systems and Control Letters 58, 102–108 (2009)
22. Isidori, A.: Nonlinear control systems, 2nd edn. Springer, New York (1989)
23. Fridman, L., Levant, A.: Higher order sliding mode modes. In: Perruquetti, W., Barbot, J.P. (eds.) Sliding mode control in Engineering, pp. 53–101 (2002)

Task Partitioning in Swarms of Robots: Reducing Performance Losses Due to Interference at Shared Resources

Giovanni Pini, Arne Brutschy, Mauro Birattari, and Marco Dorigo

IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
{gpini,arne.brutschy,mbiro,mdorigo}@ulb.ac.be

Abstract. The performance of large groups of robots is often limited by a commonly shared resource. This effect, termed interference, can have a large impact on robotic swarms. This article studies the issue of interference in a swarm of robots working on a harvesting task. The environment of the robots is spatially constrained, i.e., there is a commonly shared resource, the nest, which limits the group's performance when used without any arbitration mechanism. The article studies the use of task partitioning for reducing concurrent accesses to the resource, and therefore limiting the impact of interference on the group's performance. In our study, we spatially partition the environment into two subparts, thereby partitioning the corresponding harvesting task as well. We employ a simple method to allocate individuals to the partitions. The approach is empirically studied both in an environment with a narrow nest area and an environment without this constraint. The results of the task partitioning strategy are analyzed and compared to the case in which task partitioning is not employed.

1 Introduction

In collective robotics, interference is a critical problem limiting the growth of a group: the time each robot spends in non-task-relevant behaviors such as obstacle avoidance increases when the density of individuals rises—see e.g., [1]. The performance on tasks that suffer from physical interference can typically be improved by spatial partitioning; for example, by keeping each robot in its own "working area". A known approach that uses this rationale is the so called bucket-brigade [2,3]. In this approach, robots hand over objects to robots working in the following area, until the objects reach their destination. As tasks usually cannot be partitioned arbitrarily, this approach effectively limits the number of robots that can be employed in the task. A possible solution to this problem, treating working areas as non-exclusive, raises other problems: How should individuals be allocated to tasks? How can such an allocation help in limiting the amount of interference?

In this paper, we study how task partitioning can help in reducing sources of interference. Additionally, we show a simple way to achieve self-organized allocation to such a task partition when using a robotic swarm.

We use the foraging problem, one of the canonical testbeds for collective robotics, as a base for our studies. In our experiments, a swarm of homogeneous robots has to

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 217–228.
springerlink.com

harvest prey objects from a source area and transport them to a home area. In this study, we limit ourselves to a harvesting task that is pre-partitioned by the designer into two subtasks with a sequential interdependency. We study a simple threshold-based model of self-organized allocation and focus on two questions: Under which environmental conditions is it advantageous to partition the task? Can this partition reduce interference between group members? These questions are studied in two experiments using a simulated robot swarm.

The paper is organized as follows. We first review related works in Sect. 2. In Sect. 3 we explain the task partitioning and the allocation method employed in this study. Section 4 gives the methods used in the experiments by describing the environments, the simulated robots, and the controller. In Sect. 5 the results of the experiments are given and discussed. Section 6 draws some conclusions and discusses future work.

2 Related Work

Interference has long been acknowledged as being one of the key issues in multi-robot cooperation [4]. Lerman and Galstyan devised a mathematical model that allows a quantification of the interference and its effect on group performance [1]. Probably, the most thorough study was published by Goldberg, who identified several types of multi-robot interactions [5]. Goldberg notes that one of the most common types of interference is physical interference in a central area, for example the nest. This kind of interference results from resource conflicts, in this case physical space, and can be arbitrated by either making sure that robots stay in different areas all the time or by employing a scheduling mechanism to ensure that robots use the same space only at different times.

A simple method for reducing interference by using the first arbitration method mentioned is the so-called bucket-brigade: robots are forced to stay in exclusive working areas and to pass objects to the following robot as soon as they cross the boundaries of their area [2,3]. Recently, this has been extended to work with adaptive working areas by Lein and Vaughan [6]. To the best of our knowledge, current works concerned with bucket brigading only studied the influence of interference due to obstacle avoidance. Other sources of interference (e.g., object manipulation) were never studied, although they might have a critical impact on the performance of any task partitioning approach. To quote Shell and Matarić [3]: *"If the cost of picking up or dropping pucks is significant [. . .], then bucket brigading may not be suitable."*

Task allocation for multi-robot systems is a wide field, which can be divided in intentional and self-organized task allocation. Intentional task allocation relies on negotiation and explicit communication to create global allocations, whereas in self-organized task allocation global allocations result from local, stochastic decisions. A formal analysis and taxonomy that covers intentional task allocation has been proposed by Gerkey and Matarić [7]. Kalra and Martinoli recently compared the two best-known approaches of intentional and self-organized task allocation [8].

The field of self-organized task allocation is in its early stages, as most studies tackle simple problems without task interdependencies. An exception is the work of Scheidler et al, in which helper components of a computing system are allocated to tasks with sequential interdependencies by using a threshold based method [9]. Studies in

self-organized task allocation are mostly based on threshold-based approaches, taking inspiration from division of labor in social insects. Krieger and Billeter were among the first to propose threshold-based approaches in multi-robot task allocation [10]. Labella et al. used threshold-based task allocation in a multi-foraging task [11]. Similarly, Campo and Dorigo used a notion of the group's internal energy to allocate individuals to a multi-foraging task [12]. Finally, Liu et al. studied a multi-foraging task while focusing on the influence of the use of different social cues on the overall group performance [13].

3 Task Partitioning and Allocation

In this work, we study a collective foraging task. By spatially partitioning the environment, the global foraging task is automatically partitioned into two subtasks: 1) harvesting prey objects from a harvesting area (*source*) and 2) transporting them to a home area (*nest*). Robots working on the first subtask harvest prey objects from the source and pass them to the robots working on the second subtask, which store the objects in the nest. These subtasks have a sequential interdependency in the sense that they have to be performed one after the other in order to complete the global task once: delivering a prey object to the home area. Robots can decide to switch from one subtask to the other, thus creating a task allocation problem: individual robots have to be allocated to subtasks and different allocations yield different performance. As a prey object has to be passed directly from one robot to the other, a robot usually has to wait some time before passing a prey object to or receiving a prey object from a robot working on the other subtask. This waiting time can therefore give an indication of the allocation quality for the respective subtask: if the waiting time is very long, there might not be enough robots allocated to the other subtask. Thus, the robots can use this waiting time to decide whether to switch subtask or not. Ideally, the waiting time should be the same for the two subtasks in order for the system to reach a stable state and deliver optimal performance.

Our robots exploit a simple threshold-based model to decide when to switch task: when the waiting time t_w is higher than a threshold Θ, a robot switches its subtask. The robot's waiting time is a function of the average time the robots working in the other subtask need to complete their task. The task-completion time of a robot depends on two factors: 1) round-trip-time (i.e., distance to travel) and 2) time lost due to interference. Thus, the robot's threshold Θ is a function of the round-trip-time and the interference of the robots in the other subtask. Therefore, the optimal task switching threshold depends on the task (i.e., time to harvest a prey object) and the environment (i.e., distance between the source and the nest). As the parameters of the environment are not pre-programmed into the robots, determining the optimal threshold can be a complex problem. In this work, we limit ourselves to a simple method for setting this threshold: at the start of the experiment, each robot draws a random threshold that is used as its task switching threshold throughout the experiment.

In the following, we study the properties of this simple self-organized task allocation strategy, compare this strategy to a strategy without task partitioning, and analyze how it can help to reduce interference. We refer to the two strategies as *partitioned* and *non-partitioned*, respectively.

4 Methods

This section describes the environments in which the experiments are carried out, the simulated robots, and the robot's controller. Additionally, we describe how we run the experiments and we introduce some metrics that we use to evaluate the properties of the system.

4.1 Environments

We study task allocation in two different environments. In these two environments, the nest is marked by a light source that can be perceived by all robots, thus providing directional information. The environment is spatially partitioned in two parts: the source is located on the left and the nest is located on the right side of the arena. We refer to the two sides of the arena as *harvest area* and *store area*, respectively. The *exchange zone* is located between these two areas. Robots working on the left side, called *harvesters*, gather prey objects in the source and move them to the exchange zone, where they pass them to the robots working on the other side. These are referred to as *storers*: their role is to transport prey objects to the nest and store them there. The nest, the source, and the exchange zone can be detected through environmental cues (ground color).

At time $t = 0$, the robots are randomly placed in the harvest area. The experiments run for $t_{\text{max}} = 18,000$ time steps (a simulated time of one hour, with a time step length of $200\,\text{ms}$). The experiments are run in two different arenas (see Fig. 1). The first arena (Fig. 1a) is $4.125\,\text{m}$ long with a width of $1.6\,\text{m}$ at the source and exchange zone, whereas the nest is $0.4\,\text{m}$ wide. The exchange zone is located $3.125\,\text{m}$ away from the source. This arena is characterized by the presence of an area, critical for the task, in which high interference between robots can be expected (the nest). Thus, this arena is referred to as the *narrow-nest* environment.

The second arena (Fig. 1b) has a rectangular shape: it is $3.75\,\text{m}$ long and $1.6\,\text{m}$ wide. Here as well the exchange zone is located $3.125\,\text{m}$ from the source. The arena shape does not suggest the presence of any zone where interference can be higher than in other places. This arena is referred to as the *wide-nest* environment.

The area of both arenas is $6\,\text{m}^2$, $5\,\text{m}^2$ for the harvest area and $1\,\text{m}^2$ for the store area. The overall area is the same in the two arenas, so that the same group size results in the same robot density. Thus, results are comparable across the two environments.

4.2 Simulation

The experiments are carried out in a custom simulation environment that models geometries and functional properties of simple objects and robots. Our robots' model is purely kinematic. Prey objects are simulated as an attribute a robot can posses and not as physical entities. Although the experiments are conducted in simulation only, the simulated robots have a real counterpart: the swarm-bot robotic platform [14]. The platform consists of a number of mobile autonomous robots called s-bots, which have been used for several studies, mainly in swarm intelligence and collective robotics—see for instance Groß at al. [15] and Nouyan et al. [16]. The simulated s-bots are of round

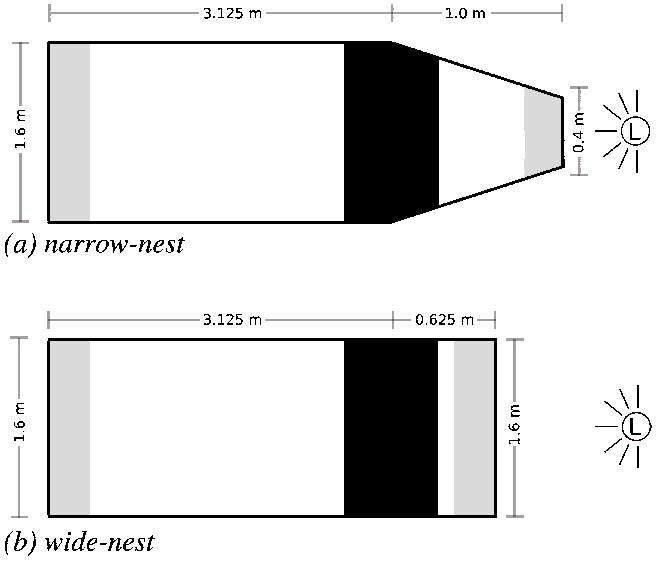


Fig. 1. Depiction of (a) the *narrow-nest* environment used in the first experiment and (b) the *wide-nest* environment used in the second experiment. The gray stripes are the source (left), and the nest (right), each $0.25\,\mathrm{m}$ deep. The black stripe is the exchange zone, that is $0.5\,\mathrm{m}$ deep. The light source is marked with "L".

shape, with a diameter of $0.116\,\mathrm{m}$. Each of them is equipped with 16 infrared proximity sensors, used to perceive obstacles up to a distance of $0.15\,\mathrm{m}$. Eight ambient light sensors can be used to perceive light gradients up to a distance of $5.0\,\mathrm{m}$. The robots are equipped with 4 ground sensors used to perceive nest, source and exchange zone. A 8 LEDs ring is used to signal when a prey object is carried. An omni-directional camera allows the perception of LEDs in a circle of radius $0.6\,\mathrm{m}$ surrounding the robot. A uniform noise of 10% is added to all sensor readings at each simulation step. The robots can move at a maximum speed of $0.1\,{}^{\mathrm{m}}/_{\mathrm{s}}$ by means of a differential drive system.

4.3 Controller

All the robots share the same, hand coded, finite state machine controller depicted in Fig. 2. The controller consists of two parts, each corresponding to a possible subtask a robot can perform. Gray states refer to the harvest subtask, white states to the store subtask. Since all the robots start in the harvest area, their controller is initially set to perform anti-phototaxis. In this way they will reach the source, where they can start retrieving prey objects. The behavior of each robot is a function of the task it is performing. Harvesters not carrying a prey object move towards the source, where they can find prey. Harvesters carrying a prey object, move to the exchange zone and wait for a free storer. Upon arrival of such a storer, the harvester passes the prey object to it. Storers carrying a prey object move towards the nest, where they can deposit the object. Storers not carrying a prey object head to the exchange zone and search for a harvester with a prey

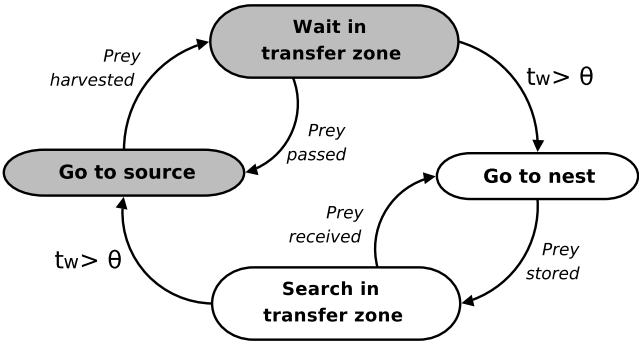


Fig. 2. Simplified state diagram of the controller of the robots. Gray states belong to the harvest task, white states to the store task. The obstacle avoidance state has been omitted for clarity, as it is applicable in all states of the robot. t_{w} is the time spent in the exchange zone and Θ is the threshold.

object. Robots can detect other robots carrying a prey on the basis of the color of their LED ring. While moving, each robot avoids obstacles (walls and other robots).

Task switches can occur: a harvester carrying a prey object can decide to become a storer, and a storer not carrying a prey object can decide to become a harvester. As mentioned before, robots switch task depending on an internal threshold Θ, representing the maximum amount of control cycles they can spend in the transfer zone waiting to pass (harvesters) or receive (storers) a prey object. If a robot remains in the transfer zone longer than its threshold without passing or receiving prey objects ($t_{\mathrm{w}} > \Theta$), it switches its task. The optimal threshold value is not trivial to determine. In the work presented here, we use a simple method to set the threshold Θ: at the beginning of the experiment, each robot draws a random threshold, sampled uniformly in the interval $[0, 1000]$. We chose this method because it is independent of the environment and does not rely on complex approximation techniques. The threshold value does not change during the experiment. In case of the non-partitioned strategy, the threshold is set to $\Theta = 0$, causing the robots to switch subtask immediately as soon as they reach the exchange zone.

4.4 Experiments

The goal of the experiments is to investigate whether task partitioning can reduce interference in task-critical zones, and how to allocate a robotic swarm to partitions. As pointed out by Lerman and Galstyan [1], interference is related to the number of individuals in the system. Additionally, the physical interference between robots is also a function of the environment the robots act in. The higher the group size, the higher the density, resulting in a higher amount of physical interference. Thus, in order to study interference in our experiments, we increase the size of the group in each of the two environments shown in Fig. 1, while using both strategies (non-partitioned and partitioned). We study the performance of the system when the group size N ranges in the interval $[1, 40]$. We run 50 repetitions for each value of N and each experimental setting.

4.5 Metrics

In order to quantify the influence of interference, we measure the *group performance* P by the number of prey objects collected by the swarm at the end of the experiment ($t_{\text{max}} = 1$ hour). From the group performance measure we can derive the *individual efficiency* as follows:

$$I_{\text{eff}} = P/N, \tag{1}$$

where N is the size of the group. Individual efficiency can help to understand the effect of interference on the performance.

In order to measure the influence of environmental features on the interference, we define an interference measure taking inspiration from Rosenfeld et al. [17]. In their work, interference is measured as the time spent performing actions not strictly related to the task, but rather lost due to negative interactions with the environment (e.g., obstacle avoidance maneuvers). By registering the number of collisions for each area of the arena, we can draw conclusions about where physical interferences happen most often. We measure interference through the state of the controller: in our case a robot is experiencing interference each time its controller perceives an obstacle.

In case of a partitioned task, there is another source of inefficiency that adds to interference: the time lost in the exchange zone. We define the *strategy cost* C as the sum of time lost because of physical interference and time lost in the exchange zone:

$$C = T_{\text{int}} + T_{\text{part}}, \tag{2}$$

where T_{int} is the amount of time steps during which the controller perceives an obstacle, and T_{part} is the total amount of time steps spent in prey passing maneuvers. By using this metric, the cost of the non-partitioned strategy is purely due to interference ($T_{\text{part}} = 0$), while in case of the partitioned strategy, prey passing costs add to interference costs. In a way, passing a prey object produces another kind of interference in the system. The strategy cost captures this effect, thus allowing for a comparison of strategies.

5 Results and Discussion

The graphs in Fig. 3a and 4a show the performance P for different group sizes in the narrow-nest and wide-nest environment respectively.

Figure 3b and and 4b show the individual efficiency I_{eff} of the robots in the narrow-nest and wide-nest environment respectively. Black curves are the average computed over the 50 repetitions of each setting, gray curves indicate the 95% confidence interval on the expected value. The performance graph in Fig. 3a shows that the partitioned strategy improves performance in the narrow-nest environment. The graph shows that the non-partitioned strategy performs better than the partitioned strategy for small group sizes (up to $N = 13$ robots). However, increasing the group size makes the non-partitioned strategy collapse: the number of gathered prey objects drops dramatically for groups larger than 13. The individual efficiency graph (Fig. 3b) can explain the behavior of the system. The robots employing the partitioned strategy are less efficient, for small group sizes, than those performing the non-partitioned strategy. However, the addition of more individuals affects the efficiency of the non-partitioned group in a

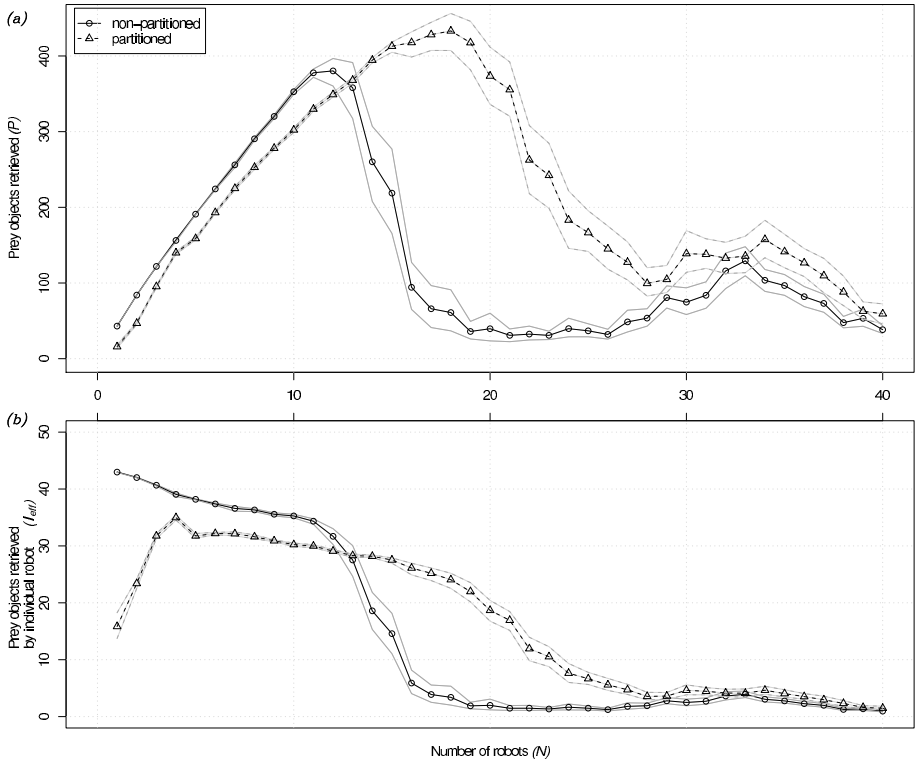


Fig. 3. (a) Performance P and (b) individual efficiency I_{eff} for increasing number of robots in the narrow-nest environment. The black continuous line refers to the case of no task partitioning, the black dashed line to the case of partitioning. Gray lines indicate the 95% confidence interval on the expected value.

more dramatic way. At a certain point, the drop in efficiency becomes very steep for the non-partitioned strategy. On the other hand, the partitioned strategy scales better: individual efficiency drops smoothly. This explains why a group using the partitioned strategy performs better: it can benefit from the work of more individuals and therefore collects more prey objects. These considerations do not hold in the wide-nest environment. Figure 4b shows that the individual efficiency drops smoothly for both strategies. In addition, for group sizes $N < 33$ it is always higher in the non-partitioned strategy. Therefore, the non-partitioned strategy performs better than the partitioned strategy, as shown by the graph in Fig. 4a. In both the environments, independently of the strategy used to accomplish the task, the system collapses when the area is saturated by the swarm. Figure 5 shows the effect of an increasing number of robots on the strategy cost in the narrow-nest environment. The graph compares the cost C of each of the two strategies for different group sizes.

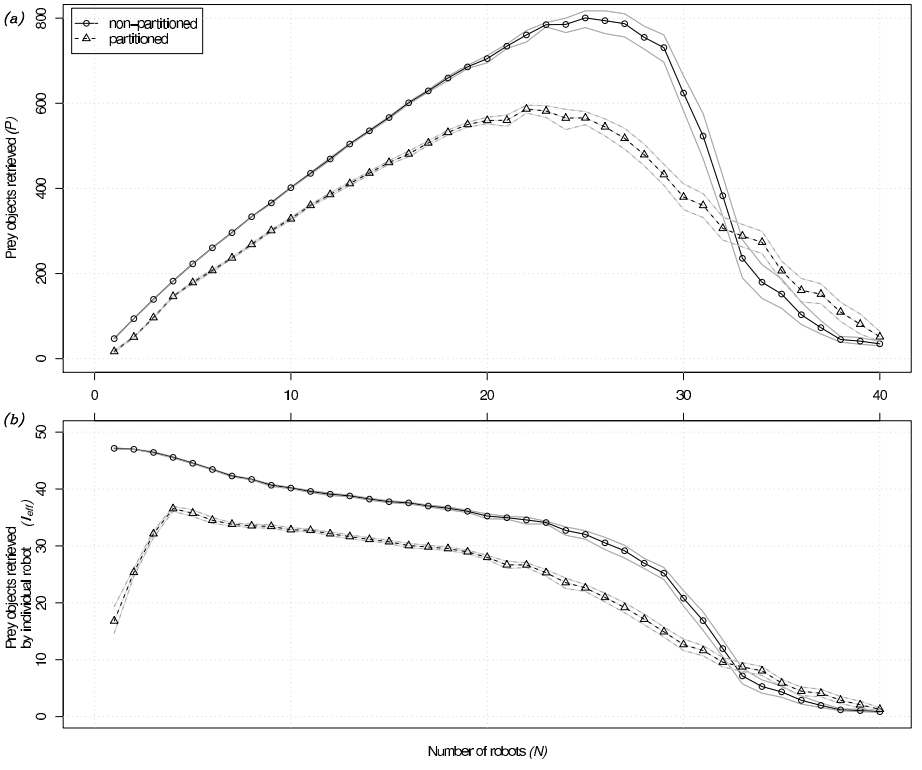


Fig. 4. (a) Performance P and (b) individual efficiency I_{eff} for increasing number of robots in the wide-nest environment. The black continuous line refers to the case of no task partitioning, the black dashed line to the case of partitioning. Gray lines indicate the 95% confidence interval on the expected value.

In case of the partitioned strategy (Fig. 5a), the graph shows each component of the cost (T_{int} and T_{part}). Clearly, task partitioning has the effect of reducing the cost due to interference but has the disadvantage of increasing the cost due to time lost. The probability of two or more robots encountering each other increases with the robot density. Although this determines a higher interference cost (i.e., T_{int}), it decreases the cost due to lower waiting time (i.e., T_{part}) in the case of the partitioned strategy. Partitioning performs better when the gain from interference reduction is greater than the loss of performance due to partitioning inefficiencies. These considerations hold in the narrow-nest environment, where the likelihood of physical interference in a task-critical zone is very high. In the wide-nest environment, interference in the nest is as likely as interference in the exchange zone. Thus, it is not beneficial to pay the cost of waiting and the non-partitioned strategy performs better for any group size. The mechanism by which partitioning reduces interference costs can be deduced by comparing the interference graphs in Fig. 6. The graphs show the number of times that physical interference (as defined in Sect. 4.5) was registered in each region of the narrow-nest environment. The total area was discretized in squares of $1\,\text{cm}^2$. Figure 6 shows the results obtained with

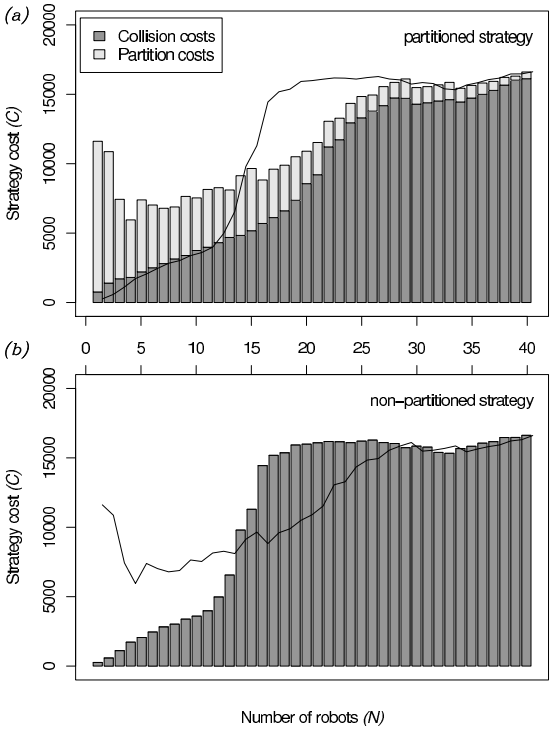


Fig. 5. Cost of interference in the narrow-nest environment. Bars represent the cost C, sum of interference time T_{int} and partition time T_{part} (i.e., waiting times). For easy reference, the outline of the bars of the respective other graph has been added to each graph. (a) Costs for the partitioned strategy, where interference cost stem from waiting times and collisions. (b) Cost in case of the non-partitioned strategy, where only physical interference through collisions exists.

18 robots, in the case of the non-partitioned strategy (Fig. 6a) and in the case of the partitioned strategy (Fig. 6b). The graphs show that the use of the non-partitioned strategy leads to high interference in the nest, which becomes congested. Partitioning the task reduces the robot density in the nest, thus spreading the interference more uniformly across the arena. In addition, the overall interference diminishes because the exchange zone is wider: the robots have more freedom of movement and collide less often. Although the graphs show only data collected with 18 robots, experiments with different group sizes produced similar results.

6 Conclusions and Future Work

Interference can be an issue when working with swarms of robots. In this work, we used task partitioning and allocation to reduce interference between robots sharing the same physical space. We manually partitioned the environment and employed a simple self-organized strategy for allocating individuals to subtasks. Results show that a partitioning strategy improves performance in a constrained environment. Additionally, we

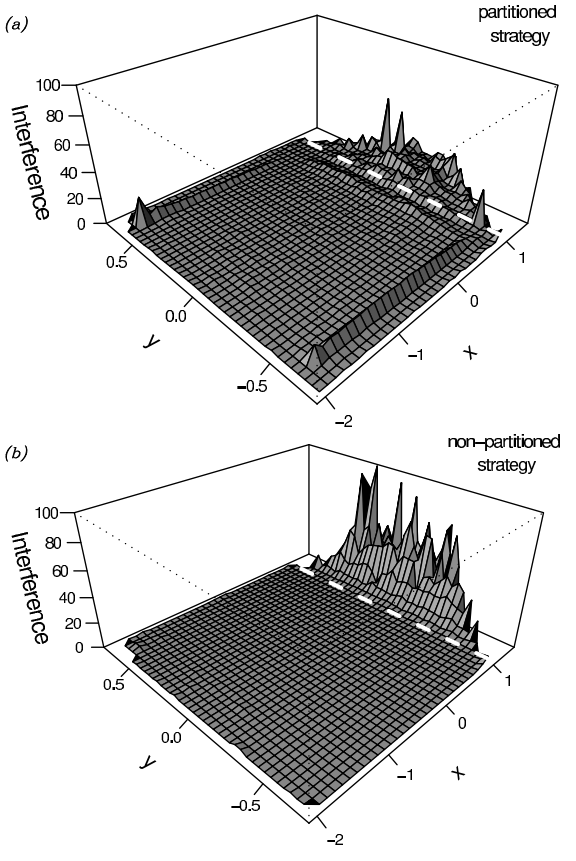


Fig. 6. Mean interference values registered for (a) the partitioned strategy and (b) the non-partitioned strategy, both in the narrow-nest environment. Shown values are observation means of 50 repetitions with $N = 18$ robots. Coordinates on the x- and y-axis are given in meters. The arena is stretched along the y-axis for better visualization. The dashed white line marks the location of the exchange zone.

identified cases in which partitioning is not advantageous and a non-partitioned strategy should be used. The proposed strategy is fairly simple and far from being an optimal solution, nevertheless we improved the performance of the swarm when interference was costly.We believe that our approach can be generalized to a wider range of applications. It can be applied to those situations in which system's performance drops because of concurrent access to a shared resource. If the accesses can be distributed more efficiently by partitioning the resource, the impact of interference can be limited. Future work will concern the identification of the optimal allocation in the studied environments as well as the development and study of a strategy that can find this optimal allocation in a self-organized and adaptive way. In addition, the interference metric proposed in Sect. 4.5 could be used by the robots to decide whether to partition the task.

In this way, we could achieve even better performance, since partitioning would be employed only when strictly needed. Finally, the goal is to validate the system using the real robots.

Acknowledgements. This work was supported by the SWARMANOID project, funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission, under grant IST-022888 and by the VIRTUAL SWARMANOID project funded by the Fund for Scientific Research F.R.S.–FNRS of Belgium's French Community. The information provided is the sole responsibility of the authors and does not reflect the European Commission's opinion. The European Commission is not responsible for any use that might be made of data appearing in this publication. Marco Dorigo and Mauro Birattari acknowledge support from the Fund for Scientific Research F.R.S.–FNRS of Belgium's French Community, of which they are a research director and a research associate, respectively.

References

1. Lerman, K., Galstyan, A.: Mathematical model of foraging in a group of robots: Effect of interference (2002)
2. Fontán, M.S., Matarić, M.J.: A study of territoriality: The role of critical mass in adaptive task division (1996)
3. Shell, D., Matarić, M.J.: On foraging strategies for large-scale multi-robot systems (2006)
4. Goldberg, D., Matarić, M.J.: Maximizing reward in a non-stationary mobile robot environment. Autonomous Agents and Multi-Agent Systems 6, 287–316 (2003)
5. Goldberg, D.: Evaluating the dynamics of agent-environment interaction (2001)
6. Lein, A., Vaughan, R.: Adaptive multi-robot bucket brigade foraging (2008)
7. Gerkey, B.P., Matarić, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems (2004)
8. Kalra, N., Martinoli, A.: A comparative study of market-based and threshold-based task allocation (2006)
9. Scheidler, A., Merkle, D., Middendorf, M.: Stability and performance of ant queue inspired task partitioning methods (2008)
10. Krieger, M.J.B., Billeter, J.B.: The call of duty: Self-organised task allocation in a population of up to twelve mobile robots (2000)
11. Labella, T.H., Dorigo, M., Deneubourg, J.L.: Division of labor in a group of robots inspired by ants' foraging behavior (2006)
12. Campo, A., Dorigo, M.: Efficient multi-foraging in swarm robotics (2007)
13. Liu, W., Winfield, A., Sa, J., Chen, J., Dou, L.: Towards energy optimization: Emergent task allocation in a swarm of foraging robots (2007)
14. Mondada, F., Pettinaro, G.C., Guignard, A., Kwee, I.V., Floreano, D., Deneubourg, J.L., Nolfi, S., Gambardella, L.M., Dorigo, M.: SWARM-BOT: A new distributed robotic concept (2004)
15. Groß, R., Bonani, M., Mondada, F., Dorigo, M.: Autonomous self-assembly in swarm-bots (2006)
16. Nouyan, S., Campo, A., Dorigo, M.: Path formation in a robot swarm. Self-organized strategies to find your way home (2008)
17. Rosenfeld, A., Kaminka, G.A., Kraus, S.: A study of scalability properties in robotic teams (2005)

Exploiting Hierarchical Probabilistic Motion Planning for Robot Reachable Workspace Estimation*

Jing Yang, Patrick Dymond, and Michael Jenkin

Department of Computer Science and Engineering
York University, 4700 Keele Street, Toronto, Canada
{jyang,dymond,jenkin}@cse.yorku.ca
http://www.cse.yorku.ca

Abstract. Given an environment and a robot, how much of the environment is reachable or accessible to the robot? This fundamental problem in robotics is known as reachable workspace estimation and is closely related to the problem of determining possible kinematic motions of a robot. For mobile kinematic structures with high degrees of freedom (DOFs) in cluttered environments, the motion planning problem is known to be NP-hard. Given the intractability of the problem, we present an efficient probabilistic method for workspace estimation based on the use of a hierarchical strategy and a probabilistic motion planner. The probabilistic motion planner is used to identify reachable portions of the workspace but rather than treating each DOF equally, a hierarchical representation is used to maximize the volume of the robot's workspace that is identified as reachable for each probe of the environment. Experiments with a simulated mobile manipulator demonstrate that the hierarchical approach is an effective alternative to the use of an estimation process based on the use of a traditional probabilistic planner.

Keywords: Reachable workspace, Probabilistic motion planning, Mobile robotics.

1 Introduction

The reachable workspace $\mathcal{W}_{reach}$ of a robot is defined as the volume or space that can be reached by a reference point of the mechanism [8]. Reachable workspace estimation is a fundamental problem in robotics as workspace properties can represent important criteria in the evaluation and design of mechanical manipulators [10,3], humanoid robots [14] and environmental layouts [13]. To take a specific example of the task, consider the problem of assessing the accessibility of an environment for a user confined to a wheelchair. For an environment to be well suited for wheelchair use it should be sufficiently clear of obstacles so that the wheelchair can navigate the environment. The process of evaluating an environment for wheelchair accessibility requires the detailed involvement of skilled clinicians and many countries provide prescriptive mechanisms (e.g., [1]) to guide this process. By reformulating this task as the problem

* This is an extended version of "Hierarchical probabilistic estimation of robot reachable workspace" which appeared in the 6th Int. Conf. on Informatics in Control, Automation and Robotics (ICINCO), Milan, Italy, 2009.

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 229–241.
springerlink.com

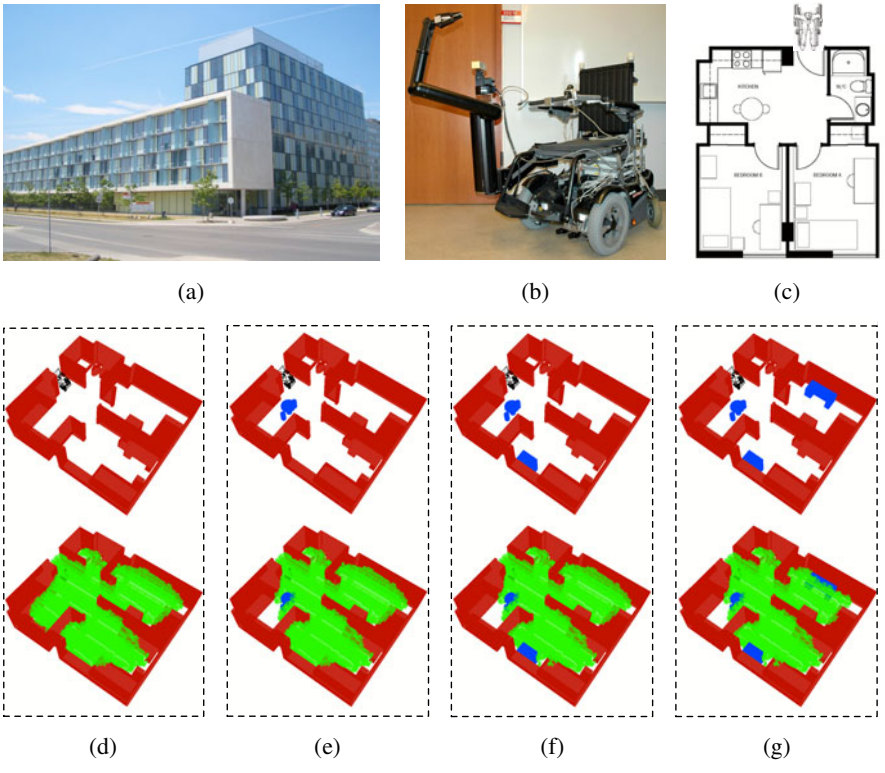


Fig. 1. Establishing accessibility for a robotic wheelchair. (a) The Pond Residence in York University; (b) The robotic wheelchair 'PlayBot' [12]; (c) For the Playbot (b) placed in the entrance of a two-bedroom apartment in building (a), how much of the environment is reachable? (d-f) Accessible regions of the apartment under different furniture layouts. The top views of (d-f) show the residence with different placements of furniture within the environment. The bottom views of (d-f) show reachable regions (colored green) for the corresponding furniture placements as determined by the algorithm presented in this paper.. Note that moving objects around the environment can lead to significant changes in the reachable portion of the environment.

of establishing $\mathcal{W}_{reach}$ for a user confined to a wheelchair it is possible to develop algorithmic solutions to the problem. Fig. 1 illustrates the problem of estimating the reachable workspace in a student apartment at York University. Student apartments consists of a number of fixed pieces of furniture (built in cabinets, etc.) and a number of movable components (tables chairs, etc.). Fig. 1(a) provides an external view of a student apartment at York University. Fig. 1(b) shows the robotic wheelchair robot (PlayBot [12]) with its manipulator. Fig. 1(c) shows the floor plan of the internal student apartment with the wheelchair placed at the entrance. Fig. 1(d)-(g) show the estimated workspace $\mathcal{W}_{reach}$ of the robotic wheelchair in the environment under different furniture placement options. Certain furniture placements restrict the motion of the wheelchair considerably.

Estimating $\mathcal{W}_{reach}$ for a mobile robot can be expressed in terms of the ability of the device to plan motions within its environment. In basic motion planning, given a robot $\mathcal{A}$ and a static workspace $\mathcal{W}$containing a set of obstacles, the objective is to determine a collision-free path between the specified start and goal for $\mathcal{A}$ A configuration c of $\mathcal{A}$ is a specification of the position and orientation of $\mathcal{A}$ in $\mathcal{W}$[9]. A configuration c is said to be free if $\mathcal{A}$ positioned at c does not collide with any obstacles in $\mathcal{W}$. The free configuration space $\mathcal{C}_{free}$ is defined as the set of all free configurations of $\mathcal{A}$ The motion planning problem can therefore be formulated in terms of computing a path in $\mathcal{C}_{free}$ between two given configurations.

Traditionally the problem of motion planning assumes that each of the degrees of freedom (DOFs) are equivalent but in estimating a robot's workspace it is important to observe that some DOFs are likely to be more important than others in terms of determining how much of the workspace is reachable. In addition, it may be possible to construct subspaces of the workspace defined by arbitrary configurations of certain combinations of joints and to use these subspaces to carve out large regions of the workspace as being reachable for each solution of the motion planning problem. Motivated by these observations we explore the use of a hierarchical structure of the DOFs of the kinematic device to establish the entire $\mathcal{W}_{reach}$. In this hierarchy we order the DOFs of the robot in terms of their predefined "importance". Then we consider corresponding sub-versions of the kinematic structure in which sub-versional joints are considered over their range of motion. Each of these sub-versions defines a reachability subspace that can be established as being reachable with a single probe into the robot's workspace. This hierarchical search mechanism can be used to enhance the probabilistic algorithms for reachable workspace estimation and to potentially carve out large regions of the space with each query within it.

This paper is structured as follows. Section 2 reviews existing algorithms for reachable workspace estimation. Section 3 outlines our hierarchical approach. Section 4 includes comparison results from applying the hierarchical approach and basic PRM to a simulated mobile manipulator. Finally Section 5 summarizes the work and provides possible directions for future research.

2 Previous Work

Although a range of techniques exist for reachable workspace estimation [3,6,14] most approaches consider the problem for robotic manipulators and do not consider arbitrary obstacles in the environment. Robotic manipulators are fixed at one end and this assumption provides certain efficiencies for reachable workspace estimation. For example, one straightforward method to compute $\mathcal{W}_{reach}$ is to take planar sections of the workspace defined by the joint angles that make up the kinematic structure and determine the contour of the section in the plane. Rotating and translating this plane based on other joints in the chain yields the three-dimensional workspace [11].

In [8] a numerical approach calculates the exact $\mathcal{W}_{reach}$ by tracing boundary surfaces of the workspace. In this approach, an imaginary force is applied to the reference point at the end-effector in order to achieve the maximum extension in the direction of the applied force. The manipulator reaches its maximum extension when the force's line of

action intersects all joint axes of rotational joints and is perpendicular to all joint axes of prismatic joints. The drawback of this algorithm is its exponential time complexity and that it only deals with manipulators that have ideal joints (without limits). A more efficient system was later developed for computing $\mathcal{W}_{reach}$ for manipulators with joint limits [2]. The system decomposed the problem into two subproblems: workspace point generation by direct kinematic based techniques and surface computation by extracting the workspace contours utilizing a subset of the points generated in the first module.

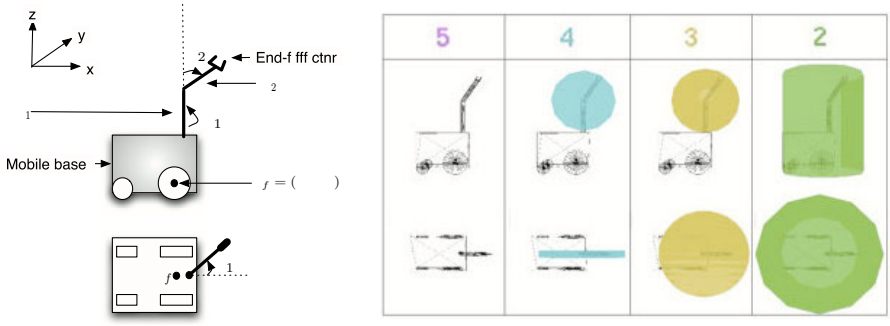
Much work has been done on capturing workspace properties for interesting kinematic structures such as human arms [10,14] and reconfigurable robotic arms [3]. Lenaric and Umek [10] developed a simplified kinematic model to represent human arm motion. $\mathcal{W}_{reach}$ was determined by calculating the reference point on the wrist for all combinations of values of joint coordinates inside the given ranges. Another approach presented by Zacharias et al. [14] discretizes the workspace into equally sized small cubes. Into each cube a sphere is inscribed and sample points on the sphere are examined using inverse kinematics. The percentage of the points on the sphere that are reachable is used to represent its level of reachability.

By reformulating the reachable workspace estimation problem in terms of the kinematic planning problem we can leverage results from the robotic motion planning literature. Motion planning emerged as a crucial and productive research area in robotics in the late 1960's [9] and its applications in real world problems continue to attract researchers from all over the world. A complete solution to the motion planning problem is known to be exponential to the robot's DOF [4]. As a consequence a number of heuristic approaches to path planning have been developed. The main difference between the probabilistic approaches and earlier complete approaches is that the probabilistic approaches do not attempt to construct an exact representation of $\mathcal{C}_{free}$. Rather they create a simplified graph that approximately "covers" $\mathcal{C}_{free}$ and captures its connectivity in reasonable time. The Probabilistic Roadmap Method (PRM) [5,7] is a popular heuristic motion planner. The algorithm first constructs a roadmap by connecting randomly sampled collision free configurations and then answers multiple queries by attempting to connect them to the roadmap.

In this paper we develop a hierarchical probabilistic algorithm based on PRM to give a proper estimation for $\mathcal{W}_{reach}$ for an arbitrary mobile device operating in a known and cluttered environment.

3 Hierarchical Reachable Workspace Estimation

For a robot $\mathcal{A}$ moving in the workspace $\mathcal{W}$, the robot's degrees of freedom (DOFs) are the minimum set of independent displacements/orientations that specify $\mathcal{A}$'s complete position and orientation in $\mathcal{W}$. Thus a configuration c of $\mathcal{A}$ with n DOFs can be specified as a set of n parameters, say $j_1, ..., j_n$, and theoretically there can be $O(n^2)$ different orderings of the joints. First we define an ordering of the DOFs such that more important DOFs have a lower index. Although there can be many definitions according to the nature of the problem, here we define the importance of a DOF by its effect on the volume of $\mathcal{A}$'s occupation in $\mathcal{W}$. This importance weight is expressed more formally below.



(a) (b)

Fig. 2. (a) $\mathcal{A}$ consists of $\mathcal{A}_{base}$ and $\mathcal{A}_{arm}$ that has two links L_1 and L_2 connected by revolute joints. The configuration of $\mathcal{A}$ is written as $(x, y, \theta, \phi_1, \phi_2)$. (b) Representations of the hierarchical occupancy of $\mathcal{A}$ are shown in different colors. The body itself also has occupancy constraints but these are not shown.

For a point a in $\mathcal{A}$, let $P_a : \mathcal{C} \rightarrow \mathcal{W}$ be the mapping that calculates the position of point a in $\mathcal{W}$ when $\mathcal{A}$ is placed at configuration c. Depending on the position of a in $\mathcal{A}$, P_a is a function of $c' \subseteq c$. We say that a is determined by a DOF j_i if $j_i \in c'$. Now we can define a weight function of a DOF j_i in terms of the volume of $\mathcal{A}$ in $\mathcal{W}$ determined by it:

Defn 1: $w(j_i) =\mid \{a \in A \mid a \; is \; determined \; by \; j_i\} \mid$.

A DOF j_x is more important than j_y if $w(j_x) > w(j_y)$. Therefore we can write the configuration c as a vector of length n in a decreasing order of their importance, say $c = (j_1, j_2, ...j_n)$, i.e. for $i = 1, 2, ...n-1, w(j_i) \geq w(j_{i+1})$. Take the 5-DOF mobile manipulator shown in Figure 2 as an example. x and y determines the entire robot, and θ determines all the portions except the rotation center of the mobile base (assume the mobile robot can rotate around its center). ϕ_1 determines links L_1 and L_2. ϕ_2 determines link L_2. Therefore, $w(x) = w(y) > w(\theta) > w(\phi_1) > w(\phi_2)$, and the ordered configuration can be written as $(x, y, \theta, \phi_1, \phi_2)$ (or $(y, x, \theta, \phi_1, \phi_2)$).

Given an ordering of the DOFs, we seek a hierarchical representation within which certain joint angles are 'free' and can assume arbitrary values within some previously defined domain. Let the domain of j_i be $\mathcal{D}_i$, $c^r = (j_1, j_2, ...j_r)$ is a subset of $\mathcal{D}_1 \times \mathcal{D}_2 \times ...\mathcal{D}_n$, given by $\{\forall x_{r+1} \in \mathcal{D}_{r+1}, x_{r+2} \in \mathcal{D}_{r+2}, ...x_n \in \mathcal{D}_n \mid (j_1, j_2, ..j_r, x_{r+1}, x_{r+2}, ...x_n)\}$. That is c^r is the set of possible configurations with joints $1...r$ having specific values but joints $r+1...n$ being free. This hierarchical concept applies to general kinematic structures in the domain of motion planning.

In order to integrate the above hierarchy into the workspace estimation process, we first perform two types of analysis to the hierarchical representations: occupation analysis and reachability analysis. Define the occupied area $OA_c(c^r)$ of c^r as the union of the occupied volume in $\mathcal{W}$ of every element in c^r, and the reachable area $RA_c(c^r)$ of c^r as the union of the reachable points of every element in c^r. Under the hierarchy nodes

with lower r occupy and reach larger workspace than those with higher r. To be precise, we have these two lemmas:

Lemma 1: $\forall i, j \in [0, n], i < j \implies OA_c(c^i) \supseteq OA_c(c^j)$. For any configuration of $\mathcal{A}$ the occupied workspace of the lower hierarchy is the superset of that of the higher hierarchy.

Lemma 2: $\forall i, j \in [0, n], i < j \implies RA_c(c^i) \supseteq RA_c(c^j)$. For any configuration of $\mathcal{A}$ the reachable workspace of the lower hierarchy is a superset of that of the higher hierarchy.

In addition, let $V(c)$ be a function that returns true iff $\mathcal{A}$ is collision free when it is at configuration c. Similarly, $V(c^r)$ returns true iff every element in c^r is valid, i.e. $OA_c(c^r)$ does not collide with any obstacle in the environment. Therefore, we have the following lemma:

Lemma 3: $\forall i, j \in [0, n], i < j \wedge V(c^i) \implies V(c^j)$. For some configuration of $\mathcal{A}$, that its lower hierarchical representation is free implies the higher hierarchical representation is free, too.

The spatial representation of both the reachability and occupancy spaces can be very complex. In practice the computation of the exact hierarchical representations is unnecessary. Conservative representations of these complex shapes can provide significant computational savings and this computation can be done prior to the execution of the motion planner. This needs to be done only once for each DOF of the robot, independent of the robot's configuration.

To establish the general representation of $OA(c^r)$ and $RA(c^r)$, we fix the first r DOFs and take all possible values of the remaining DOFs to construct the hierarchical body. Note that $RA(c^r)$ defines a subspace in the robot's workspace that is reachable by some unique configuration within c^r, and that for any two configurations within c^r there exists a continuous path between them. As an example, consider the mobile manipulator $\mathcal{A}$ shown in Fig. 2(a), let $\mathcal{D}_1 = [x_{min}, x_{max}]$, $\mathcal{D}_2 = [y_{min}, y_{max}]$, $\mathcal{D}_3 = [-\pi, \pi)$, $\mathcal{D}_4 = [-\pi, \pi)$, and $\mathcal{D}_5 = [-\pi, \pi)$. Fig 2(b) shows the hierarchical representation of $\mathcal{A}$ from level 5 down to 2, indicated by color. For simplicity we used ideal ranges $[-\pi, \pi)$ for the rotational joints of $\mathcal{A}$

The hierarchy can be integrated into workspace estimation using a probabilistic planner such as PRM. We now describe the main steps involved in the construction of a hierarchical roadmap $\mathcal{R}$ for efficient reachable workspace estimation. Nodes with large reachable areas are preferred (they establish more of the environment as being reachable for each calculation). So for each configuration c, we look for the minimum value r_{min} such that $V(c^{r_{min}})$ is true, and we call r_{min} the rank of c. The procedure described in the pseudocode in Algorithm 1 finds a random configuration and establishes its most general representation in the hierarchy. In the "for" loop from Line 4 to Line 9, the algorithm computes the rank of the node by checking collisions of the hierarchical representations of the robot's occupancy. Once the minimal valid hierarchical node

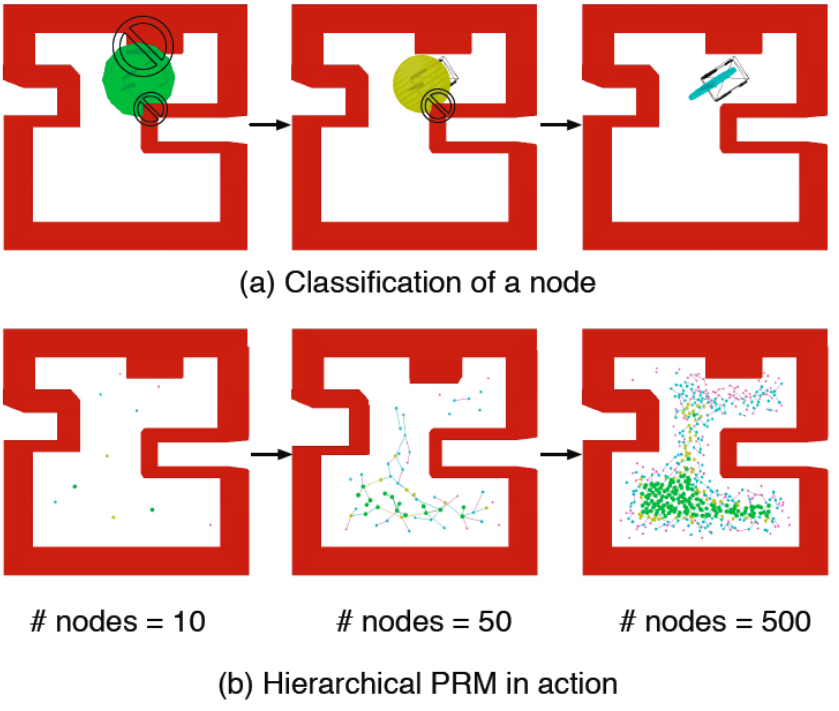


Fig. 3. Construction of $\mathcal{R}$. Upper row shows the generation and classification of one node. The rank of a node c^r is calculated by checking the hierarchical occupancy representation of $\mathcal{A}$ The lower row shows the hierarchical PRM in operation. The coverage and connectivity of $\mathcal{R}$ increases as more nodes are added. Node color encodes the classification of the node.

is established the configuration together with the computed rank is added to the set of nodes N (Line 11).

Algorithm 1: Node selection

```
1:  nodeFound ← false
2:  while ¬nodeFound do
3:      c ← a randomly chosen configuration in C
4:      for k ← 1 to n do
5:          if V(c^k) then
6:              nodeFound ← true
7:              break
8:          end if
9:      end for
10: end while
11: N ← N ∪ {c^k}
```

Whenever a new hierarchical node is found, we select a number of candidate nodes from the current set N and try to connect the new node to each of them. In addition to the connection computation performed by the traditional local planner, the rank of the edge should be established. For an edge e we look for the minimum hierarchy r_{min} such that $V(e^{r_{min}})$ is true along the edge.

The hierarchical node interconnection is built upon a local path locator and a hierarchy establisher. The local path locator returns an edge candidate, i.e. a local path that $\mathcal{A}$ can follow from one configuration to another. Then the hierarchy establisher checks if the edge candidate is collision free and meanwhile establishes the edge's most general representation in the hierarchy. The process of establishing hierarchical node interconnections is outlined in Algorithm 2. In line 3, the hierarchy r is initialized to be the maximum value of the ends of the edge. There is an obvious lemma according to the definition of the hierarchical edge connecting two nodes a^{r_1} and b^{r_2}:

Algorithm 2: Connect(a^{r_1}, b^{r_2})

```
1:  τ ← the edge candidate returned by the local path locator
2:  Discretize τ into a list of configurations τ' = (c_1, c_2, ..., c_m)
3:  r_current ← MAX(r_1, r_2)
4:  for all c_i ∈ τ' do
5:      for k ← r_current to n do
6:          if V(c_i^k) then
7:              r_current ← k
8:              break
9:          else
10:             exit and report failure
11:         end if
12:     end for
13: end for
14: E ← E ∪ {(a, b)^{r_current}}
```

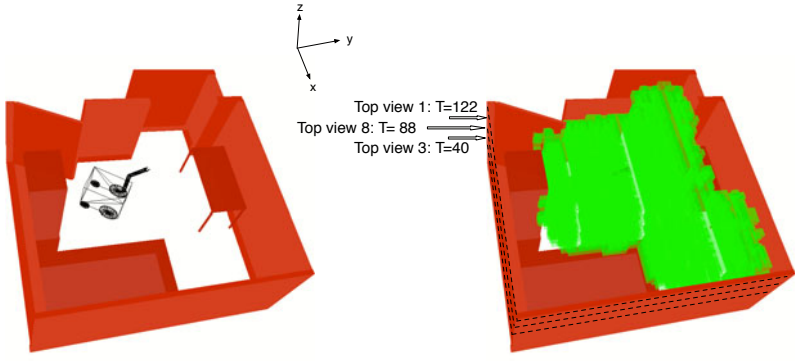
Lemma 4: $V(e^r) = true \implies r \geq r_1 \wedge r \geq r_2$, i.e. the rank of an edge is not less than the rank of either end node of the edge.

Algorithm 2 searches over the sequence of configurations on the edge for verification and hierarchy establishment. This general approach is straightforward to implement. Note that the hierarchy is established through the validation of the configurations. Similarly, to apply this strategy to other probabilistic motion planners it can be done when samples are checked for collision.

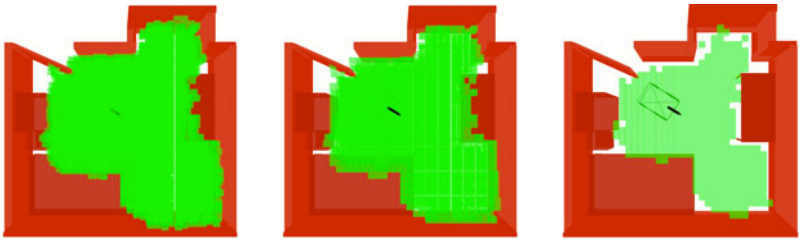
After the hierarchical roadmap $\mathcal{R}$ is built, $\mathcal{W}_{reach}$ can be computed from the connected component of $\mathcal{R}$ that contains the initial configuration of $\mathcal{A}$ through the mapping function RA. Uniform cell decomposition can be used to represent $\mathcal{W}$. Fig. 4 shows the estimated $\mathcal{W}_{reach}$ of a 3D environment.

4 Experimental Validation

We conducted experiments of our algorithm using the simulated 5-DOF mobile manipulator $\mathcal{A}$ shown in Fig. 2. First we provide an example that illustrates the algorithm



(a) 3D workspace view of the mobile robot in a room



(b) Slices through the workspace

Fig. 4. Given a mobile robot with 5 DOF entering an apartment-like environment, the estimate of the 3D reachable workspace $\mathcal{W}_{reach}$ (green area) shown in different horizontal layers

in operation (see Fig. 3). Initially $\mathcal{R}$ contains only one node that represents the initial configuration of $\mathcal{A}$ The rank of each randomly generated node is determined by looking for the most general occupancy representation of $\mathcal{A}$ that does not collide with any obstacles. Similarly the rank of each edge is determined by looking for the most general occupancy representation of $\mathcal{A}$ along the edge that does not collide with any obstacles. Fig. 3(a) shows tests for a randomly generated node. c^2 and c^3 generate collisions while c^4 does not, so this specific node is classified as c^4. The construction of $\mathcal{R}$ is incremental. Fig. 3(b) shows incremental changes in $\mathcal{R}$ as more nodes are added to the graph. As more nodes are added both coverage and connectivity of $\mathcal{R}$ increases.

Fig. 6(a) and (b) compare the hierarchical PRM and random sampling using repetitive PRM for reachable workspace estimation on the environments shown in Fig. 5(a) and (b) respectively. The x-axis represents the number of samples used in the estimation. The y-axis represents the percentage of the estimated $\mathcal{W}_{reach}$ (the true $\mathcal{W}_{reach}$ of the robot was computed by brute force search). Because randomness is involved, the workspace volume for ten independent runs for each case were averaged. Standard deviations are also plotted. Both graphs show that the hierarchical approach is more effective than repetitive PRM sampling for reachable workspace estimation.

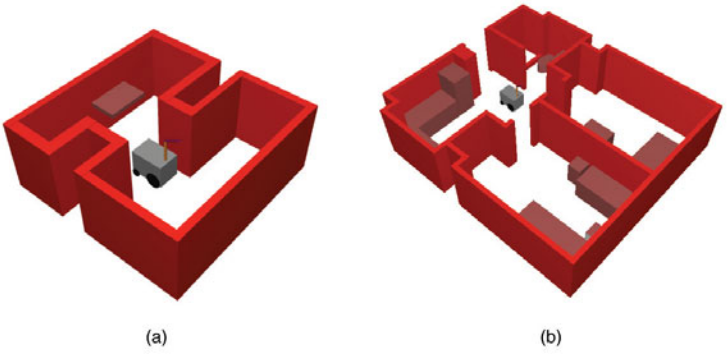
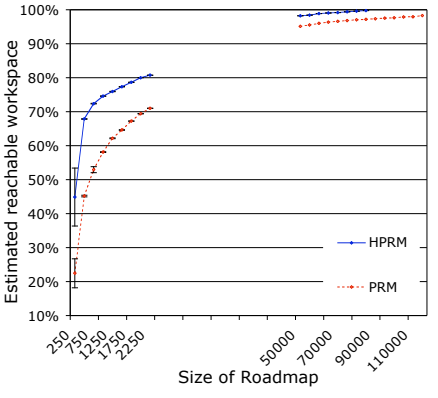
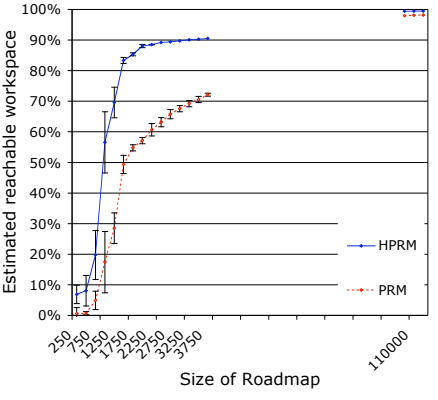


Fig. 5. Experimental setup. (a) The mobile manipulator is placed in a simple 3D environment; (b) The mobile manipulator is placed in an apartment-like environment.



(a) Experimental result for the environment shown in Fig. 5(a).



(b) Experimental result for the environment shown in Fig. 5(b).

Fig. 6. Comparisons of workspace volume computed from hierarchical PRM and repetitive PRM sampling for the environment shown in Fig. 5(a) and (b). Each data point plots the average of ten experiments. Standard deviations are plotted.

Finally we evaluate the time efficiency of our hierarchical approach. Fig. 7 compares the running time of the roadmap construction using the basic PRM and the hierarchical approach. Because randomness is involved, running times for 20 independent runs were averaged for each case. The hierarchical PRM performed reasonably well in these experiments. As can be seen from the results, creating a hierarchical roadmap takes less time than creating a traditional roadmap. This is because the hierarchical PRM saves time in collision checking in easy regions.

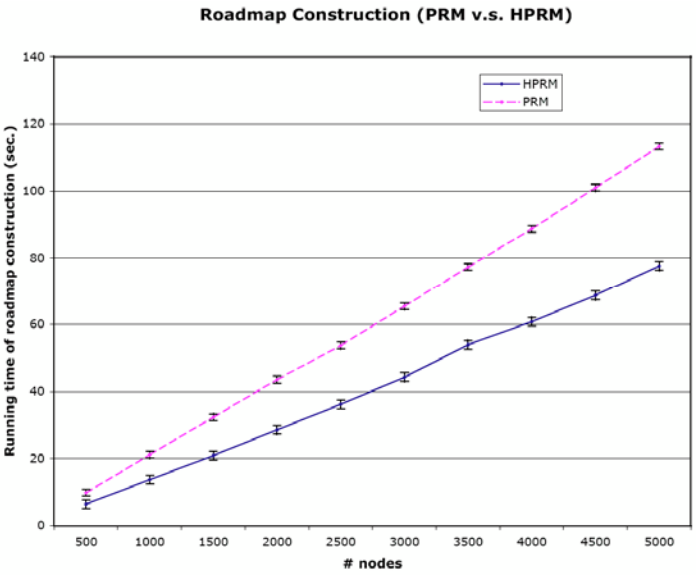


Fig. 7. Comparison of the average running time of the construction of traditional roadmaps and the hierarchical ones for the model given in Fig. 5(b). Averages are for 20 trials. Standard deviations are shown.

5 Summary and Future Work

A hierarchical approach was presented for adapting probabilistic motion planners for reachable workspace estimation. Unlike traditional probabilistic motion planners that treat each DOF equally, we order the DOFs of the kinematic structure and consider a hierarchical approach to the planning task. Considering the characteristic of the reachable workspace estimation problem, this hierarchy exploration improves the planning process through two critical computations: occupational analysis and reachability analysis. Validation of configurations begins by doing fast tests on simple occupational representations and only progresses to more accurate (and more expensive) evaluations as necessary. Because randomness is involved it is hardly possible to estimate the entire reachable workspace from the probabilistic roadmap within reasonable time. However, by iteratively computing the maximal reachable workspace from each node and edge our hierarchical motion planner can be more effective in the computation process than the traditional ones.

The hierarchical workspace estimation algorithm is especially useful for mobile robots in environments with obstacles. Experiments were conducted on a simulated 5-DOF mobile manipulator in two 3D environments. Experiments show that the hierarchical approach can be an effective and efficient alternative to the repetitive PRM for reachable workspace estimation.

Our current hierarchical algorithm uses the coarse-to-fine hierarchical nature in the process of estimating the workspace. In the examples shown in this paper, we assumed

the DOFs of the robot are ordered in terms of their predefined "importance", which is determined by its effect on the volume of the kinematic structure's occupancy in the workspace. When the kinematic structure executes a path from the start to goal, it only moves its 'less important' DOF (e.g. the attached manipulator is less important than the mobile base) when necessary. This reduces unnecessary flailing of less important DOFs (e.g. the attached manipulator) while the mobile base is moving. Motivated by this feature of the hierarchical approach, our ongoing work includes the use the hierarchical space to do path planning after the space is constructed. The goal is to address practical problems where paths should not only meet hard constraints that absolutely must be satisfied (e.g. that the path is collision-free) but meet certain soft constraints that serve as guides to 'encourage' or 'influence' the robot to behave in certain ways. While hard constraints are satisfied during the construction of the hierarchical space, soft constraints can be considered by planning within the constructed space.

For a complex robotic platform the ordering of the DOFs may not be obvious, e.g. a mobile robot with two manipulators attached and a more adaptive definition of the ordering may be required. A hierarchical characteristic might also be employed in other aspects of motion planners. For example, one heuristic would be to let the established hierarchy lead the sampling process toward the boundaries of obstacles, i.e. to sample more densely near nodes with higher hierarchy labels than those with lower hierarchy labels.

We can also imagine a more sophisticated definition of reachable workspace which might involve establishing the number of configurations from which the kinematic structure can reach a given location. This might provide insights into different levels of reachability. A space for where there exists many reachable configurations should probably be considered more reachable than one with just a few.

Acknowledgements. The financial support of NSERC Canada is gratefully acknowledged.

References

1. Access Board. Americans with Disabilities Act Accessibility Guide. U.S. Architectural and Transportation Barriers Compliance Board (1997)
2. Alameldin, T., Sobh, T.: On the Evaluation of Reachable Workspace for Redundant Manipulators. Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE) 2, 1079–1085 (1990)
3. Badescu, M., Mavroidis, C.: New Performance Indices and Workspace Analysis of Reconfigurable Hyper-redundant Robotic Arms. International Journal of Robotics Research 2, 643–659 (2004)
4. Canny, J.F.: The Complexity of Robot Motion Planning. MIT Press, Cambridge (1988)
5. Horsch, T., Schwarz, F., Tolle, H.: Motion Planning for Many Degrees of Freedom – Random Reflections at C-Space Obstacles. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA 1994), pp. 3318–3323 (1994)
6. Hsu, M.-S., Kohli, D.: Boundary Surfaces and Accessibility Regions for Regional Structures of Manipulators. Mechanism and Machine Theory 22, 277–289 (1987)
7. Kavraki, L.E., Svestka, P., Latombe, J.-C., Overmars, M.: Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces. IEEE Transactions on Robotics and Automation 12(4), 566–580 (1996)

8. Kumar, A.: Characterization of Manipulator Geometry. PhD thesis, University of Houston (1980)
9. Latombe, J.-C.: Robot Motion Planning, Cluwer (1991)
10. Lenarcic, J., Umek, A.: Simple Model of Human Arm Reachable Workspace. IEEE Transactions on Systems, Man and Cybernetics 24(8), 1239–1246 (1994)
11. Morecki, A., Knapczyk, J.: Basics of Robotics: Theory and Components of Manipulators and Robots. Springer, Wien New York (1999)
12. Tsotsos, J.K., Vergheses, G., Dickinson, S., Jenkin, M., Jepson, A., Milios, E., Nuflo, F., Stevenson, S., Black, M., Metaxas, D., Culhane, S., Ye, Y., Mann, R.: PLAYBOT: A visually-guided robot to assist physically disabled children in play. Image and Vision Computing, Special Issue on Vision for he Disabled 16, 275–292 (1998)
13. Yang, J., Dymond, P., Jenkin, M.: Accessibility assessment via workspace estimation. International Journal of Smart Home 3, 73–90 (2008)
14. Zacharias, F., Borst, C., Hirzinger, G.: Capturing Robot Workspace Structure: Representing Robot Capabilities. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3229–3236 (2007)

Defocus-Based Threedimensional Tracking in SEM Images

Christian Dahmen

Division Microrobotics and Control Engineering, University of Oldenburg
Ammerländer Heerstr. 118, 26129 Oldenburg, Germany
christian.dahmen@uni-oldenburg.de
http://www.amir.uni-oldenburg.de

Abstract. A new tracking algorithm is proposed for threedimensional object tracking. It is based on an active contours algorithm with region-based minimization for twodimensional tracking. The algorithm is extended to use the sharpness of the objects enclosed by the contour to determine an estimate for the z position. This depth-from-defocus approach is enabled through different sharpness calculations, including the grey level variance and column-wise and row-wise variance calculation, which allow to analyze the astigmatism. By this, the ambiguity of the out-of-focus displacement is resolved. The algorithm can deliver estimates for the z position of an object if the object is in a certain range around the focal point, without relying on external assumptions. The characteristic of the defocusing is determined by executing a focus sweep and calculating the defocus curve for the object of interest. The ability of the algorithm to generate threedimensional position coordinates is evaluated in a setup with nanopositioners and the principal feasibility is shown.

Keywords: Image processing, Object tracking, Depth estimation, Focus analysis.

1 Introduction

The SEM has been widely used as a imaging tool for the automated handling of micro- and nanoscale objects (see e.g. [1]). While there have been many manipulations executed manually, and with the necessary experience these manipulations have a high success rate, fully automated handling and manipulation of micro- or even nanoscale objects in the SEM still is very rarely encountered. The problems which have to be solved are manifold and not easy to handle. Integrated sensors in the actuators or the setup may deliver some information about the positions of end-effectors and tools, and enable estimates about the position of objects to be manipulated. The real actual position information of objects or tools though is difficult to recover from this data, because of various factors like thermal drift, play or object interactions on the nanoscale. A specific requirement for non-teleoperated processes is therefore the need for sensor feedback generation based on SEM images. The SEM is the only sensor which may deliver an overview over the whole scene, enabling the extraction of the positions of most or all objects engaged in the manipulation. Only the SEM determines these positions in a common coordinate system, which makes it possible to evaluate relative distances

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 243–254.
springerlink.com

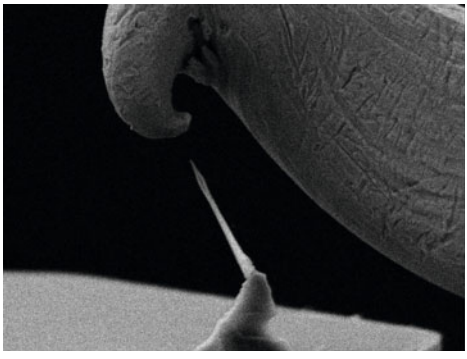


Fig. 1. A screenshot showing two objects having different z-positions. One object is a silicon nanowire, the other a deformed STM tip. The distance between the objects in z-direction is not immediately visible.

between the objects. One problem which is crucial for the success of any automation approach is the missing information about the position of objects in z-direction, which means orthogonal to the image plane. An example image illustrating the problem is shown in figure 1.

2 State of the Art

Different algorithms have been described in the literature extracting the twodimensional position of objects from SEM images for automation purposes. The performance of these algorithms is good, making first simple automation scenarios possible. The approaches used for twodimensional tracking on SEM images and their possible extensions to 3D-tracking will be summarized in the following.

One of the first and most simple approaches used template matching as the basis of the algorithm [2]. A template image is extracted or loaded which contains the object to be tracked. The template image is then cross-correlated with a search area in the input image. The maximum value of the resulting array is in the place where the template is most likely to be found. Due to the use of cross correlation, this approach is very robust against additive noise, which is an advantage especially for fast scanned SEM images. Problems of this approach are that the algorithm is sensitive against certain changes in the object appearance which may occur during handling processes. Examples for these appearance changes are rotation of the object, scaling of the object due to magnification changes and partial occlusion by other objects in the setup. Removing these weaknesses for this method comes with increased computational effort so that a fast enough calculation is not always possible. Extraction of the z-position is not featured and cannot easily be added.

If instead of a template image a CAD model of the object is available, it is possible to use rigid body models to track the object in the SEM image [3]. The implementation uses measurement lines orthogonal to the model edges and tries to fit the model to visible edges in the image. Model edges which should be invisible are identified and not

used for the pose estimation. Though edge detection is difficult in noisy SEM images, the approach yields good results using advanced techniques for discarding or outweighting false edges and through the high number of measurement lines used. When threedimensional CAD-models are used, it is possible to recover the threedimensional pose including in-plane and out-of-plane rotations, except the z-position. The extension for true threedimensional tracking relies on a model of the SEM image projection to yield the z-component of the position. This seems to be working for low magnifications.

Another possibility is the use of active contours or snakes (for details about this concept see [4]), which do not rely on much pre-existing knowledge about the object. Active contours are parametrized curves in twodimensional space, that means in the image plane. After coarse initialization the contour is evolving to segment the object from the scene. The contours are coupled with an energy function dependent on their shape or appearance, and on the image data. This energy function is being minimized by moving contour points or the contour as a whole. The part dependent on the contour is called internal energy, the part dependent on the image data is the external energy. In the original formulation, the external energy function was defined to be dependent on the distance of the contour from edges in the image, as explained in [5]. For the use with noisy SEM images, a region-based approach (see [6] and [7]) has shown to be useful. The external energy function here is dependent on the region statistics and the noise characteristics of the imaging source. The goal is to maximize the compound probability of the enclosed region. This approach has proven to be very robust to additive noise, and is inherently robust against scaling and rotation. If the contour minimization is restricted to the euclidean transform space, robustness against partial occlusion is added. Due to the model-free nature of this approach, threedimensional tracking is not immediately possible, but the coupling with focus-based methods is principally possible and shows first promising results in the SEM.

In this paper, the last tracking approach is taken as a basis, and extended to use defocus analysis for depth estimation. The extracted information is only the z-position of the tracked object, without any structural information about the object. For the recovery of threedimensional structure of objects, different methods may be used (see e.g. [8] or [9]).

3 Principle

The principle of the twodimensional tracking has been explained already in [6]. The important aspect is that the active contour algorithm does not only deliver the position information of the object, but at the same time calculates a segmentation of the object from the rest of the scene. This enables further analysis of the enclosed object.

Due to the working principle of the SEM, only a certain range around the set working distance is depicted sharp. Though this range is quite big in comparison to optical microscopes, defocusing is still evident, as can be seen in figure 2. The defocusing in the SEM has been used already in [10] to determine the z-position of objects by generating a sharpness curve over a certain working distance range. One drawback is the amount of time needed to obtain this image sequence and therefore also the disability to monitor dynamic processes. But by analyzing the sharpness of the object which is of interest, it is also possible to directly conclude to the out-of-focus displacement of the object, as has been recently demonstrated for the SEM in [11].

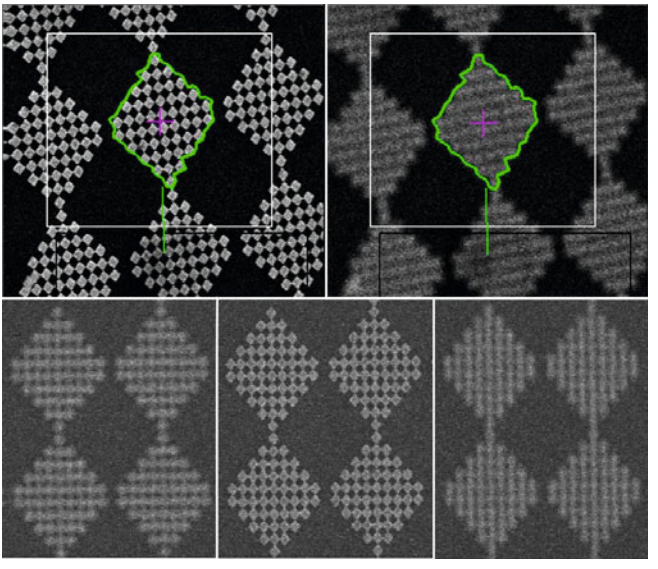


Fig. 2. Different sharpness states of objects. Top left: sharp object, top right: blurry object, bottom: influence of astigmatism on object blur.

The sharpness measure used here is the grey-level variance

$$\sigma^2(I, A(C)) = \frac{1}{N_{A(C)}} \sum_{p \in C} I(p) - \bar{I}(A(C)) \tag{1}$$

with

$$\bar{I}(A(C)) = \frac{1}{N_{A(C)}} \sum_{p \in A(C)} I(p)), \tag{2}$$

and C the contour, $A(C)$ the enclosed area and I the image.

For the object enclosed by the contour this means

$$\sigma^2(I(WD), A(C)) = Max \Rightarrow z(Object) = WD \tag{3}$$

with the working distance WD and the z-position of the Object enclosed by the contour $z(Object)$.

Also, assuming a calibration curve

$$Cal(\Delta z) = \sigma^2(I(\Delta z), A(C)) \tag{4}$$

is available which is monotonic, bijective and describes the relation between out-of-focus displacement and sharpness, we can calculate the out of focus displacement of the object from the object sharpness for each frame:

$$\Delta z = Cal^{-1}(\sigma^2(I, A(C))) \tag{5}$$

One problem which still persists after this analysis is that the out-of-focus displacement turns out to be ambiguous. Two naive possibilities exist for the solution of this problem, one assuming that the object is nearer than the working distance, the other assuming that the object is further away. This is a situation which is not optimal for automation purposes. Though in some cases the correct solution may be determined by the setup and context knowledge, it is desirable to determine the solution without additional information apart from the image. For the solution of this problem, a typically undesired effect in SEM imaging may be taken advantage of.

During normal use of the SEM, astigmatism is something that should be diminished or removed by astigmatism correction. Astigmatism in the SEM leads to blurry images for the user. An important property of astigmatism is that the sharpness is direction dependent. The focal points of the electron beam are different for two perpendicular directions, as can be seen in figure 2. If we name the two perpendicular directions w_0 and w_1:

$$WD_{w_0} \neq WD_{w_1} \tag{6}$$

To take advantage of this, we additionally to the grey level variance of the image calculate the grey level variance of the rows and columns separately by using

$$\sigma_x^2(I, A(C)) = \sum_y \frac{1}{N_{A(C)}(y)} \sum_{p(x,y)\in C} I(p) - \bar{I}(A(C), y) \tag{7}$$

with

$$\bar{I}(A(C), y) = \frac{1}{N_{A(C)}(y)} \sum_{p(x,y)\in A(C)} I(p) \tag{8}$$

and

$$\sigma_y^2(I, A(C)) = \sum_x \frac{1}{N_{A(C)}(x)} \sum_{p(x,y)\in C} I(p) - \bar{I}(A(C), x) \tag{9}$$

with

$$\bar{I}(A(C), x) = \frac{1}{N_{A(C)}(x)} \sum_{p(x,y)\in A(C)} I(p) \tag{10}$$

and C the contour, $A(C)$ the enclosed area and I the image.

The two values are normalized to their maximum, which is determined during the initialization phase (see section 4):

$$\hat{\sigma}_y^2(I, A(C)) = \frac{\sigma_y^2(I, A(C))}{max(\sigma_y^2(I, A(C)))} \tag{11}$$

and

$$\hat{\sigma}_x^2(I, A(C)) = \frac{\sigma_x^2(I, A(C))}{max(\sigma_x^2(I, A(C)))} \tag{12}$$

In this case we expect a working distance sweep to generate two slightly displaced curves like depicted on the left in figure 3, under the assumption that the object in the image has suitable structure.

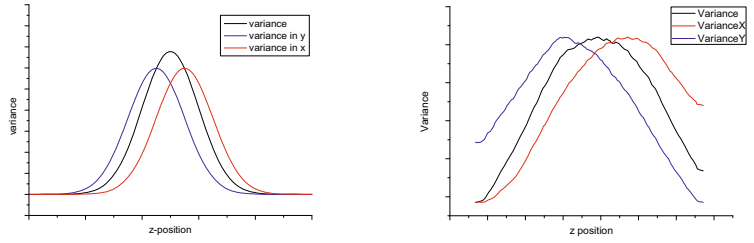


Fig. 3. Expected and measured curves for the different sharpness measures

With the two curves having slightly displaced maxima, we can estimate from the ratio of the two values

$$r = \frac{\hat{\sigma}_x^2(I, A(C))}{\hat{\sigma}_y^2(I, A(C))} \tag{13}$$

to which side the out-of-focus displacement occurs:

$$\begin{aligned} r < 1 &\Rightarrow \Delta z < 0 \\ r > 1 &\Rightarrow \Delta z > 0 \end{aligned} \tag{14}$$

4 The Algorithm

The twodimensional base algorithm is similar to the active contour algorithm described in section 2. The principle is depicted in figure 4. Important is that after an initial free minimization, the tracker does only translate and rotate the contour as a whole. This is necessary to enable the out-of-focus estimator to work, because else the enclosed area may change. The same is valid for scaling, which is therefore disabled in the algorithm.

In order to enable continuous threedimensional tracking, the algorithm has to be initialized like shown in figure 5. An initial working distance sweep is carried out to acquire the characteristic curve of the tracked object. During this sweep, the twodimensional tracking has to be enabled already. After this, certain calculations and optimizations are carried out on the acquired curves, e.g. it is made sure that the curves are monotonic in sections. After the data has been processed, the tracker can track continuously until either object changes or imaging changes require a reinitialization.

The threedimensional tracking itself consists of the twodimensional tracking algorithm, augmented with a sharpness calculation component, an out-of-focus displacement estimator and a direction estimator, like shown in figure 6. The active contour tracker delivers the twodimensional position and the segmentation of the object. This segmentation is then used to mask the original image. From the masked image, the object sharpness is calculated using variance calculation and the directional sharpness measures mentioned in the last section.

The variance value is used to estimate the out-of-focus displacement by comparison with the data acquired during initialization.

The directional sharpness values are used to estimate on which sidelobe of the initially acquired sharpness curve the object is in the actual image. After this is known, the

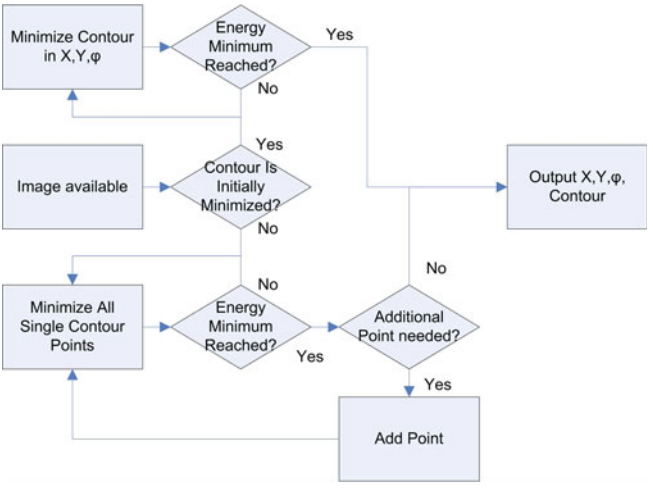


Fig. 4. The basic tracking algorithm for the twodimensional part

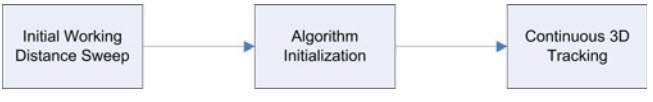


Fig. 5. The initialization steps for the threedimensional tracking

information about the displacement value and the displacement direction is combined with the working distance at which the current image was captured. The result is the estimated working distance at which the object is placed. This is then joined with the position information from the twodimensional tracking to generate a complete coordinate set.

5 Experiments

For the evaluation of the algorithm, the performance has been tested in a setup inside the SEM. The first target object was a chessboard pattern as seen left in figure 8. This chessboard pattern has been put on a stub which is mounted to a XYZ positioning setup using piezo slip-stick actuated axes with internal position sensors. For the experiment, the internal sensors of the axes were read out and the acquired data joined with the tracking data. During the experiment the axes were moved in closed loop mode using the internal sensors. The movement was in a pattern to verify the algorithm performance. All measurements have been executed at a magnification of 800x and a scanspeed of 5 (frame averaging of 2^5 frames) on a LEO scanning electron microscope. The twodimensional pattern of the movement has a size of $21\mu m$ times $39\mu m$. The pattern has been repeated at the z-axis positions of $0.17mm$, $0.256mm$ and $0.32mm$.

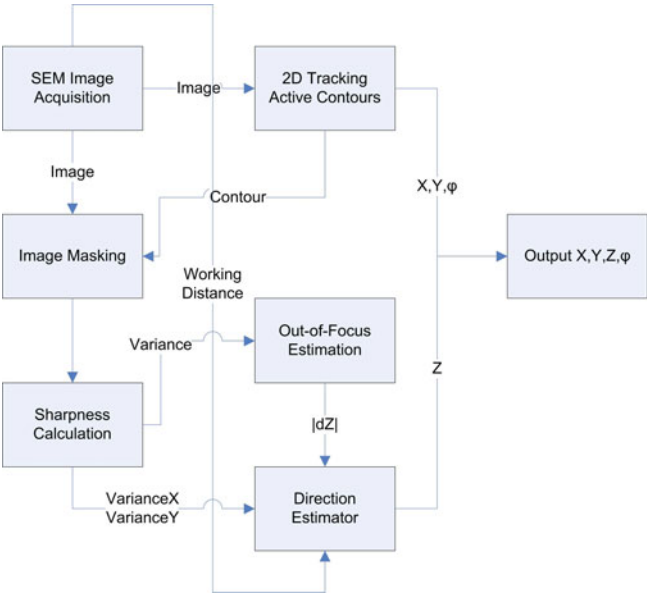


Fig. 6. The complete 3D tracking algorithm in the initialized state

For a second experiment, the setup seen in figure 7 is used. Here, a gripper is attached to a positioning setup with piezo slip-stick actuated axes. Similarly, the internal sensors of the axes were used to obtain the reference data.

6 Results

The first result was the generation of the expected curves from left in figure 3. The result can be seen on the right in figure 3. The shift of the variance calculated in rows and columns separately is clearly visible, which enables the algorithm to work in the anticipated way, estimating the direction of the defocusing.

As can be seen in figure 10 left, the movement pattern used incorporated movement along each axis as well as in diagonal. Also visible is a slight distortion of the shape in comparison to the desired shape. The reason of this may be a slightly decalibrated sensor, according to further investigation. Therefore the positions are not perfectly set.

In figure 9 this gets more obvious. While the tracking x position in figure 9 relatively closely follows the sensor information, this is not the case for the y position. Apart from the slightly decalibrated sensor it can be stated that the twodimensional tracking is working and functional, and the dependence of the measured position on the set position is roughly linear.

The threedimensional component of the position is shown in figure 10 right over the x-position. Visible is a slight dependence of the measured z-position from the x-position. One factor in this may be the orientation of the setup, but due to the fact that the dependence is not equal at different z-positions, it may be concluded that this

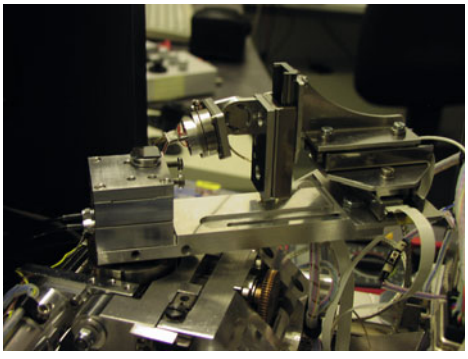


Fig. 7. The manipulation setup used inside the SEM

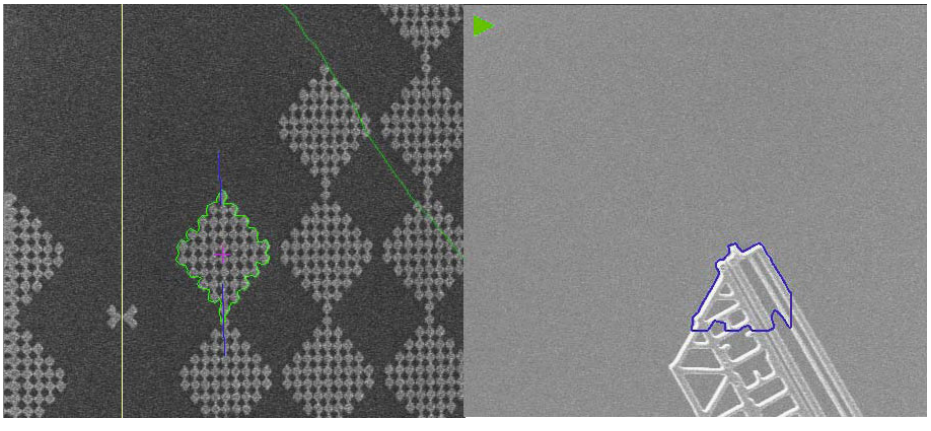


Fig. 8. The structures used for the experiment, a chessboard pattern mounted on a XYZ positioning setup, and a gripper attached to the same setup. Visible is the active contour tracking and segmenting one chessboard block and part of the gripper respectively.

effect could be generated by image characteristics changing slightly when the position is changed. This cannot be easily avoided, due to the complex setup and the principle of SEM image formation.

Figure 11 shows the determined z-Position from the tracking algorithm. Visible is that there is a certain systematic error in the tracked position, as the calculated values deviate from the expected values. The set working distance of the SEM is in the middle of the working distance range shown in figure 11, at the value 0.28. The problem which occurs here is that the algorithm is most accurate not in the point of maximal focus, but within a certain range on the sidelobes of the sharpness curve. As can be seen in figure 3, the sharpness curve is relatively flat around the maximum value. In this interval around the maximum, small changes in detected sharpness, which may also occur due to time variant behavior of the SEM imaging process or due to certain changes in the surrounding setup, will result in large errors in the estimated out-of-focus displacement.

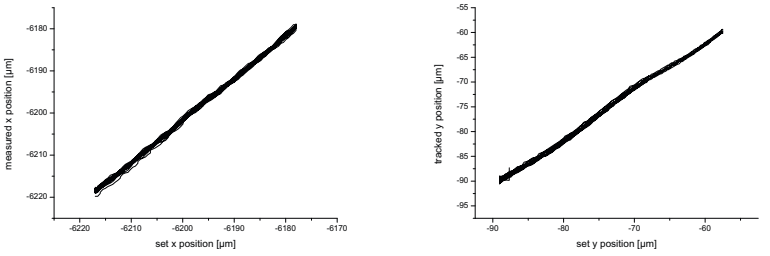


Fig. 9. The tracked x-position and y-position over the set x-position and y-position

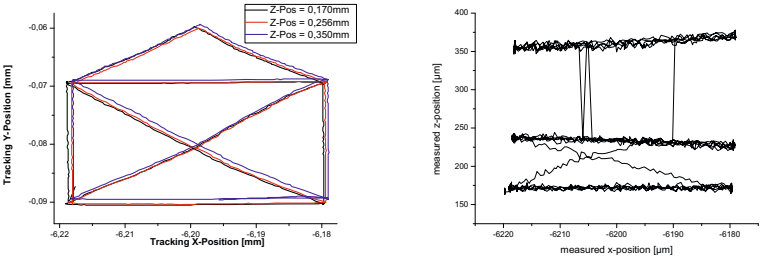


Fig. 10. Tracked X-Y position of the chessboard pattern in the repeated measurement

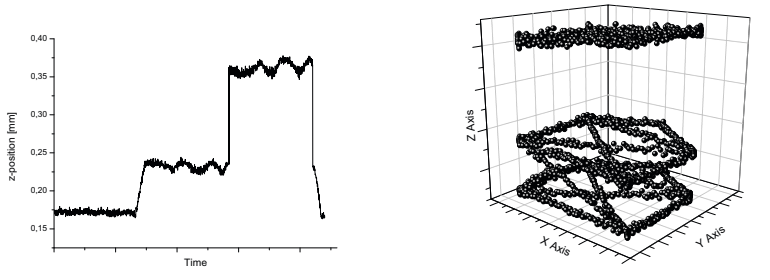


Fig. 11. The tracked z-position over time and the 3D trajectory

This explains also not only the deviation from the axis set value, but also the large amount of variation during the movement on the same z-position. So it has to be stated that the optimal working condition for this algorithm is a slightly defocused image.

In figure 11 on the right, the tracking result can be seen in three dimensions. The movement pattern is qualitatively visible, though the tracking in z-direction is not as good as in the image plane. Still the goal of the algorithm design has been reached, an estimate has been calculated for the z-position of the object which is principally useable.

In figure 12, the result of the tracking of the gripper is shown. Clearly visible here is that the structure tracked on the gripper may not be perfectly suited for the 3D-tracking. The chessboard in comparison has a certain variance in any direction, while here the structure tracked does have more direction dependence. Additionally,

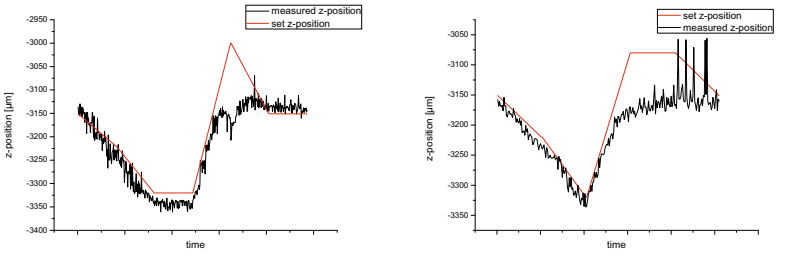


Fig. 12. The set and measured z-positions of the gripper during two tracking sessions

the structure contributing to the sharpness calculation is mostly the holes in the gripper, which are relatively few and big compared to the tracked area. This leads to suboptimal measurement curves. despite the errors which occur in both curves, it can be stated that threedimensional tracking is possible and working.

7 Summary

In this paper a threedimensional tracking algorithm for the tracking of objects in the SEM has been presented. The algorithm takes advantage of the image defocusing which is evident when objects leave the focal plane. A twodimensional tracking algorithm based on active contours with a region-based minimization has been taken as the base algorithm. Added extensions include the segmentation of the object and the consecutive sharpness calculation. Additionally the variance of the rows and columns is calculated separately for determining the direction of the defocusing. This enables the analysis of the sharpness in different directions. If the image contains astigmatism, it is possible to estimate if the image focal plane lies in front of or behind the object. Experiments have shown that this approach is working and after an initialization phase qualitatively delivers a threedimensional position information. The z-position still contains a systematic error, which is most influential around the best focused point. This error has to be diminished by further analysis and change and optimization of the implementation. Overall the feasibility of this threedimensional tracking algorithm has been shown.

References

1. Fatikow, S.: Automated Nanohandling by Microrobots (2007)
2. Sievers, T., Fatikow, S.: Real-time object tracking for the robot-based nanohandling in a scanning electron microscope. Journal of Micromechatronics 3, 267–284 (2006)
3. Kratochvil, B.E., Dong, L.X., Nelson, B.J.: Real-time rigid-body visual tracking in a scanning electron microscope. In: Proc. of the 7th IEEE Conf. on Nanotechnology (IEEE-NANO2007), Hong Kong, China (2007)
4. Blake, A., Isard, M.: Active Contours. Springer, Heidelberg (2000)
5. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision 1, 321–331 (1988)

6. Sievers, T.: Global sensor feedback for automatic nanohandling inside a scanning electron microscope. In: Proc. of 2nd I*PROMS NoE Virtual International Conference on Intelligent Production Machines and Systems, pp. 289–294 (2006)
7. Sievers, T.: Echtzeit-Objektverfolgung im Rasterelektronenmikroskop. PhD thesis, University of Oldenburg (2007)
8. Fernandez, J., Sorzano, C., Marabini, R., Carazo, J.: Image processing and 3-D reconstruction in electron microscopy. IEEE Signal Processing Magazine 23, 84–94 (2006)
9. Jähnisch, M., Fatikow, S.: 3d vision feedback for nanohandling monitoring in a scanning electron microscope. International Journal of Optomechatronics 1, 4–26 (2007)
10. Eichhorn, V., Fatikow, S., Wich, T., Dahmen, C., Sievers, T., Andersen, K., Carlson, K., Bøggild, P.: Depth-detection methods for microgripper based cnt manipulation in a scanning electron microscope. Journal of Micro-Nano Mechatronics (2008) (accepted)
11. Dahmen, C.: Focus-based depth estimation in the sem. In: Proceedings of the International Symposium on Optomechatronic Technologies, vol. 7266, SPIE (2008) 72661O

Real Time Environment Representation in Driving Scenarios Based on Object Delimiters Extraction

Andrei Vatavu, Sergiu Nedevschi, and Florin Oniga

Computer Science Department, Technical University of Cluj-Napoca
26-28 G. Baritiu Street, Cluj-Napoca, Romania
{Andrei.Vatavu,Sergiu.Nedevschi,Florin.Oniga}@cs.utcluj.ro

Abstract. In this paper we present and evaluate several methods for real-time environment representation by extracting object delimiters from the traffic scenes using a Dense Stereovision System. The delimiters detection is based on processing the information provided by a 3D classified occupancy grid obtained from the raw dense stereo information. One of the problems in representing the environment through the occupancy grid is a large volume of data. Therefore we propose a more compact 2.5D model by representing the environment as a set of polylines with associated features. Two approaches to extract object delimiters are presented: an improved contour tracing called 3A Tracing and a polyline extraction method through the radial scanning of the occupancy grid. We discuss the advantages and drawbacks for each of these methods.

Keywords: Environment Representation, Contour Tracing, Border Scanning, Environment Perception, Polyline Extraction, Object Delimiters.

1 Introduction

In the context of in-vehicle navigation systems, the environment perception and its convenient representation is an important requirement [1]. The process of environment representation building has to be accurate and characterized by a low computational cost.

Usually, the Driving Assistance Applications detect the objects through 2D or 3D points grouping processes. The detected objects are represented by geometric primitives such as 2D bounding boxes [2] or 3D cuboids [3]. As an alternative approach, the objects may be represented by polylines. One of the advantages of the polyline based objects representation is the close approximation of the object contour by the polygonal model while having a number of vertices as small as possible. In the same time the polyline could inherit the type, position and height properties of the associated object.

The polyline object representation may lead to the creation of subsequent algorithms that are computationally fast due to the fact that only a small subset of points is employed.

The road feature identification through the object delimiters detection can be used in the unstructured environments as an alternative solution to the lane detection algorithms.

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 255–267.
springerlink.com

The object delimiters extraction is studied in some areas like mobile robots [4], [5], [6], [7], [8] or autonomous vehicle systems [9], [10], [11]. The polyline representation is very common in many algorithms, such as localization and mapping [6], [7], [8], [10] contour tracking [12] and path planning [10].

The polyline extraction methods differ by the nature of the information as well as by the sensors used for data acquisition process. Current systems use laser [4], [8], [9], [10], sonar [11], [7] or vision sensors [11].

Two main directions can be distinguished for the delimiters extraction:

- **The contour processing of already detected objects from the scene** [13];
- **The radial scanning of the environment**. This method is common for the systems based on sonar or laser sensors [4], [9].

A method for map representation as a set of line segments or polylines is described in [7]. An occupancy grid is created here from sonar information. The data is converted to a list of vertices using the Douglas Peucker line reduction algorithm.

In [8] a method that learns sets of polylines from laser range information is presented. The polylines are iteratively optimized using the Bayesian Information Criterion.

The polyline representation was chosen in [10] for terrain-aided localization of autonomous vehicle. The new range data obtained from the sensor are integrated into the polyline map by attaching line segments to the end of the polyline as the vehicle moves gradually along the tunnel.

In this paper we present and evaluate several methods for real-time environment representation by extracting object delimiters from the traffic scenes using a Dense Stereovision System [3]. The delimiters detection is based on processing the information provided by a 3D classified occupancy grid obtained from the raw dense stereo information. One of the problems in representing the environment through the occupancy grid is a large volume of data. Therefore we propose a more compact 2.5D model by representing the environment as a set of polylines with associated height features. We present two approaches to extract object delimiters:

- **The 3A Tracing.** The classical algorithm for contour tracing is improved by developing a new method named 3A Tracing Algorithm;
- **The radial scanning of the occupancy grid.** We have developed a Border Scanning method that is able to detect delimiters of complex objects taking into account the nature of information from the traffic scene (curb, object, and road).

A polyline map is generated as the result of the delimiters extraction process. Each polyline element inherits the type (object, curb), position and height properties of the associated objects from the occupancy grid.

In the next section, we describe the proposed Delimiters Extraction architecture. The delimiters detection approaches are presented in section 3. Experimental results are given in section 4, and section 5 concludes the paper with final remarks.

2 System Architecture

Our delimiters detection approaches have been conceived for an urban driving assistance system. We extended our Dense Stereo-Based Object Recognition System (DESBOR) by developing an Object Delimiters Detection component. A detailed description about the DESBOR system is presented in [3].

The Object Delimiters Detection system architecture consists in the following modules (see Fig. 1):

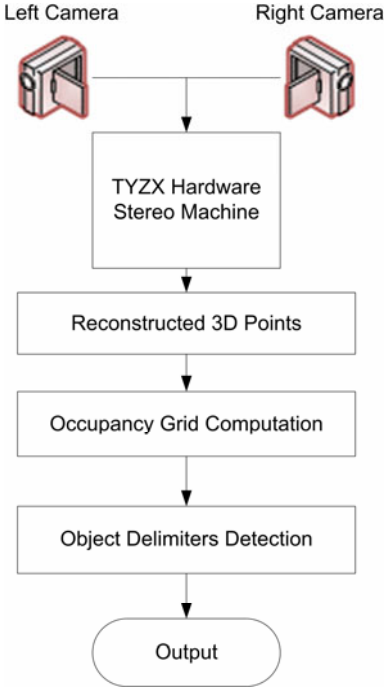


Fig. 1. System Architecture

TYZX Hardware Stereo Machine. The 3D reconstruction is performed by the "TYZX" hardware board [14].

Reconstructed 3D Points. The reconstructed 3D points are used for the occupancy grid generation.

Occupancy Grid Computation. The occupancy grid (see Fig. 2.c) represents a description of the scene, computed from the raw dense stereo information represented as a digital elevation map (see Fig. 2.b). The occupancy grid cells are classified into road, traffic isle and object cells. A detailed description about the occupancy grid computation is presented in [15].

Object Delimiters Detection. The Object Delimiters detection uses the occupancy grid results as the input and generates a set of unstructured polygons approximated

with the objects contour. The delimiters can be extracted from the occupancy grid through both 3A Tracing and Border Scanning algorithms.

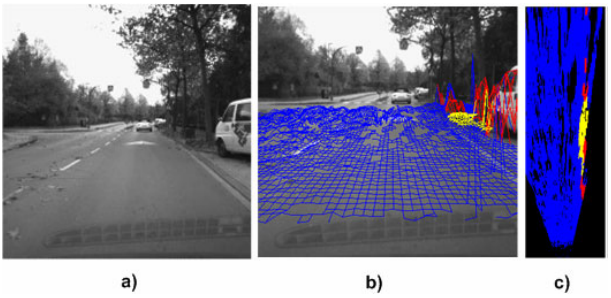


Fig. 2. The Occupancy Grid (c) is computed from the Elevation Map (b) of a scene (a). The occupancy grid cells are roughly classified (blue – road, yellow – traffic isle, red – obstacles).

Object Delimiters Detection Output. A polyline map is generated as the result of delimiters extraction process. For each polyline element we keep the following information: a list of vertices, the delimiter type (object, curb), and the height of the object for which we apply the polyline extraction.

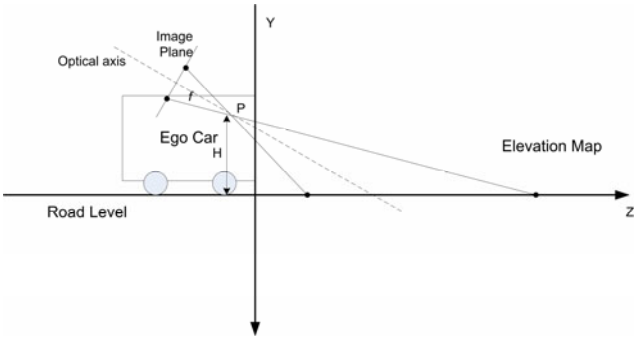


Fig. 3. The car coordinate system

It must be noted that the car coordinate system coincide with the world coordinate system having its origin on the ground in front of the car (see Fig. 3). The position and orientation of the stereo cameras are determined by the absolute extrinsic parameters [16].

3 Object Delimiters Extraction Methods

A set of steps have been identified for the delimiters extraction:

Step 1: Object Labeling. In this step each object from the occupancy grid is labeled with a unique identifier.

Step 2: The Contour Extraction. We compute the contours of the non-drivable blobs (objects, traffic isles) from the occupancy grid. Each contour point will represent a single cell in the grid map.

Step 3: The Polygonal Approximation. Given a curve C we will find a polygon that closely approximates C while having as small a number of vertices as possible.

Next, we will present several algorithms developed by us for delimiters extraction. All these methods have in common the 1st and 3rd step. The 2nd step is different in each case. We have used two main approaches for the contour extraction:

1) The Contour Tracing for a given object - once an object cell has been identified, contour tracing is performed starting from this point, adding each traversed cell to the current contour. In this paper we present an improved version of contour tracing, the 3A Tracing Algorithm.

2) The Border Scanning – a radial scanning is performed with a given radial step, traversing the interest zone and accumulating the contour points at the same time. The main difference of this approach is that we scan only the visible parts from the ego-car position. Two main improvements of the Border Scanning method are discussed: the Border Scanning using a variable step, and the Combined Border Scanning, taking into account the occupancy grid blob's nature (traffic isles, obstacles).

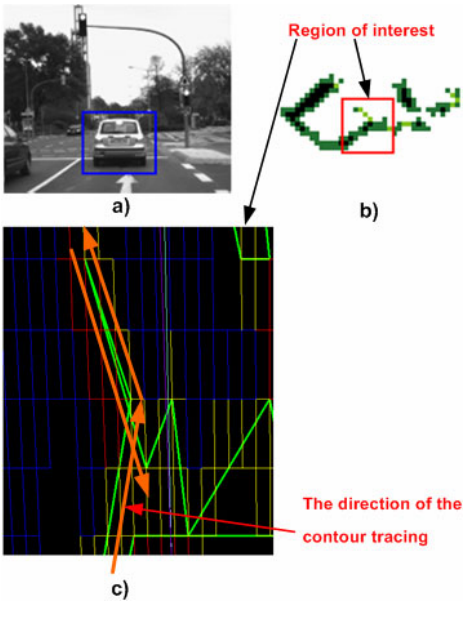


Fig. 4. Contour tracing of the care points (b) from the scene (a). There are cases when two polygonal segments can intersect each other (c), after the polygonal approximation of the car contour.

3.1 The 3A Tracing Algorithm

The classical contour tracing algorithm collects the contour points of an object by traversing the object boundary.

A disadvantage of the classical algorithm is that there are cases when the same delimiter point can be passed many times. This may lead to the incorrect representation, after the contour approximation step (see Fig. 4).

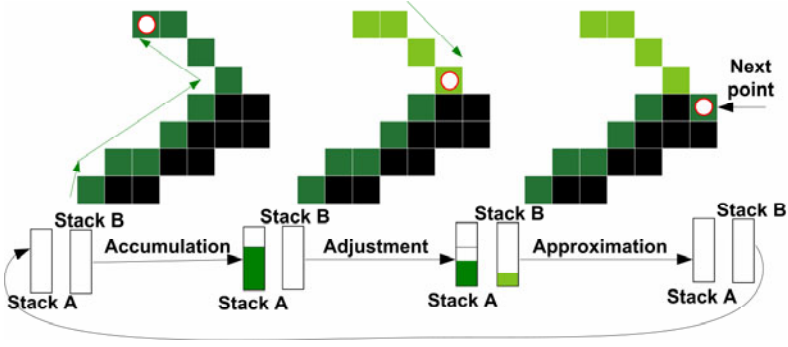


Fig. 5. 3A Tracing Algorithm. In the *Accumulation* phase, all traversed points are pushed onto the *Stack A*. In the *Adjustment* stage, the already passed points are extracted form the *Stack A* and pushed onto the *Stack B*. *Polygonal approximation* is applied in the last step of algorithm.

To avoid this problem we have developed an extended contour tracing algorithm named 3A Tracing. In this method we use two stacks, Stack A and Stack B. The name 3A Tracing comes from the next three main phases (see Fig. 5):

Phase 1: Accumulation. The tracing is made analogue to the Contour Tracing algorithm. All accumulated points are pushed onto the stack A. The traversed points are marked with a flag in order to know whether they were traversed or not at least one time. Once we found a terminal point (from which the tracing is made in the inverse sense) we pass to the 2nd phase of the algorithm.

Phase 2: Adjustment. In this phase the tracing continues in the inverse sense by extracting already passed points (drawn with light green) from the Stack A, and pushing them onto the Stack B. The Adjustment is repeated until we reach a contour point that has not passed yet. Once the new contour point is found we pass to the 3rd phase of the algorithm.

Phase 3: Approximation. Polygonal approximation is applied to each of the two stacks. After the polygonal approximation process the two stacks will be cleared and the algorithm is repeated from the Phase 1.

The algorithm stops when the start point is reached once again.

Although the 3A Tracing algorithm eliminates some particular cases in which two polygonal segments may intersect, like in the Contour Tracing, it works only on the connected components. Therefore this method does not take into account the cases of more complex objects, when a single obstacle is represented as many disjoint patches.

Therefore we have elaborated an extraction method through the radial scanning of the Elevation Map.

3.2 The Border Scanner Algorithm

The Border Scanner algorithm performs a radial scanning with a given radial step. The scanning axis moves in the radial direction, having a fixed center at the Ego Car position. The scanning process is made into the limits of *Q_from* and *Q_to* angles, thus only the interest area are scanned, where the delimiters can be detected (see Fig. 6). Having a radial axis with a *Qrad* slope, *Q_from* < *Qrad* < *Q_to*, we try to find the nearest point from the Ego Car situated on this axis. In this way, all subsequent points will be accumulated into a *Contour List*, moving the scanning axis in the radial direction. At each radial step we verify that a new object has been reached. If a new label has been found then the polygonal approximation on the *Contour List* points is performed. The list will be cleared, and the algorithm will be continued finding a new polygon.

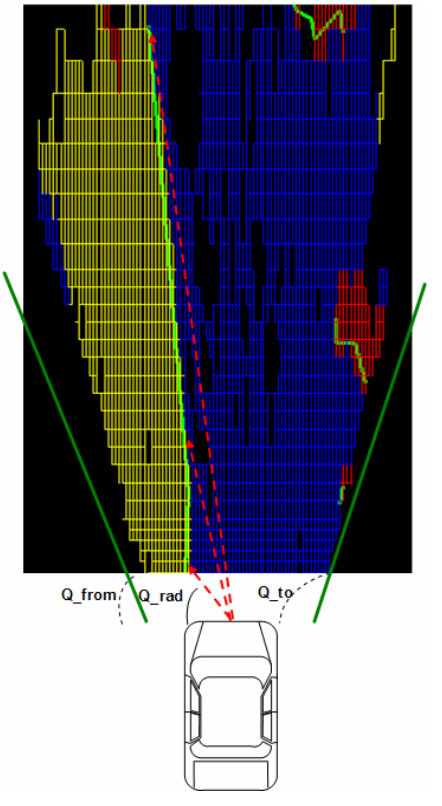


Fig. 6. Border Scanning on the Occupancy Grid Points

Advantages: The obtained results are more close to the real obstacle delimiters from the scene. The problem of the complex objects presented in the case of Contour Tracing algorithms is eliminated. Therefore many disjoint patches that belong to the same object can be enveloped by a single delimiter.

Disadvantages: A little obstacle (noise present in the occupancy grid) can occlude a great part from the scene, if this obstacle is too near to the Ego-Car. The scanning is influenced by the presence of such false obstacles.

The Border Scanning Algorithm using Variable Step. Having a constant radial step, the detected pixel density will decrease with the depth distance. The distance between two consecutive detected pixels is greater at the far depths. The idea is that some important information about the delimiters can be lost at the far distances.

A good solution is to use a scanning method with a variable step, thus the radial step should be adapted with the distance.

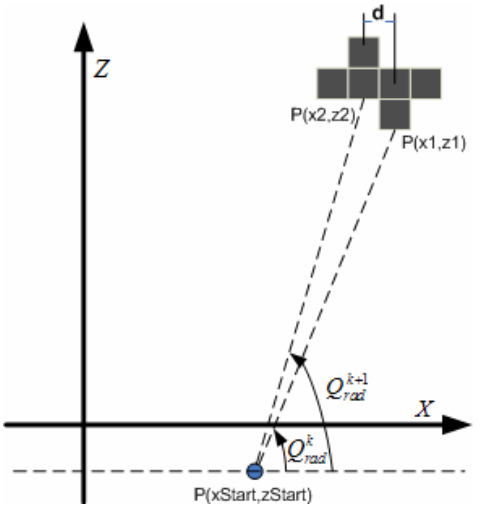


Fig. 7. Radial angle estimation for the next step in the Variable Step Border Scanning approach

If we have a point $P_1(x_1, z_1)$ of a given object and a radial axis containing the point P_1 with a radial angle Q^k_{rad} at the k step, then we estimate the radial angle at the $k+1$ step (see Fig. 7):

$$Q^{k+1}_{rad} = \arctan\left(\frac{z_1 - z_{Start}}{x_1 - x_{Start} - d}\right). \quad (1)$$

Where:

- x_1, z_1 are the coordinates of the P_1 point;;
- x_{Start}, z_{Start} are the ego-car point coordinates.
- d is considered the distance between any two adjacent points.

However, there are situations when no object point can be reached on the current scanning axis. Therefore we cannot estimate the radial angle for the next step, because we don't know the distance of the current object point from the Ego-Car. In this case, like in the simple Border Scanning method, we use a fixed step, until a new object point will be found.

The Combined Border Scanning. We know that the occupancy grid cells are classified into obstacles (cars, pedestrians etc.) or traffic isles (road-parallel patches). If we take into account only the first nearest point from the car, many relevant objects delimiters may be omitted. For example, the first obstacle from the car can be a curb. In this case, we are interested not only in the curb delimiters but also in the delimiters above the curb or behind the curb. Therefore we extended our Border Scanning algorithm by developing a method that takes into consideration the obstacle's nature making a decision based on two types of information "What have we found?" and "What we have to find?". The algorithm consists in two passes: one for the object delimiters detection, and second for the traffic isles delimiters detection.

In the Table 1 is presented the returned result when we want to find a delimiter taking in account the point type we have found.

Table 1. The Combined Border Scanning method. The result is returned, taking into consideration the found point type.

Delimiter's type we want to find	Point Type we have found	Returned result
OBJECT	OBJECT	FOUND
CURB	OBJECT	NOT FOUND
CURB	CURB	FOUND

4 Experimental Results

For the experimental results we have tested a set of 15 scenarios from the urban traffic environment using a 2.66GHz Intel Core 2 Duo Computer with 2GB of RAM.

Fig. 8 shows a comparative result between the Contour Tracing and 3A Tracing algorithms, using an approximation error of two points. One can notice that the polygonal segment intersection in the case of classical contour tracing algorithm (see Fig. 8.b) was eliminated by applying the 3A Tracing algorithm (see Fig. 8.c).

The difference between the result of delimiters detection in the case of Simple border Scanner and Combined Border Scanner algorithms is presented in the Fig. 9. It can be observed that in the case of Combined Border Scanner (see Fig. 9.b) the side fence's delimiter is detected in spite of his position behind the curb (see Fig. 9.c).

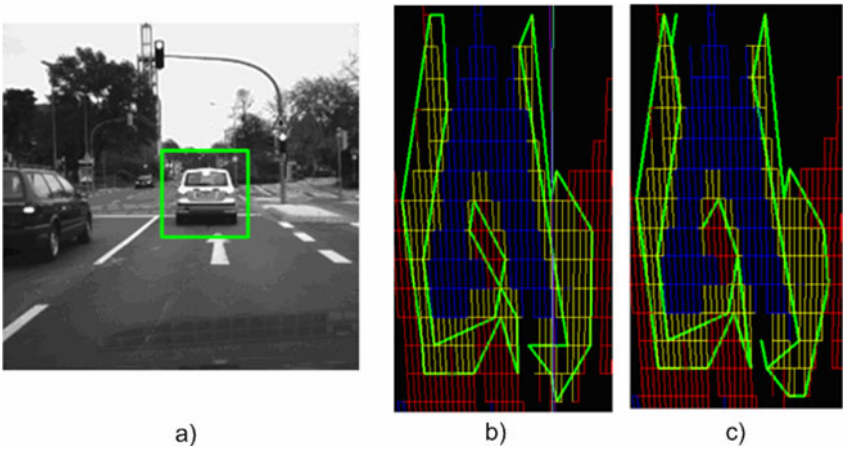


Fig. 8. Delimiters detection through the *Contour Tracing* algorithm (b) and *3A Tracing* algorithm (c). The detection is performed on the occupancy grid computed from the scene (a).

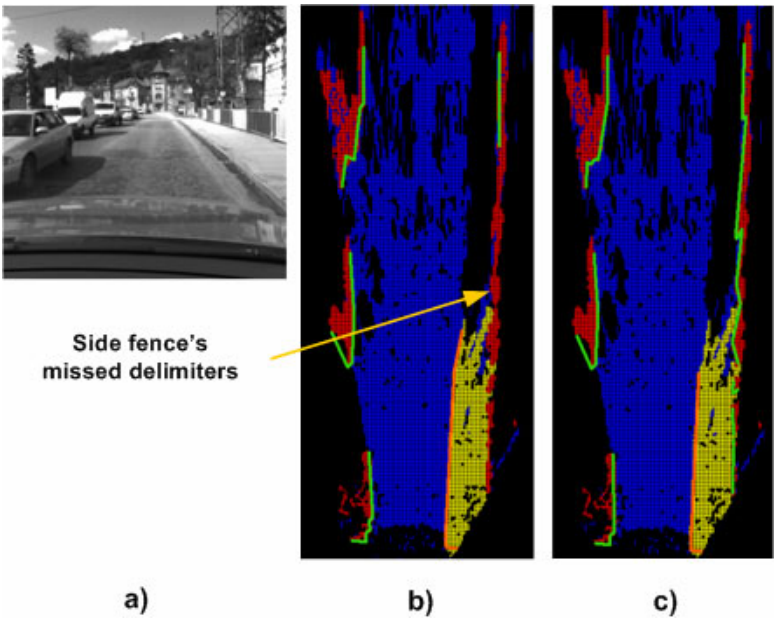


Fig. 9. Border scanning of a scene (a). The side fence's delimiters are missed in the case of the *Simple Border Scanning* (b) and have been detected in the case of the *Combined Border Scanning* algorithm (c).

In the Table 2 the results from the Variable Step Border Scanner and Fixed Step Border Scanner are computed for the same driving scene. It can be observed that the number of detected points is greater in the case of Variable Step Border Scanner algorithm, thereby 11466 points, which means 28 detected points per frame in comparison with 22 detected points per frame in the case of Fixed Step Border Scanner algorithm.

Table 2. Fixed Step Border Scanner vs. Variable Step Border Scanner

	Fixed Step Border Scanner	Variable Step Border Scanner
Number of Frames	406	406
Detected points	9058	11466
The radial step (radians)	0.01	variable
Points per Frames	22	28
Average processing time per frame	4 ms	5 ms

The average extraction time using the 3A Tracing algorithm is about 0.7ms per frame and depends on the angular resolution in the case of Border Scanner approach.

Fig. 10a shows how the radial step size variation affects the system response time using the Combined Border Scanning approach.

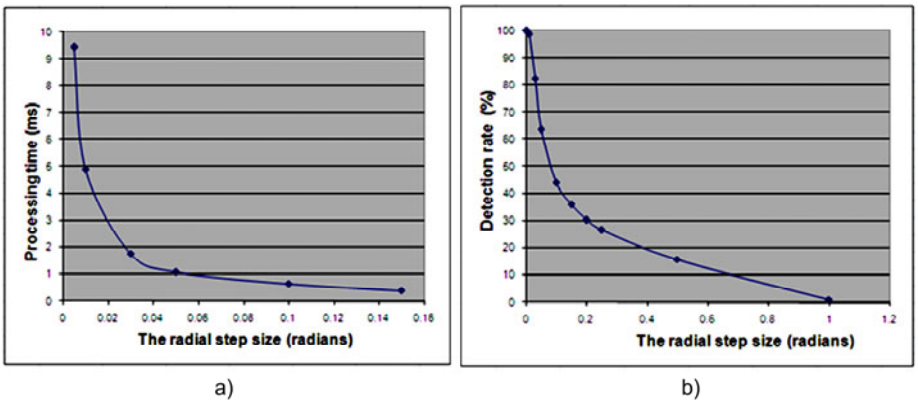


Fig. 10. a) The processing time vs. the radial step size. b) The detection rate vs. the radial step size

Fig. 10b is a diagram that shows the impact of radial step size on delimiters detection rate using the border scanner method. We can observe that, with a higher radial step size we obtain an increase in processing time while the detection rate decreases. The solution is a tradeoff between the system processing time and detection rate.

Fig. 11 presents results for various traffic scenes using the Combined Border Scanning method. For the border scanning algorithm with a radial step of 0.01 radians the average processing time is about 5ms and the delimiters detection rate is 98.66%.

Fig. 11. Object delimiters detection through the Combined Border Scanning algorithm for various traffic scenes. The delimiters are projected onto the Left Image and are represented as grids labeled as Traffic Isles (orange) or Objects (light green). The grid height is the same as the enveloped object by the current delimiter.

5 Conclusions

In this paper we present and evaluate several methods for real-time environment representation through the object delimiter extraction and characterization from dense stereovision images. The delimiters detection is based on processing the information provided by a 3D classified occupancy grid obtained from the raw dense stereo information. The result is a more compact 2.5D model for representing the environment, as a set of polylines. Each polyline element inherits the type (object, curb), position and height properties of the associated object from the occupancy grid.

We have developed an improved Contour Tracing method named 3A Tracing algorithm that eliminates the situation when two polygonal segments can intersect each other.

Another approach presented in this paper is the polyline extraction through the radial scanning of the occupancy grid. Although the tracing approach is more computationally-efficient, the results provided by the Border Scanner algorithm are more appropriate for detecting the real obstacle delimiters from the scene. The algorithm is able to extract only the visible area from the ego-vehicle since the occluded points do not offer relevant information. Using the Border Scanner algorithm, our system is fast and achieves a high rate of detection: 98.66%.

As future work we propose to extend our approaches by using temporal filtering of delimiter points in order to improve the polyline representation accuracy. The final goal of our research is to develop a system for complex environments that would achieve high performances with respect to accuracy, confidence and real-time capabilities.

References

1. Pijpers, M.: Sensors in adas. Universiteit Twente (2007)
2. Dellaert, F., Thorpe, C.: Robust Car Tracking using Kalman filtering and Bayesian templates. In: Proceedings of Conference on Intelligent Transportation Systems, vol. 3207, pp. 72–83 (1997)
3. Nedevschi, S., Danescu, R., Marita, T., Oniga, F., Pocol, C., Sobol, S., Tomiuc, C., Vancea, C., Meinecke, M.M., Graf, T., To, T.B., Obojski, M.A.: A sensor for urban driving assistance systems based on dense stereovision. In: Intelligent Vehicles, Istanbul, pp. 278–286 (2007)
4. Harati, A., Siegwart, R.: A new approach to segmentation of 2d range scans into linear regions. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA (2007)
5. Magin, G., Russ, A.: Supporting real-time update of an environment representation for autonomous mobile robots real-time systems. In: EuroMicro Workshop on Real-Time Systems, pp. 34–39 (1994)
6. Joshi, R.R.: Novel metrics for map-matching in in-vehicle navigation system. IEEE Intelligent Vehicle Symposium 1, 36–43 (2002)
7. Laviers, K.R., Peterson, G.L.: Cognitive robot mapping with polylines and an absolute space representation. In: IEEE International Conference on Robotics and Automation, pp. 3771–3776. Hilton New Orleans Riverside, New Orleans (2004)
8. Veeck, M., Burgard, W.: Learning polyline maps from range scan data acquired with mobile robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS (2004)
9. Kolski, S., Ferguson, D., Bellino, M., Siegwart, R.: Autonomous driving in structured and unstructured environments. In: IEEE Intelligent Vehicles Symposium (2006)
10. Madhavan, R.: Terrain aided localization of autonomous vehicles. In: Symposium on Automation and Robotics in Construction, Gaithersburg (2002)
11. Goncalves, A., Godinho, A., Sequeira, J.: Lowcost sensing for autonomous car driving in highways. In: ICINCO 2007 - 4th International Conference on Informatics in Control, Automation and Robotics, Angers, France (2007)
12. Prakash, S., Thomas, S.: Contour tracking with condensation/stochastic search. Dept. of CSE. IIT Kanpur (2007)
13. Gonzales, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Addison Wesley, Reading (2002)
14. Woodill, J.I., Gordon, G., Buck, R.: Tyzx deepsea high speed stereo vision system. In: IEEE Computer Society Workshop on Real Time 3-D Sensors and Their Use, Conference on Computer Vision and Pattern Recognition (2004)
15. Oniga, F., Nedevschi, S., Meinecke, M.M., To, T.B.: Road surface and obstacle detection based on elevation maps from dense stereo. In: IEEE Intelligent Transportation Systems Conference, Seattle, WA, pp. 859–865 (2007)
16. Marita, T., Oniga, F., Nedevschi, S., Graf, T., Schmidt, R.: Camera calibration method for far range stereovision sensors used in vehicles. In: IEEE Intelligent Vehicles Symposium (IV2006), Tokyo, Japan, pp. 356–363 (2006)

Vision-Based Automatic Approach and Landing of Fixed-Wing Aircraft Using a Dense Visual Tracking

Tiago F. Gonçalves[1,⋆], José R. Azinheira[1], and Patrick Rives[2]

[1] IDMEC, IST/TULisbon
Av. Rovisco Pais, N°1, 1049-001 Lisbon, Portugal
{t.goncalves,jraz}@dem.ist.utl.pt
[2] INRIA-Sophia Antipolis
2004 Route des Lucioles, BP93, 06902 Sophia-Antipolis, France
Patrick.Rives@inria.sophia.fr

Abstract. This paper presents a feasibility study of a vision-based autonomous approach and landing for an aircraft using a direct visual tracking method. Auto-landing systems based on the *Instrument Landing System* (ILS) have already proven their importance through decades but general aviation stills without cost-effective solutions for such conditions. However, vision-based systems have shown to have the adequate characteristics for the positioning relatively to the landing runway. In the present paper, rather than points, lines or other features susceptible of extraction and matching errors, dense information is tracked in the sequence of captured images using an *Efficient Second-Order Minimization* (ESM) method. An optimal control design is then proposed using the homography matrix as visual information in two distinct approaches: position-based visual servoing (PBVS) and image-based visual servoing (IBVS). To illustrate the proposed concepts, simulation results under realistic atmospheric disturbances are presented.

Keywords: Automatic approach and landing, Vision-based control, Linear optimal control, Dense visual tracking.

1 Introduction

Approach and landing are known to be the most demanding flight phases in fixed-wing flight operations. Due to the altitudes involved in flight and the consequent nonexisting depth perception, pilots must interpret position, attitude and distance to the runway using only two-dimensional cues like perspective, angular size and movement of the runway. At the same time, all six degrees of freedom of the aircraft must be controlled and coordinated in order to meet and track the correct glidepath till the touchdown.

In poor visibility conditions and degraded visual references, landing aids must be considered. The *Instrument Landing System* (ILS) is widely used in most of the international airports around the world allowing pilots to establish on the approach and follow the ILS, in autopilot or not, until the decision height is reached. At this point, the

⋆ This work is funded by the FP6 3rd Call European Commission Research Program under grant Project N°30839 - PEGASE.

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 269–282.
springerlink.com

pilot must have visual contact with the runway to continue the approach and proceed to the flare manoeuvre or, if it is not the case, to abort. This procedure has proven its reliability through decades but landing aids systems that require specific equipment are still not cost-effective for most of the general airports. However, in the last years, *Enhanced Visual Systems* (EVS) based on *Infrared* (IR) allowed to proceed to non-precision approaches and obstacle detection for all weather conditions. The vision-based control system proposed in the present paper intends then to take advantage of these emergent vision sensors in order to allow precision approaches and autonomous landing.

The intention of using vision systems for autonomous landings or simply estimate the aircraft position and attitude (pose) is not new. Flight tests of a vision-based autonomous landing relying on feature points on the runway were already referred by [1] whilst [2] present a feasibility study on pose determination for an aircraft night landing based on a model of the *Approach Lighting System* (ALS). Many others have followed in using vision-based control on fixed/rotary wings aircraft, and even airships, in several goals since autonomous aerial refueling ([3], [4]), stabilization with respect to a target ([5], [6]), linear structures following ([7], [8], [9]) and, obviously, automatic landing ([10], [11], [12], [13], [14]). In these problems, different types of visual features were considered including geometric model of the target, points, corners of the runway, binormalized Plucker coordinates, the three parallel lines of the runway (left, center and right sides) and the two parallel lines of the runway along with the horizon line and the vanishing point. Due to the standard geometry of the landing runway and the decoupling capabilities, the last two approaches have been preferred in problems of autonomous landing.

In contrast with feature extraction methods, direct or dense methods are known by their accuracy because all the information in the image is used without intermediate processes, reducing then the sources of errors. The usual disadvantage of such method is the computational consuming of the sum-of-squared-differences minimization due to the computation of the Hessian matrix. The *Efficient Second-order Minimization* (ESM) [15] method does not require the computation of the Hessian matrix maintaining however the high convergence rate characteristic of the second-order minimizations as the Newton method. Robust under arbitrary illumination changes [16] and with real-time capability, the ESM suits all the requirements to use images from the common CCD/CMOS to IR sensors.

In vision-based or visual servoing problems, a planar scene plays an important role since it simplifies the computation of the projective transformation between two images of the scene: the planar homography. The Euclidean homography, computed with the knowledge of the calibration matrix of the imagery sensor, is here considered as the visual signal to use into the control loop in two distinct schemes. The *position-based visual servoing* (PBVS) uses the recovered pose of the aircraft from the estimated projective transformation whilst the *image-based visual servoing* (IBVS) uses the visual signal directly into the control loop by means of the interaction matrix. The controller gains, from standard output error LQR design with a PI structure, are common for both schemes whose results will be then compared with a sensor-based simulation scheme where precise measures are considered.

The present paper is organized as follows: In the Section 2, some useful notations in computer vision are presented, using as example the rigid-body motion equation, along with an introduction of the considered frames and a description of the aircraft dynamics and the pinhole camera models. In the same section, the two-view geometry is introduced as the basis for the IBVS control law. The PBVS and IBVS control laws are then presented in the Section 3 as well as the optimal controller design. The results are shown in Section 4 while the final conclusions are presented in Section 5.

2 Theoretical Background

2.1 Notations

The rigid-body motion of frame b with respect to frame a can be expressed as

$$ {}^a\mathbf{X} = {}^a\mathbf{R}_b \, {}^b\mathbf{X} + {}^a\mathbf{t}_b \tag{1} $$

where ${}^a\mathbf{R}_b \in \mathbb{SO}(3)$ and ${}^a\mathbf{t}_b \in \mathbb{R}^3$ denote the relative rotation matrix and translation vector, respectively, and ${}^a\mathbf{X} \in \mathbb{R}^3$ the coordinates of a 3D point in frame a. Alternatively, in homogeneous coordinates, the same relation can by given by

$$ {}^a\mathcal{X} = {}^a\mathbf{T}_b \, {}^b\mathcal{X} = \begin{bmatrix} {}^a\mathbf{R}_b & {}^a\mathbf{t}_b \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^b\mathbf{X} \\ 1 \end{bmatrix} \tag{2} $$

where, ${}^a\mathbf{T}_b \in \mathbb{SE}(3)$, $\mathbf{0}$ denotes a matrix of zeros with the appropriate dimensions and ${}^a\mathcal{X} \in \mathbb{R}^4$ are the homogeneous coordinates of the point ${}^a\mathbf{X}$. In the same way, the Coriolis theorem applied to 3D points can also be expressed in homogenous coordinates, as a result of the derivative of the rigid-body motion relation in Eq. (2),

$$ {}^a\dot{\mathcal{X}} = {}^a\dot{\mathbf{T}}_b \, {}^b\mathcal{X} = {}^a\dot{\mathbf{T}}_b \, {}^a\mathbf{T}_b^{-1} \, {}^a\mathcal{X} = \begin{bmatrix} \widehat{\omega} & \mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix} {}^a\mathcal{X} = {}^a\widehat{\mathbf{V}}_{ab} \, {}^a\mathcal{X}, \quad {}^a\widehat{\mathbf{V}}_{ab} \in \mathbb{R}^{4\times4} $$

where, the angular velocity tensor $\widehat{\omega} \in \mathbb{R}^{3\times3}$ is the skew-symmetric matrix of the angular velocity vector ω such that $\omega \times \mathbf{X} = \widehat{\omega}\mathbf{X}$ and the vector ${}^a\mathbf{V}_{ab} = [\mathbf{v}, \omega]^\top \in \mathbb{R}^6$ denotes the velocity screw and indicates the velocity of the frame b moving relative to a and viewed from the frame a. Also important in the present paper is the definition of stacked matrix, denoted by the superscript " ^{s} ", where the matrix columns are rearranged into a single column vector.

2.2 Aircraft Dynamic Model

Let $\mathcal{F}_0$ be the earth frame, also called NED for North-East-Down, whose origin coincides with the desirable touchdown point in the runway. The latter, unless explicitly indicated and without loss of generality, will be considered aligned with North. The aircraft linear velocity $\mathbf{v} = [u, v, w]^\top \in \mathbb{R}^3$, as well as its angular velocity $\omega = [p, q, r]^\top \in \mathbb{R}^3$, is expressed in the aircraft body frame $\mathcal{F}_c$ whose origin is at the center of gravity where u is defined towards the aircraft nose, v towards the right

wing and w downwards. The attitude, or orientation, of the aircraft with respect to the earth frame $\mathcal{F}_0$ is stated in terms of Euler angles $\Phi = [\phi, \theta, \psi]^\top \subset \mathbb{R}^3$, the roll, pitch and yaw angles respectively.

The aircraft motion in atmospheric flight is usually deduced using Newton's second law and considering the motion of the aircraft in the earth frame $\mathcal{F}_0$, assumed as an inertial frame, under the influence of forces and torques due to gravity, propulsion and aerodynamics. As mentioned above, both linear and angular velocities of the aircraft are expressed in the body frame $\mathcal{F}_b$ as well as for the considered forces and moments. As a consequence, the Coriolis theorem must be invoked and the kinematic equations appear naturally relating the angular velocity rate ω with the time derivative of the Euler angles $\dot{\Phi} = \mathbf{R}^{-1}\omega$ and the instantaneous linear velocity $\mathbf{v}$ with the time derivative of the NED position $[\dot{N}, \dot{E}, \dot{D}]^\top = \mathbf{S}^\top \mathbf{v}$.

In order to simplify the controller design, it is common to linearize the non-linear model around a given equilibrium flight condition, usually a function of airspeed $\mathbf{V}_0$ and altitude h_0. This equilibrium or trim flight is frequently chosen to be a steady wings-level flight, without presence of wind disturbances, also justified here since non-straight landing approaches are not considered in the present paper. The resultant linear model is than function of the perturbation in the state vector $\mathbf{x}$ and in the input vector $\mathbf{u}$ as

$$\begin{bmatrix} \dot{\mathbf{x}}_v \\ \dot{\mathbf{x}}_h \end{bmatrix} = \begin{bmatrix} \mathbf{A}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_h \end{bmatrix} \begin{bmatrix} \mathbf{x}_v \\ \mathbf{x}_h \end{bmatrix} + \begin{bmatrix} \mathbf{B}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_h \end{bmatrix} \begin{bmatrix} \mathbf{u}_v \\ \mathbf{u}_h \end{bmatrix} \tag{3}$$

describing the dynamics of the two resultant decoupled, lateral and longitudinal, motions. The longitudinal, or vertical, state vector is $\mathbf{x}_v = [u, w, q, \theta]^\top \in \mathbb{R}^4$ and the respective input vector $\mathbf{u}_v = [\delta_E, \delta_T]^\top \in \mathbb{R}^2$ (elevator and throttle) while, in the lateral case, the state vector is $\mathbf{x}_h = [v, p, r, \phi]^\top \in \mathbb{R}^4$ and the respective input vector $\mathbf{u}_h = [\delta_A, \delta_R]^\top \in \mathbb{R}^2$ (ailerons and rudder). Because the equilibrium flight condition is slowly varying for manoeuvres as the landing phase, the linearized model in Eq. (3) can be considered constant along all the glidepath.

2.3 Two-Views Geometry

The onboard camera frame $\mathcal{F}_c$, rigidly attached to the aircraft, has its origin at the center of projection of the camera, also called pinhole. The corresponding z-axis, perpendicular to the image plane, lies on the optical axis while the x- and y- axis are defined towards right and down, respectively. Note that the camera frame $\mathcal{F}_c$ is not in agreement with the one usually defined in flight mechanics.

Let $\mathcal{P}$ be a 3D point whose coordinates in the current camera frame ${}^c\mathbf{X}$ could be related with those ${}^*\mathbf{X}$ in a second camera frame $\mathcal{F}_*$, denoted reference camera frame, by

$${}^c\mathbf{X} = {}^c\mathbf{R}_* {}^*\mathbf{X} + {}^c\mathbf{t}_* \tag{4}$$

Considering that $\mathcal{P}$ lies on a plane Π defined by the unit normal vector ${}^*\mathbf{n} \in \mathbb{R}^3$ and the distance d^* such as

$$\frac{1}{d^*} {}^*\mathbf{n}^\top {}^*\mathbf{X} = 1 \tag{5}$$

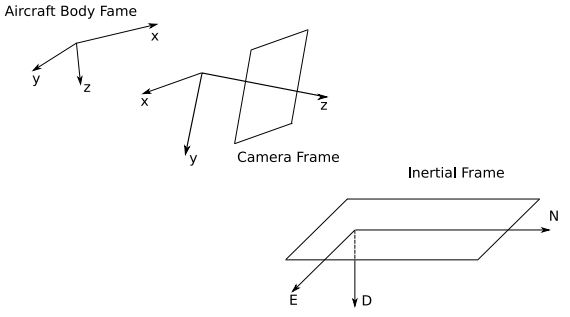


Fig. 1. Illustration of the considered frames and notations

there exists a linear transformation ${}^c\mathbf{H}_*$ between both coordinates given by

$$ {}^c\mathbf{H}_* = {}^c\mathbf{R}_* + \frac{1}{d^*} {}^c\mathbf{t}_* {}^c\mathbf{n}_*^\top \tag{6} $$

where ${}^c\mathbf{H}_* \in \mathbb{R}^{3\times3}$ is the so-called Euclidean homography matrix. Therefore, given a set of matched point, lines or even dense information from two images $\mathcal{I}$ and $\mathcal{I}^*$, it is possible to compute the projective homography matrix $\mathbf{G} \in \mathbb{R}^{3\times3}$ up to a scale factor. Knowing an estimative of the calibration matrix $\mathbf{K}$, the Euclidean homography matrix $\mathbf{H}$ can then be computed also up to a scale factor

$$ \mathbf{H} = \mathbf{K}^{-1}\mathbf{G}\mathbf{K} \tag{7} $$

The calibration matrix, or the camera intrinsical parameters, $\mathbf{K} \in \mathbb{R}^{3\times3}$ is defined as follows

$$ \mathbf{K} = \begin{bmatrix} f_x & fs & p_{x_0} \\ 0 & f_y & p_{y_0} \\ 0 & 0 & 1 \end{bmatrix} \tag{8} $$

where, the coordinates $\mathbf{p}_0 = [p_{x_0}, p_{y_0}, 1]^\top \in \mathbb{R}^3$ define the principal point, corresponding to the intersection between the image plane and the optical axis, s is the skew factor (zero for most of the cameras) and finally, f_x and f_y are the focal lengths in both directions such that when $f_x = f_y$ the camera sensor presents square pixels.

3 Vision-Based Automatic Approach and Landing

3.1 Visual Tracking

The visual tracking is achieved by directly estimating the projective transformation between the image taken from the airborne camera and a given reference image. The reference images are then the key to relate the motion of the aircraft $\mathcal{F}_b$, through its airborne camera $\mathcal{F}_c$, with respect to the earth frame $\mathcal{F}_0$. For the PBVS scheme, it is the known pose of the reference camera with respect to the earth frame that will allow us to reconstruct the aircraft position with respect to the same frame. In what concerns

the IBVS, where the aim is to reach a certain configuration expressed in terms of the considered feature, the path planning is then an implicit need of such scheme. For example, if lines are considered as features, the path planning is defined as a function of the parameters which define those lines. In the present case, the path planning shall be defined by images because it is the dense information that is used in order to estimate the projective homography ${}^{c}\mathbf{G}_{*}$.

3.2 Linear Controller

The standard LQR optimal control technique was chosen for the controller design, based on the linearized models of both longitudinal and lateral motions in Eq. (3). Since not all the states are expected to be driven to zero but to a given reference, the control law is more conveniently expressed as an optimal state error feedback. The objective of the following vision-based control approaches is then to express the respective control laws as a function of the visual information, which is directly or indirectly related with the pose of the aircraft. As a consequence, the pose state vector $\mathbf{P} = [n, e, d, \phi, \theta, \psi]^{\top} \in \mathbb{R}^6$, in agreement to the type of vision-based control approach, is given differently from the velocity screw $\mathbf{V} = [u, v, w, p, q, r]^{\top} \in \mathbf{R}^6$, which could be provided from an existent *Inertial Navigation System* (INS) or from some filtering method based on the estimated pose. Thus, the following vision-based control laws are more correctly expressed as

$$\mathbf{u} = -\mathbf{k}_{\mathbf{P}}(\mathbf{P} - \mathbf{P}^{*}) - \mathbf{k}_{\mathbf{V}}(\mathbf{V} - \mathbf{V}^{*}) \tag{9}$$

where, $\mathbf{k}_{\mathbf{P}}$ and $\mathbf{k}_{\mathbf{V}}$ are the controller gains relative to the pose and velocity states, respectively.

3.3 Position-Based Visual Servoing

In the position-based, or 3D, visual servoing (PBVS) the control law is expressed in the Cartesian space and, as a consequence, the visual information computed in the form of planar homography is used to reconstruct explicitly the pose (position and attitude). The airborne camera will be then considered as only another sensor that provides a measure of the aircraft pose.

In the same way that, knowing the relative pose between the two cameras, $\mathbf{R}$ and $\mathbf{t}$, and the planar scene parameters, $\mathbf{n}$ and d, it is possible to compute the planar homography matrix $\mathbf{H}$ it is also possible to recover the pose from the decomposition of the estimated projective homography $\mathbf{G}$, with the additional knowledge of the calibration matrix $\mathbf{K}$. The decomposition of $\mathbf{H}$ can be performed by singular value decomposition [17] or, more recently, by an analytical method [18]. These methods result into four different solutions but only two are physically admissible. The knowledge of the normal vector $\mathbf{n}$, which defines the planar scene Π, allows us then to choose the correct solution.

Therefore, from the decomposition of the estimated Euclidean homography

$${}^{c}\widetilde{\mathbf{H}}_{*} = \mathbf{K}^{-1}{}^{c}\widetilde{\mathbf{G}}_{*}\mathbf{K}, \tag{10}$$

both ${}^c\widetilde{\mathbf{R}}_*$ and ${}^c\widetilde{\mathbf{t}}_*/d^*$ are recovered, being respectively the rotation matrix and normalized translation vector. With the knowledge of the distance d^*, it is then possible to compute the estimated rigid-body relation of the aircraft frame $\mathcal{F}_b$ with respect to the inertial one $\mathcal{F}_0$ as

$$ {}^0\widetilde{\mathbf{T}}_b = {}^0\mathbf{T}_* \left({}^b\mathbf{T}_c {}^c\widetilde{\mathbf{T}}_* \right)^{-1} = \begin{bmatrix} {}^0\widetilde{\mathbf{R}}_b & {}^0\widetilde{\mathbf{t}}_b \\ \mathbf{0} & 1 \end{bmatrix} \tag{11} $$

where, ${}^b\mathbf{T}_c$ corresponds to the pose of the airborne camera frame $\mathcal{F}_c$ with respect to the aircraft body frame $\mathcal{F}_b$ and ${}^0\mathbf{T}_*$ to the pose of the reference camera frame $\mathcal{F}_*$ with respect to the earth frame $\mathcal{F}_0$. Finally, without further considerations, the estimated pose $\widetilde{\mathbf{P}}$ obtained from ${}^0\widetilde{\mathbf{R}}_b$ and ${}^0\widetilde{\mathbf{t}}_b$ could then be applied to the control law in Eq. (9) as

$$ \mathbf{u} = -\mathbf{k_P}(\widetilde{\mathbf{P}} - \mathbf{P}^*) - \mathbf{k_V}(\mathbf{V} - \mathbf{V}^*) \tag{12} $$

3.4 Image-Based Visual Servoing

In the image-based, or 2D, visual servoing (IBVS) the control law is expressed directly in the image space. Then, in contrast with the previous approach, the IBVS does not need the explicit aircraft pose relative to the earth frame. Instead, the estimated planar homography $\widetilde{\mathbf{H}}$ is used directly into the control law as some kind of pose information such that reaching a certain reference configuration $\mathbf{H}^*$ the aircraft presents the intended pose. This is the reason why an IBVS scheme needs implicitly for path planning expressed in terms of the considered features.

Interaction Matrix. In IBVS schemes, an important definition is that of the interaction matrix which is responsible to relate the time derivative of the visual signal vector $\mathbf{s} \in \mathbb{R}^k$ with the camera velocity screw ${}^c\mathbf{V}_{c*} \in \mathbb{R}^6$ as

$$ \dot{\mathbf{s}} = \mathbf{L_s} {}^c\mathbf{V}_{c*} \tag{13} $$

where, $\mathbf{L_s} \in \mathbb{R}^{k\times 6}$ is the interaction matrix, or the feature jacobian. Let us consider, for a moment, that the visual signal vector $\mathbf{s}$ is a function of the Euclidean homography matrix ${}^c\mathbf{H}_*$, the visual feature considered in the present paper. Thus, the time derivative of $\mathbf{s}$, admitting the vector ${}^*\mathbf{n}/d^*$ as slowly varying, is

$$ \dot{\mathbf{s}} = {}^c\dot{\mathbf{H}}_* = {}^c\dot{\mathbf{R}}_* + \frac{1}{d^*} {}^c\dot{\mathbf{t}}_* {}^*\mathbf{n}^\top \tag{14} $$

Now, it is known that both ${}^c\dot{\mathbf{R}}_*$ and ${}^c\dot{\mathbf{t}}_*$ are related with the velocity screw ${}^c\mathbf{V}_{c*}$, which could be determined using Eq. (3), as follows

$$ {}^c\widehat{\mathbf{V}}_{c*} = \begin{bmatrix} {}^c\dot{\mathbf{R}}_* {}^c\mathbf{R}_*^\top & {}^c\dot{\mathbf{t}}_* - {}^c\dot{\mathbf{R}}_* {}^c\mathbf{R}_*^\top {}^c\mathbf{t}_* \\ \mathbf{0} & 1 \end{bmatrix} $$

from where, ${}^c\dot{\mathbf{R}}_* = {}^c\widehat{\omega}\, {}^c\mathbf{R}_*$ and ${}^c\dot{\mathbf{t}}_* = {}^c\mathbf{v} + {}^c\widehat{\omega}\, {}^c\mathbf{t}_*$.

By using such results back in Eq. (14) results on

$$ \begin{aligned} {}^c\dot{\mathbf{H}}_* &= {}^c\widehat{\omega} \left({}^c\mathbf{R}_* + \frac{1}{d^*} {}^c\mathbf{t}_* {}^*\mathbf{n}^\top \right) + \frac{1}{d^*} {}^c\mathbf{v}^* \mathbf{n}^\top = \\ &= {}^c\widehat{\omega}\, {}^c\mathbf{H}_* + \frac{1}{d^*} {}^c\mathbf{v}^* \mathbf{n}^\top \end{aligned} \tag{15} $$

Hereafter, in order to obtain the visual signal vector, the stacked version of the homography matrix ${}^{c}\dot{\mathbf{H}}^{s}_{*}$ must be considered and, as a result, the interaction matrix is given by

$$\dot{\mathbf{s}} = {}^{c}\dot{\mathbf{H}}^{s}_{*} = \mathbf{L}_{\mathbf{H}}{}^{c}\mathbf{V}_{c*} = \begin{bmatrix} \mathbf{I}(3)^{*}\mathbf{n}_1/d^{*} & -{}^{c}\widehat{\mathbf{H}}_{*1} \\ \mathbf{I}(3)^{*}\mathbf{n}_2/d^{*} & -{}^{c}\widehat{\mathbf{H}}_{*2} \\ \mathbf{I}(3)^{*}\mathbf{n}_3/d^{*} & -{}^{c}\widehat{\mathbf{H}}_{*3} \end{bmatrix} \begin{bmatrix} {}^{c}\mathbf{v} \\ \\ {}^{c}\omega \end{bmatrix} \tag{16}$$

where, $\mathbf{I}(3)$ is the 3×3 identity matrix and $\mathbf{H}_i$ is the ith column of the matrix as well as $\mathbf{n}_i$ is the ith element of the vector. Note that, $\widehat{\omega}\mathbf{H}$ is the external product of ω with all the columns of $\mathbf{H}$ and $\omega \times \mathbf{H}_1 = -\mathbf{H}_1 \times \omega = -\widehat{\mathbf{H}}_1\omega$. However, the velocity screw considered in Eq. (13) denotes the velocity of the reference frame $\mathcal{F}_*$ with respect to the airborne camera frame $\mathcal{F}_c$ and viewed from $\mathcal{F}_c$, *i.e.* ${}^{c}\widehat{\mathbf{V}}_{c*}$, which is not in agreement with the aircraft velocity screw that must be applied into the control law in Eq. (9). Instead, the velocity screw shall be expressed with respect to the reference camera frame $\mathcal{F}_*$ and viewed from aircraft body frame $\mathcal{F}_b$, *i.e.* ${}^{b}\widehat{\mathbf{V}}_{*c}$, where the control law is effectively applied. In this manner, knowing the following relation from the adjoint map

$${}^{c}\widehat{\mathbf{V}}_{c*} = {}^{c}\mathbf{T}_{b}\,{}^{b}\widehat{\mathbf{V}}_{c*}\,{}^{c}\mathbf{T}_{b}^{-1} = {}^{b}\mathbf{T}_{c}^{-1}(-{}^{b}\widehat{\mathbf{V}}_{*c})^{b}\mathbf{T}_{c} \tag{17}$$

it is possible to find the velocity screw transformation ${}^{c}\mathbf{W}_{b} \in \mathbb{R}^{6\times 6}$ given by

$${}^{c}\mathbf{W}_{b} = \begin{bmatrix} {}^{b}\mathbf{R}_{c}^{\top} & -{}^{b}\mathbf{R}_{c}^{\top}{}^{b}\widehat{t}_{c} \\ \mathbf{0} & {}^{b}\mathbf{R}_{c}^{\top} \end{bmatrix} \tag{18}$$

such that

$${}^{c}\mathbf{V}_{c*} = -{}^{c}\mathbf{W}_{b}\,{}^{b}\mathbf{V}_{*c} \tag{19}$$

where ${}^{b}\mathbf{R}_{c}$ and ${}^{b}\mathbf{t}_{c}$ define the pose of the onboard camera frame $\mathcal{F}_c$ with respect to the aircraft body frame $\mathcal{F}_b$. Using Eq. (16) along with the result from Eq. (18) results as follows

$$\dot{\mathbf{s}} = {}^{c}\dot{\mathbf{H}}^{s}_{*} = \mathbf{L}_{\mathbf{H}}{}^{c}\mathbf{W}_{b}\,{}^{b}\mathbf{V}_{*c} \tag{20}$$

Finally, let us consider the linearized version of the previous result as

$$\mathbf{s} - \mathbf{s}^{*} = {}^{c}\mathbf{H}^{s}_{*} - \mathbf{H}^{*s} = \mathbf{L}_{\mathbf{H}}{}^{c}\mathbf{W}_{b}\,{}^{b}\mathbf{W}_{0}\,(\mathbf{P} - \mathbf{P}^{*}) \tag{21}$$

where,

$${}^{b}\mathbf{W}_{0} = \begin{bmatrix} \mathbf{S}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_0 \end{bmatrix} \tag{22}$$

is composed by the kinematic and navigation equations, respectively, linearized for the same trim point as for the aircraft linear model $[\phi, \theta, \psi]_0^{\top} = [0, \theta_0, 0]^{\top}$. It is then possible to relate the pose error $\mathbf{P} - \mathbf{P}^{*}$ of the aircraft with the Euclidean homography error ${}^{c}\mathbf{H}^{s}_{*} - \mathbf{H}^{*s}$.

Since the approach glidepath is a well-defined trajectory the existence of a database composed by images $\{\mathcal{I}^{*}_{k}\}_{k=0}^{N}$, taken all along the reference path for equidistant longitudinal distances $d_{\mathcal{I}}$, can be considered, as illustrated in Fig 2. The easiest way to defined the error function $\mathbf{s} - \mathbf{s}^{*}$ to be minimized by the control law is then to establish

$$\mathbf{s}^{*} = \mathbf{H}^{*s} = \mathbf{I}^{s} \tag{23}$$

such that the current and reference images match for $\mathbf{s} - \mathbf{s}^{*} = \mathbf{0}$.

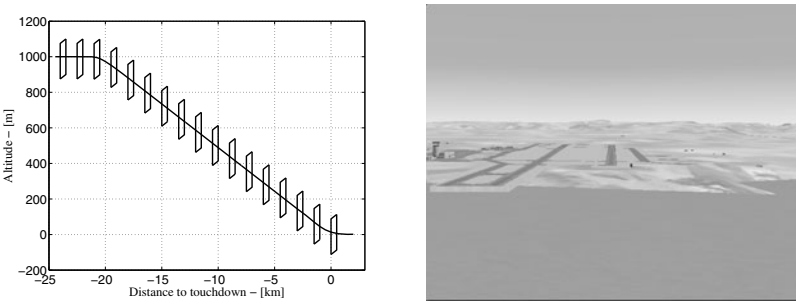


Fig. 2. Illustration of the sequence of reference images along the path to follow along with one of the images from the database

Due to the slow longitudinal dynamics of the aircraft, an acceptable result can be achieved using a sufficiently small distance $d_{\mathcal{I}}$ between consecutive reference images but the size of the required database will be significant. For larger $d_{\mathcal{I}}$ and consequent altitude errors, important oscillations on the longitudinal motion of the aircraft are verified. An interpolation of the visual reference $\mathbf{s}^*$ is then desirable.

Path Interpolation. Let us consider $\left\{{}^{k-1}\mathbf{H}_k\right\}_{k=1}^{N}$, the set of Euclidean homographies, computed off-line, between two consecutive reference images $\mathcal{I}_k^*$. Since the glidepath is essentially a straight line, it is possible to consider an interpolation in the homography space as

$$\mathbf{H}_k^*(\mu) = \mathbf{I} + \mu\left({}^{k-1}\mathbf{H}_k - \mathbf{I}\right) \tag{24}$$

such that it corresponds to an interpolation in the Euclidean space since ${}^{k-1}\mathbf{R}_k = \mathbf{I}$ and, as a consequence,

$$\mathbf{H}_k^*(\mu) = \mathbf{I} + \mu\left(\mathbf{I} + \frac{1}{d_k^*}{}^{k-1}\mathbf{t}_k{}^*\mathbf{n}_k^\top - \mathbf{I}\right) = \mathbf{I} + \frac{\mu}{d_k^*}{}^{k-1}\mathbf{t}_k{}^*\mathbf{n}_k^\top \tag{25}$$

where $\mu \in [0, 1]$ is the interpolator parameter.

The interpolator parameter μ, associated to the longitudinal distance between the current camera frame $\mathcal{F}_c$ and the reference camera frame $\mathcal{F}_*$, can be computed directly from the estimated Euclidean homography ${}^c\mathbf{H}_*$ as follows

$$\mu = \frac{d_k^*}{d_{\mathcal{I}}}\left[1, 0, 0\right]\text{vec}\left({}^c\mathbf{H}_* - {}^c\mathbf{H}_*^\top\right) \tag{26}$$

which, for small attitude errors and knowing that ${}^*\mathbf{n} \approx [0, 1, 0]^\top$ then

$$\mu \approx \frac{{}^c t_{*_z}}{d_{\mathcal{I}}} \tag{27}$$

Control Law. The proposed homography-based IBVS control law is then finally given by

$$\mathbf{u} = -\mathbf{k_P} \left(\mathbf{L_H}^c\mathbf{W}_0 \right)^\dagger \left({}^c\widetilde{\mathbf{H}}^s_* - \mathbf{H}^{*s}_k(\mu) \right) - \mathbf{k_V}(\mathbf{V} - \mathbf{V}^*) \tag{28}$$

where, $\mathbf{A}^\dagger = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top$ is the Moore-Penrose pseudo-inverse matrix.

4 Results

The vision-based control schemes proposed above have been developed and tested in an simulation framework where the non-linear aircraft model is implemented in Matlab/Simulink along with the control aspects, the image processing algorithms in C/C++ and the simulated image is generated by the FlightGear flight simulator. The aircraft model considered corresponds to a generic category B business jet aircraft with $50m/s$ stall speed, $265m/s$ maximum speed and $20m$ wing span. This simulation framework has also the capability to generate atmospheric condition effects like fog and rain as well as vision sensors effects like pixels spread function, noise, colorimetry, distortion and vibration of different types and levels.

The chosen airport scenario was Marseille-Marignane Airport with a nominal initial position defined by an altitude of $450m$ and a longitudinal distance to the runway of $9500m$, tracking a 3° descent for an airspeed of $60m/s$. In order to present an illustrative set of results and to verify the robustness of the proposed control schemes, we imposed an initial lateral error of $50m$, an altitude error of $30m$ and a steady wind composed by $10m/s$ headwind and $1m/s$ of crosswind. In what concerns the visual tracking aspects, a database of $500m$ equidistant images along the runway axis till the $100m$ height, and $50m$ after that, was considered and the following atmospheric conditions imposed: fog and rain densities of 40% and 80%. The airborne camera is considered rigidly attached to the aircraft and presents the following pose ${}^b\mathbf{P}_c = [4m, 0, 0.1m, 0, -8^\circ, 0]^\top$. The simulation framework operates with a $50ms$, or 20Hz, sampling rate.

For all the following figures, the results of the three considered cases are presented simultaneously and identified in agreement with the legend Figure 4(a). When available, the corresponding references are presented in dashed lines.

Let us start with the longitudinal trajectory illustrated in Figure 4(a) where it is possible to verify immediately that the PBVS result is almost coincident with the one where the sensor measurements were considered ideal (Sensors). Indeed, because the same control law is used for these two approaches, the results differ only due to the pose estimation errors from the visual tracking software. For the IBVS approach, the first observation goes to the convergence of the aircraft trajectory with respect to the reference descent that occurs later than for the other approaches but with a similar behavior with the other approaches thanks to the path interpolation. This fact is a consequence not only of the limited validity of the interaction matrix in Eq (21), computed for a stabilized descent flight, but also of the importance of the camera orientation over the position, for high altitudes, when the objective is to match two images. In more detail, the altitude error correction in Figure 4(b) shows then the IBVS with a slower response and an error after stabilization not greater than $3m$ as a consequence not only of the

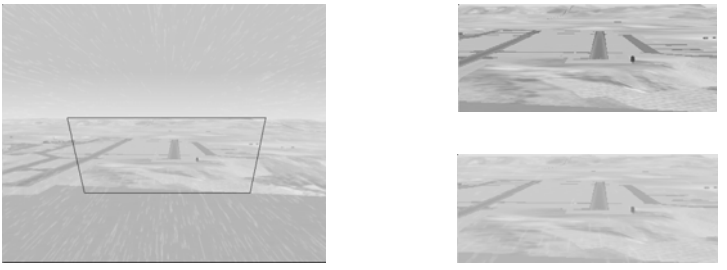
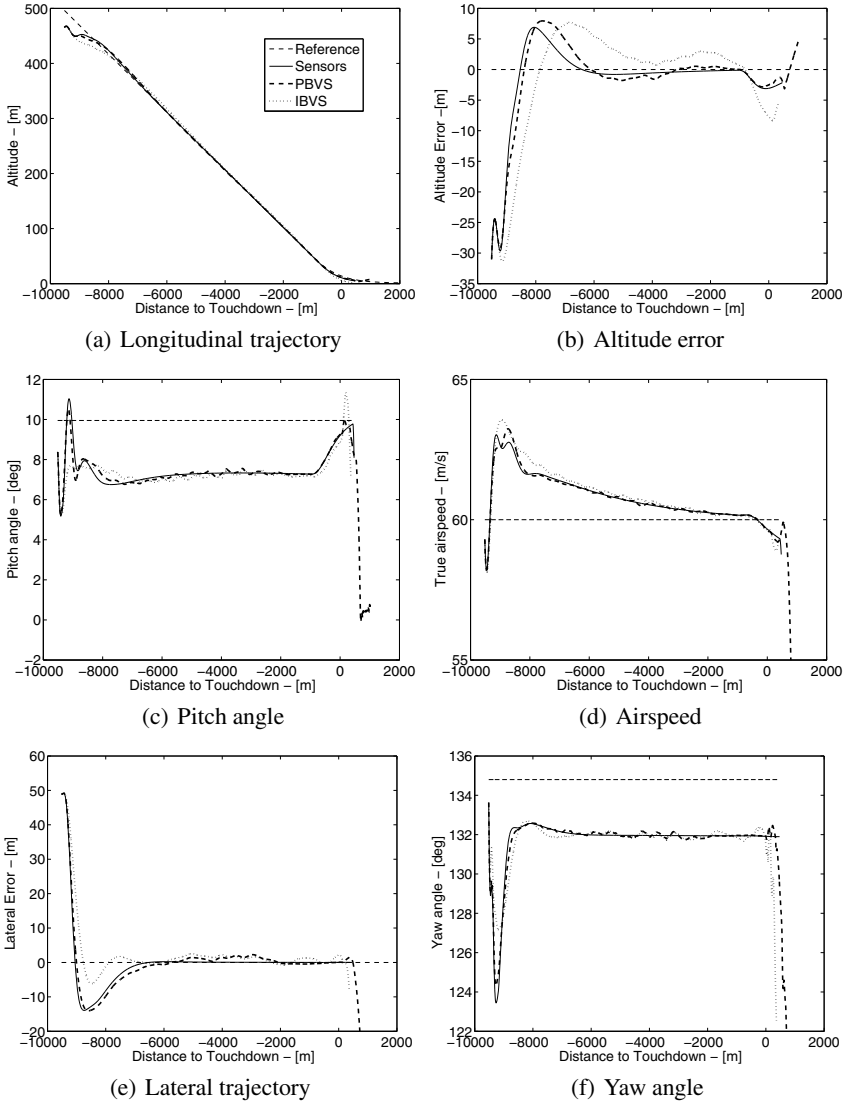


Fig. 3. Screenshot of the dense visual tracking software. The delimited zone (left) corresponds to the bottom-right image warped to match with the top-right image. The warp transformation is the estimated homography matrix.

respective estimation error but also due to the influence of the wind disturbance. Indeed, the path planning does not contemplate the presence of the wind, from which the aircraft attitude is dependent, leading to the presence of static errors. The increasing altitude error at the distance of $650m$ before the touchdown corresponds to the natural loss of altitude when proceeding to the pitch-up, or flare, manoeuvre (see Figure 4(c)) in order to reduce the vertical speed and correctly land the aircraft. In what concerns the touchdown distances, both Sensors and PBVS results are again very close and at a distance around $500m$ after the expected while, for the IBVS, this distance is of approximately $200m$. It should be referred that, only a distance between reference images of $200m$ provided acceptable results for the longitudinal motion of the aircraft when not considering path interpolation of the IBVS scheme. The path interpolation strategy allowed then to reduce the size of the required data and to improve the general behavior of the aircraft due to the absence of discontinuities on the control actions.

The lateral trajectory illustrated in Figure 4(e) shows a smooth lateral error correction for all the three control schemes where both visual control laws maintain an error below the $2.5m$, after convergence. Once more, the oscillations around the reference are a consequence of pose estimation errors from visual tracking software, which become more important near the Earth surface due to the high displacement of the pixels in the image and the violation of the planar assumption of the region around the runway. The consequence of these effects are perceptible in the final part not only in the lateral error correction but also in the yaw angle of the aircraft in Figure 4(f). For the latter, the static error is also an influence of the wind disturbance which imposes an error of $2.9°$ with respect to the runway orientation of exactly $134.8°$ North. As for the longitudinal motion, the lateral error correction for the IBVS is also here affected by the a more restricted regulation of the attitude angles leading to a slower and more damped response.

We should note here the precision of the dense visual tracking software. Indeed, the attitude estimation errors are often below $1°$ for transient responses and below $0.1°$ in steady state. Depending on the quantity of information available in the near field of the camera, the translation estimation error could vary between the $2m$ and $5m$ for the lateral error, $3m$ and 8 for altitude and between $50m$ and $200m$ for the longitudinal distance. The latter is naturally less precise due to its alignment with the optical axis of the camera.



(a) Longitudinal trajectory (b) Altitude error

(c) Pitch angle (d) Airspeed

(e) Lateral trajectory (f) Yaw angle

Fig. 4. Results from the vision-based control schemes (PBVS and IBVS) in comparison with the ideal situation of precise measurements (Sensors)

5 Conclusions

In the present paper, two vision-based control schemes for an automatic approach and landing of an aircraft using a direct visual tracking method are proposed. For the PBVS solution, where the vision system is nothing more than a sensor providing position and attitude measures, the results are naturally very similar with the ideal case. The IBVS

approach based on a path planning defined by a sequence of images show clearly to be able to correct an initial pose error and land the aircraft under windy conditions. Despite the inherent sensitivity of the vision tracking algorithm to the non-planarity of the scene and the high pixels displacement in the image for low altitudes, a shorter distance between the images of reference was enough to deal with potential problems. The absence of a filtering method, as the Kalman filter by instance, prove the robustness of the proposed control schemes and the reliability of the dense visual tracking. This clearly justifies the need of further studies to complete the validation and the eventual implementation of this system on a real aircraft.

References

1. Dickmanns, E.D., Schell, F.R.: Autonomous landing of airplanes by dynamic machine vision. In: IEEE Workshop on Application on Computer Vision, pp. 172–179 (1992)
2. Chatterji, G.B., Menon, P.K., Sridhar, B.: Vision-based Position and Attitude Determination for Aircraft Night Landing. AIAA Journal of Guidance, Control and Dynamics 21 (1998)
3. Kimmett, J., Valasek, J., Junkins, J.L.: Vision based controller for autonomous aerial refueling. In: Conference on Control Applications, pp. 1138–1143 (2002)
4. Mati, R., Pollini, L., Lunghi, A., Innocenti, M.: Vision-Based Autonomous Probe and Drogue Aerial Refueling. In: International Conference on Control and Automation, pp. 1–6 (2006)
5. Hamel, T., Mahony, R.: Visual servoing of an under-actuated dynamics rigid-body system: an image-based approach. IEEE Transactions on Robotics and Automation 18, 187–198 (2002)
6. Azinheira, J.R., Rives, P., Carvalho, J.R.H., Silveira, G.F., de Paiva, E.C., Bueno, S.S.: Visual Servo Control for the Hovering of an Outdoor Robotic Airship. In: IEEE International Conference on Robotics and Automation, vol. 3, pp. 2787–2792 (2002)
7. Silveira, G.F., Azinheira, J.R., Rives, P., Bueno, S.S.: Line Following Visual Servoing for Aerial Robots Combined with Complementary Sensors. In: IEEE International Conference on Robotics and Automation, pp. 1160–1165 (2003)
8. Rives, P., Azinheira, J.R.: Linear Structure Following by an Airship using Vanishing Point and Horizon Line in Visual Servoing Schemes. In: IEEE International Conference on Robotics and Automation, vol. 1, pp. 255–260 (2004)
9. Mahony, R., Hamel, T.: Image-based visual servo control of aerial robotic systems using linear images features. IEEE Transaction on Robotics 21, 227–239 (2005)
10. Sharp, C.S., Shakernia, O., Sastry, S.S.: A Vision System for Landing an Unmanned Aerial Vehicle. In: IEEE International Conference on Robotics and Automation, vol. 2, pp. 1720–1727 (2002)
11. Rives, P., Azinheira, J.R.: Visual Auto-Landing of an Autonomous Aircraft. In: Reearch Report N°4606, INRIA Sophia-Antilopis (2002)
12. Proctor, A.A., Johnson, E.N.: Vision-Only Aircraft Flight Control Methods and Test Results. In: AIAA Guidance, Navigation, and Control Conference and Exhibit (2002)
13. Bourquardez, O., Chaumette, F.: Visual servoing of an airplane for alignment with respect to a runway. In: IEEE International Conference on Robotics and Automation, pp. 1330–1355 (2007)
14. Bourquardez, O., Chaumette, F.: Visual Servoing of an Airplane for Auto-Landing. In: IEEE International Conference on Intelligent Robots and Systems, pp. 1314–1319 (2007)

15. Behimane, S., Malis, E.: Real-time image-based tracking of planes using Efficient Second-order Minimization. In: IEEE International Conference on Intelligent Robot and Systems, vol. 1, pp. 943–948 (2004)
16. Silveira, G., Malis, E.: Real-time Visual Tracking under Arbitrary Illumination Changes, pp. 1–6 (2007)
17. Faugeras, O.: Three-dimensional computer vision: a geometric view point. MIT Press, Cambridge (1993)
18. Vargas, M., Malis, E.: Deeper understanding of the homography decomposition for vision-base control. Research Report N°6303, INRIA Sophia-Antipolis (2007)

Part III

Signal Processing, Systems Modeling and Control

Studies on the Convergence of Information Theory and Control Theory

Prateep Roy *

University of Mumbai, India

Abstract. In this chapter being presented the information-theoretic explanation of Bodé Sensitivity Integral – a fundamental limitation of control theory, controllability grammian, Fisher Information and the issues of control under communication constraints. As resource-economic use of information is of prime concern in communication-constrained control problems, we need to emphasize more on informational aspect which has got direct relation with uncertainties in terms of Shannon Entropy and Mutual Information. Bodé Integral which is directly related to the disturbances can be correlated with the difference of entropies between the disturbance-input and measurable output of the system. These disturbances due to communication channel-induced noises and limited bandwidth are causing the information packet-loss and delays resulting in degradation of control performances.

Controllability Gramian is a measure of control performance and Fisher Information is a measure of accuracy of estimation (which needs to be done in uncertainties caused by packet-loss and delays of the network). Hence, in order to have better controllability, we need to explore the relation between the estimation theoretic (because information losses and delays force us to resort to estimation) parameter like Fisher Information Matrix (FIM) / information theoretic parameters and the control theoretic parameters of practical importance.

Keywords: Shannon Information, Controllability Gramian, Bode Sensitivity, Fisher Information.

1 Introduction

In recent years, there has been an increased interest for the fundamental limitations in feedback control. Bode's sensitivity integral (Bode Integral, in short) is a well-known formula that quantifies some of the limitations in feedback control for linear time-invariant systems. In [1], it is shown that there is a similar formula for linear time-periodic systems.

In this chapter, we are presenting the information-theoretic explanation of Bode Sensitivity Integral, a fundamental limitation of control theory, controllability gramian and the issues of control under communication constraints based on our paper [2]. As resource-economic use of information is of prime concern in communication-constrained control

* The author would like to thank ESIEE, Paris and Université Paris-Est for supporting this work as a part of his Doctoral Research.

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 285–299.
springerlink.com

problems, we need to emphasize more on informational aspect which has got direct relation with uncertainties in terms of Shannon Entropy and Mutual Information. Bode Integral which is directly related to the disturbances can be correlated with the difference of entropies between the disturbance-input and measurable output of the system (refer Eq. (1)). Entropy of the signals in the feedback loop helps to provide another interpretation of the Bode integral formula [3] [4] as follows :
For an LTI discrete-time system —-

$$H_c(x) - H_c(d) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \ln\left|S\left(e^{j\omega}\right)\right| d\omega = \sum_k \log(p_k) \tag{1}$$

Where $S\left(e^{j\omega}\right)$ is the transfer function of the feedback loop from the disturbance d to output x and p_{ui} 's are unstable poles ($|p_k| > 1$). of the open-loop plant; S is referred to as the sensitivity function for an open-loop plant gain P and a stabilizing feedback controller gain C, S is given by $S = \frac{1}{1+PC}$. Sensitivity shows how much sensitive is the observable output state to input Gaussian disturbance. Here, $H_c(x)$ and $H_c(d)$ denote the conditional entropy of the random processes associated with the output x and disturbance d respectively as per Fig.1. These disturbances due to communication channel-induced noises and limited bandwidth are causing the information packet-loss and delays resulting in degradation of control performances. Shannon Entropy of information theory is a stronger metric for uncertainty which hinders control of a system.

It is known that control theory and information theory share a common background as both theories study signals and dynamical systems in general. In distributed embedded systems (*DES*) or networked control systems (*NCS*), there are issues related to both control and communication since communication channels with data losses, time delays and quantization errors are employed between the plants and controllers [5]. To guarantee the overall control performance in such systems, it is important to evaluate the quantity of information that the channels can transfer. Thus, for the analysis of DES or NCS, information theoretic approaches are especially useful, and notions and results from this theory can be applied. The results in [6] and [7] show the limitation in the communication rate for the existence of controllers, encoders, and decoders to stabilize discrete-time linear feedback systems.

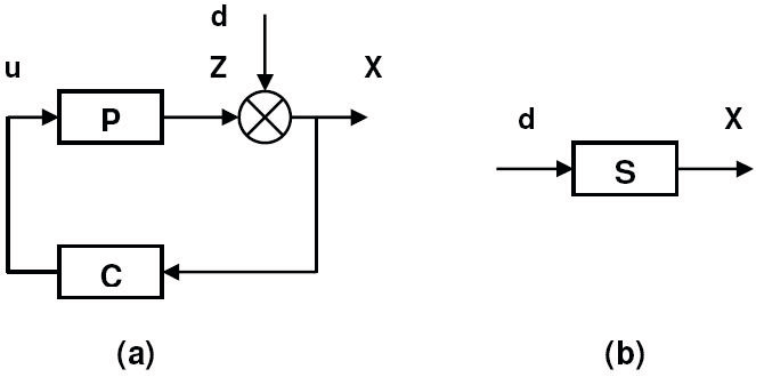


Fig. 1. (a)Feedback loop and (b)Sensitivity function

The focus of information theory is more on the signals and not on their input-output relation. Thus, based on information theoretic approaches, we may expect to extend prior results in control theory. One such result can be found in [8], where a sensitivity property is analyzed and Bode's integral formula [9] is extended to a more general class of systems. A fundamental limitation of sensitivity functions is presented in relation to the unstable poles of the plants.

2 Problem Formulation

Networked control systems suffer from the drawbacks of packet losses, delays and quantization in particular. These cause degradation of control performances and under some conditions instability. Uncertainties due to packet losses, delays, quantization, communication channel induced noises etc. have a great influence on the system's performance. If we consider only the uncertainties induced by channel noise and quantization we may write:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + w(t); \\ y(t) &= Cx(t) + Du(t) + v(t); \end{aligned} \tag{2}$$

where $A \in R^{n\times n}$ is the system or plant matrix and $B \in R^{n\times q}$ is the control or input matrix. Also, $x(t)$ is the state, $u(t)$ is the control input, $y(t)$ is the output, C is the output or measurement matrix, D is the Direct Feed matrix, $w(t)$ and $v(t)$ are the external disturbances and noises of Gaussian nature respectively.

Our aim is to achieve better control performance of system by tackling these uncertainties using Shannon's Mutual-Information, Information-Theoretic Entropy, Fisher Information and Bode Sensitivity. We present the information-theoretic model of such uncertainties and their possible reduction using information measures.

3 Preliminaries

By means of the connection between Bode integral and the entropy cost function, paper [10] provided a time-domain characterization of Bode integral. The traditional frequency domain interpretation is that, if the sensitivity of a closed-loop system is decreased over a particular frequency range – typically the low frequencies the designer "pays" for this in increased sensitivity outside this frequency range. This interpretation is also valid for the time-domain characterization presented in [10] provided one deals with time horizons rather than frequency ranges. Time-domain characterization of Bode's integral shows how the frequency domain trade-offs translate into the time-domain. Under the usual connection between the time and frequency domains: low (high) frequency signals are associated with long (short) time horizons. In Bode's result, it is important to differentiate between the stable poles, which do not contribute to the Bode sensitivity integral and the unstable poles, which do. Time-varying systems which can be decomposed into stable and unstable components are said to possess an exponential dichotomy. What the exponential dichotomy says is that the norm of the projection onto the stable subspace of any orbit in the system decays exponentially as

$t \to \infty$ and the norm of the projection onto the unstable subspace of any orbit decays exponentially as $t \to -\infty$, and furthermore that the stable and unstable subspaces are conjugate. The existence of an exponential dichotomy allows us to define a stability preserving state space transformation (a Lyapunov transformation) that separates the stable and unstable parts of the system.

3.1 Mutual Information

Mutual Information $I(X;Y)$, between X as the input variable and Y as the output variable, has the lower and upper bounds given by the following:

$R(D) =$ Rate Distortion $= MinI(X;Y)$
$C =$ Communication Channel Capacity $= MaxI(X;Y)$

where D is the distortion which happens when information is compressed (i.e. fewer bits are used to represent or code more frequent or redundant informations) and entropy is the limit to this compression i.e. if one compresses the information beyond the entropy limit there is a high probability that the information will be distorted or erroneous. This is as per Shannon's Source Coding Theorem. We code more frequently used symbols with fewer number of bits and vice-versa.

3.2 Shannon Entropy

Recalling the definition of Shannon Entropy [11] :

$$H = -\sum_i p_i \log_2 (p_i) \tag{3}$$

where p_i is the probability of the alternative i. The above quantity is known as the binary entropy in *bits* as we use logarithmic base of 2 (with logarithmic base e the entropy is in *nats*), and was shown by Shannon to correspond to the minimum average number of bits needed to encode a probabilistic source of N states distributed with probability p_i. The term entropy is associated with the uncertainty or randomness whereas information is used to reduce this uncertainty.

Uncertainty is the main hindrance to control and if we can reduce the uncertainty by getting the relevant information and utilizing the information properly so as to achieve the desired control performance of the system. Many researchers have posed the same question: "*How much information is required for controlling the system based on observed informations in the case where these informations are passed through communication channels in a NCS?*"

Mutual Information $I(X;Y)$ and Entropies $H(X)$, $H(Y)$ and joint entropy $H(X,Y)$ are related as :

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

where $H(X)$ is the uncertainty that X has about Y, $H(Y)$ is the uncertainty that Y has about X, and $H(X,Y)$ is the uncertainty that X and Y hold in common. Information value degrades over time and entropy value increases over time in general. The conditional version of the chain rule [12] :

$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$; valid for any random variables X and Y.

Mutual information $I(X;Y)$ is the amount of uncertainty in X, minus "the amount of uncertainty in X which remains after Y is known", which is equivalent to "the amount of uncertainty in X which is removed by knowing Y". This corroborates the intuitive meaning of mutual information as the amount of information (that is, reduction in uncertainty) that each variable is having about the other.

Just as entropy [13] in physical systems tends to increase in the course of time, the reverse is true for information about an information source : as information about the source is processed, it tends to decrease with time, becoming more corrupt or noisy until it is evidently destroyed unless additional information is made available. Here, information refers to the case of desired messages. Information value degrades over time and entropy value increases over time in general. We can use the timely information (information as and when it is generated) through feedback-loop to reduce the uncertainty and thus achieving better controllability because uncertainty in terms of information-theoretic entropy is the obstruction in the path of controllability. Lack of information (of appropriate quality and time-value) is the degradation in control performance in terms of controllability, stability etc.

3.3 Bode Integral

As pointed out and proved by researchers in [7], [10], [14], [15] that a stable system needs no information on its internal state or the environment to assure its stability. For unstable systems the mutual information between the initial state and the output of the system is related to its unstable poles. Stable poles do not contribute.

The simplest (and perhaps the best known) result is that, for an open loop stable plant, the integral of the logarithm of the sensitivity is zero; i.e.

$$\int_0^\infty \ln |S_0(j\omega)| \, d\omega = 0$$

Where, S_0 and ω being the sensitivity function and frequency respectively.

For linear systems Bode Integral is the difference in the entropy rates between the input and output of the systems which is an information-theoretic interpretation. For nonlinear system (if the open loop system is globally exponentially stable and has fading memory) this difference is zero. Fading Memory Requirement is used to limit the contributions of the past values of the input on the output. Fading is the commonly encountered phenomenon in wireless networks wherein the signals fade away (attenuates and varies with time and space).

Entropy of the signals in the feedback loop help provide another interpretation of the Bode integral formula [3], [4] as follows – Shannon Entropy - Bode Integral Relation can be rewritten as :

$$H_c(x) - H_c(d) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln \left| S\left(e^{j\omega}\right) \right| d\omega = \sum_k \log(p_k) \tag{4}$$

Consider a random variable $x \in \Re^m$ of continuous type with entropy associated with this is given by

$H(x) := -\int_{\Re^m} p(x) \ln p(x) dx;$

where $p(x)$ being the probability density function of x and the conditional entropy of order n is defined as
$H(x_k|x_{k-1}, \ldots, x_{k-n}) := -\int_{\Re^m} p(.) \ln p(.) dx$

where $p(.) = (x_k|x_{k-1}, \ldots, x_{k-n})$.

This conditional entropy is a measure of the uncertainty about the value of x at time k under the assumption that its n most recent values have been observed. By letting n going to infinity, the conditional entropy of x_k is defined as

$H_c(x_k) := \lim_{n\to\infty} H(x_k|x_{k-1}, \ldots, x_{k-n})$ assuming the limit exists.

Thus the conditional entropy is a measure of the uncertainty about the value of x at time k under the assumption that its entire past is observed. Difference of conditional entropies between the output and input is nothing but the Bode sensitivity integral which equals the summation of logarithms of unstable poles.

For a stationary Markov process, conditional entropy [12] is given by

$H(x_k|x_{k-1}, \ldots, x_{k-n}) = H(x_k|x_{k-1}).$

It is well known that the sensitivity and complementary sensitivity functions represent basic properties of feedback systems such as disturbance attenuation, sensor-noise reduction, and robustness against uncertainties in the plant model. Researchers have worked earlier on the issues of relating the entropy and the Bode Integral and complementary sensitivity. For details refer to the references cited in paper [2].

In [16] the sensitivity integral is interpreted as an entropy integral in the time domain, i.e., no frequency-domain representation is used.

3.4 Fisher Information Matrix

When we cannot get the exact information or the information gets distorted / corrupted / lost due to communication induced noises and the channel properties, we need to estimate the information and hence we need the Estimation Theory. Fundamental limit of Estimation Theory is the *Fisher Information Matrix* (**FIM**) and the related parameter called *Fisher Information* is of paramount importance.

Fisher information is the amount of information that an observable random variable Y carries about an unobservable parameter X upon which the likelihood function of X represented by $L(X) = p(Y; X)$ depends. It is a measure of accuracy in estimating a parameter. Fisher Information Matrix is given by

$FIM_x := -E(\frac{\partial \log p(x,y)}{\partial x})^2$

is the (Bayesian) FIM, where x is the input state, y is the observation at the output and E is the expectation operator .

Also, $\textbf{FIM} = -E\left[\left\{\frac{\partial \ln \rho(f)}{\partial x}\right\}\left\{\frac{\partial \ln \rho(f)}{\partial x}\right\}^T\right]$

$\rho(f)$ being the Gaussian Probability Density Function of random variable f. Recalling the basics of **FIM** we can say that for Gaussian distribution it is related to the variance by *Cramer-Rao Bound* (*CRB*). CRB is a general uncertainty expressing reciprocity between the mean-square-error (e) in an estimate and the Fisher Information *FI* present in the observed data. Mathematically,

$e^2 \geq \frac{1}{FI}$ or $FI \geq \frac{1}{e^2}$

It is a measure of accuracy in estimating a parameter. In other words,

$FI \geq \frac{1}{\sigma^2}$; σ^2 being the variance

Lower the variance, higher the Fisher Information and vice-versa.

Mathematically, curvature or sharpness has got a definite relation with second derivative. To a great extent Fisher Information defines what can and what can not be known about a given physical scenario. **FIM** allows detection of some non-stationary behaviour in situations where Shannon Entropy shows limited dynamics. B. Roy Frieden has characterized **FIM** as a versatile tool to describe the evolution laws of physical systems. By information on an unknown parameter θ contained in a random variable or in its distribution, we mean the extent to which uncertainty regarding the unknown value of θ is reduced as a consequence of an observed value of the random variable. If there is a unique observation with probability 1 corresponding to each value of the parameter, we have a situation where the random variable has the maximum information. On the other hand if the random variable has the same distribution for all values of the parameter, there is no basis for making statements about θ based on an observed value of the random variable. The sensitiveness of a random variable w.r.t a parameter may then be judged by the extent to which its distribution is altered by a change in the value of the parameter. For details readers may refer to [17].

The sensitivity of the random variable w.r.t. changes in the parameter may then be judged by examining the Fisher Information Matrix (**FIM**) as a whole.

For Gaussian Probability Distribution, Shannon Entropy is given by

$H = \ln \sqrt{2\pi e}\sigma,$

Where, Fisher Information is $FI = \frac{1}{\sigma^2}$

$\Rightarrow$ Fisher Information varies faster than Shannon Entropy.

Shannon's measure of information is valid for Hamiltonian Systems [18] whereas Fisher information is valid for arbitrary dynamical systems. Fisher information is more general character than Shannon's or Kullback's. Fisher information is a local measure as it is sensitive to the local behaviour of the probability density. Fisher information decreases under coarse graining (i.e. less precision or quantization accuracy!) and it also decreases with time.

One has to gather relevant information, transmit the information to the relevant agent, process the information, if needed, and then use the information to control the system. The fundamental limitation in information transmission is the achievable information rate (i.e. a fundamental parameter of Information Theory), the fundamental limitation in information processing is the Cramer-Rao Bound (*CRB*) which deals with Fisher Information Matrix (**FIM**) in Estimation Theory, and the fundamental limitation in information utilization is the *Bode Integral* (i.e. a fundamental parameter of Control Theory), seemingly different and usually separately treated, are in fact three sides of the same entity as per [19]. Hence, there is a need of dealing the control problem from information-theoretic point of view.

Even Kalman et al. in their paper [20] have stated that Controllability Gramian Matrix $\mathbf{W}_c$ - matrix is analogous to **FIM** and the determinant $\det \mathbf{W}_c$ is analogous to Shannon Information.

These research work motivated us to investigate some important correlations amongst mutual information, entropy, **FIM** and design control parameters (like controllability gramian) of practical importance rather than just concentrating on stability issues.

4 Information Induced by Controllability Gramian

In general, from the viewpoint of the open-loop system, when the system is unstable, the system amplifies the initial state at a level depending on the size of the unstable poles [14]. Hence, we can say that in systems having more unstable dynamics, the signals contain more information about the initial state. Using this extra information (in terms of mutual information between the control input and the initial state) we can reduce the entropy (uncertainty) and thus rendering the observation of initial state easier.

Suppose that we have a feedback control system in which control signal is sent through a network with limited bandwidth. We will consider the case where the state of the system is measurable and the controller can send the state of the system without error. Under these conditions we may write:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu^*(t); \\ u^*(t) &= -K_c x(t) + u^e(t); \end{aligned} \tag{5}$$

where K_c, $u^*(t)$ and $u^e(t)$, represent, respectively, the feedback controller gain, the applied control input and control error due to quantization noise of limited bandwidth network. In the sequel we are supposing that the control signal errors are caused by Gaussian White Noise which may be given by $u_i^e(t) = \sqrt{D_i}\delta(t)$. So we may write :

$$\begin{aligned} \dot{x}(t) &= (A - BK_c)x(t) + Bu^e(t); \\ u^*(t) &= -K_c x(t) + u^e(t); \end{aligned} \tag{6}$$

or more compactly:

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + Bu^e(t); \\ u^*(t) &= -K_c x(t) + u^e(t); \end{aligned} \tag{7}$$

where, $A_c = A - BK_c$.

The feedback system (7) is a stable one which is perturbed by quantization errors or noises due to the bandwidth limitation.

Lemma : The controllability gramian matrix $\mathbf{W}_c$ of system (7) can related with the information-theoretic entropy H as follows [21]:

$$H(x,t) = \frac{1}{2}\ln\{\det \mathbf{W}_c(\mathbf{D},t)\} + \frac{n}{2}(1+\ln 2\pi) \tag{8}$$

= Average *a priori* uncertainty of the state x at time t for an order n of the system.

Where **D** being the Diagonal Matrix (positive definite symmetric matrix) with D_i being the *i*th diagonal element. Here unit impulse inputs are considered.
and

$$\mathbf{W}_c(\mathbf{D},\tau) = \int_0^\tau e^{A_c t} B\mathbf{D}B^T e^{A_c{}^T t}dt$$

for a system modeled as (7).

Proof of Eqn.(8) :

Referring to [12] we are providing the proof. The input of (7) being Gaussian White Noise, the state of the system is with probability density having mean-value $\bar{x}(t) = e^{A_c t}\bar{x}(0)$ and Covariance Matrix Σ at time t is given by

$$\Sigma = E\left\{(x-\bar{x})(x-\bar{x})^T\right\} = \mathbf{W}_c(\mathbf{D},t).$$

In a more detailed form :

$$x(t) = e^{A_c t}x(0) + \int_0^t e^{A_c t}Bu^e(t)dt$$

$$E\{x(t)\} = E\left\{e^{A_c t}x(0) + \int_0^t e^{A_c t}Bu^e(t)dt\right\}$$

$$\Rightarrow \bar{x}(t) = e^{A_c t}\bar{x}(0)$$

where $\bar{x}(t)$ denotes the mean value of $x(t)$ and Covariance Matrix

$$\Sigma = E\left\{(x-\bar{x})(x-\bar{x})^T\right\}$$

$$\Rightarrow \Sigma = E\left[\left\{e^{A_c t}x(0) + \int_0^\tau e^{A_c t}Bu^e(t)dt - e^{A_c t}x(0)\right\}\left\{e^{A_c t}x(0) + \int_0^\tau e^{A_c t}Bu^e(t)dt - e^{A_c t}x(0)\right\}^T\right]$$

Therefore, $\Sigma = \int_0^\tau e^{A_c t}B\mathbf{D}B^T e^{A_c{}^T t}dt = \mathbf{W}_c(\mathbf{D},\tau)$.

where, $u_i^e(t) = \sqrt{D_i}\delta(t)$ i.e. weighted impulses and $\mathbf{D}$ being the Diagonal Matrix (positive definite symmetric matrix) with D_i being the ith diagonal element. Here, unit impulse inputs are considered.

$$p(x,t) = \frac{1}{(2\pi)^{n/2}\left\{\det \mathbf{W}_c(\mathbf{D},t)\right\}^{1/2}} e^{\left[-1/2\left\{(x-\bar{x})^T \mathbf{W}_c^{-1}(\mathbf{D},t)(x-\bar{x})\right\}\right]} \tag{9}$$

Now, for multidimensional continuous case, entropy (precisely *differential entropy*) of a continuous random variable X with probability density function $f(x)$ (if $\int_{-\infty}^{\infty} f(x)dx = 1$) is defined [**?**] as:

Differential Entropy $h(X) = -\int_S f(x)\ln f(x)dx$;

where the set S for which $f(x) > 0$ is called the *support set* of X.

As in discrete case, the differential entropy depends only on the probability density of the random variable and therefore the differential entropy is sometimes written as $h(f)$ rather than $h(X)$. Here, we call differential entropy as $H(x,t)$ and $f(x)$ as $p(x,t)$ which are correlated as

$$H(x,t) = -\int p(x,t)\ln p(x,t)dx \tag{10}$$

Using equation (9) in equation (10) we get

$$H(x,t) = -\int p(x,t)\left[-\tfrac{1}{2}(x-\bar{x})^T \mathbf{W}_c^{-1}(\mathbf{D},t)(x-\bar{x}) - \ln\left\{(2\pi)^{n/2}\left\{\det \mathbf{W}_c(\mathbf{D},t)\right\}^{1/2}\right\}\right]dx$$

$$H(x,t) = \tfrac{1}{2}E\left[\sum_{i,j}\left\{(X_i-\bar{X}_i)(\mathbf{W}_c^{-1}(\mathbf{D},t))_{ij}(X_j-\bar{X}_j)\right\}\right] + \tfrac{1}{2}\ln\left[(2\pi)^n\left\{\det \mathbf{W}_c(\mathbf{D},t)\right\}\right]$$

$$= \tfrac{1}{2}E\left[\sum_{i,j}\left\{(X_i-\bar{X}_i)(X_j-\bar{X}_j)(\mathbf{W}_c^{-1}(\mathbf{D},t))_{ij}\right\}\right] + \tfrac{1}{2}\ln\left[(2\pi)^n\left\{\det \mathbf{W}_c(\mathbf{D},t)\right\}\right]$$

$$= \tfrac{1}{2}\sum_{i,j}\left[E\left\{(X_j-\bar{X}_j)(X_i-\bar{X}_i)\right\}(\mathbf{W}_c^{-1}(\mathbf{D},t))_{ij}\right] + \tfrac{1}{2}\ln\left[(2\pi)^n\left\{\det \mathbf{W}_c(\mathbf{D},t)\right\}\right]$$

$$= \tfrac{1}{2}\sum_j\sum_i(\mathbf{W}_c(\mathbf{D},t))_{ji}(\mathbf{W}_c^{-1}(\mathbf{D},t))_{ij} + \tfrac{1}{2}\ln\left[(2\pi)^n\left\{\det \mathbf{W}_c(\mathbf{D},t)\right\}\right]$$

$$= \tfrac{1}{2}\sum_j\left\{(\mathbf{W}_c(\mathbf{D},t))(\mathbf{W}_c^{-1}(\mathbf{D},t))\right\}_{jj} + \tfrac{1}{2}\ln\left[\left\{(2\pi)^n\left\{\det \mathbf{W}_c(\mathbf{D},t)\right\}\right]\right.$$

$$= \tfrac{1}{2}\sum_j\mathbf{I}_{jj} + \tfrac{1}{2}\ln\left[\left\{(2\pi)^n\right\}\left\{\det \mathbf{W}_c(\mathbf{D},t)\right\}\right]$$

(Where $\mathbf{I}_{jj}$ is the Identity Matrix)

$$= \tfrac{n}{2} + \tfrac{1}{2}\ln\left[\left\{(2\pi)^n\right\}\left\{\det \mathbf{W}_c(\mathbf{D},t)\right\}\right] = \tfrac{n}{2} + \tfrac{1}{2}\ln\left\{(2\pi)^n\right\} + \tfrac{1}{2}\ln\left\{\det \mathbf{W}_c(\mathbf{D},t)\right\}$$
$$= \tfrac{n}{2} + \tfrac{n}{2}\ln\left\{(2\pi)\right\} + \tfrac{1}{2}\ln\left\{\det \mathbf{W}_c(\mathbf{D},t)\right\}$$

$$\therefore H(x,t) = \tfrac{1}{2}\ln\left\{\det \mathbf{W}_c(\mathbf{D},t)\right\} + \tfrac{n}{2}(1+\ln 2\pi)$$

Based on the equation (8) we can write the entropy reduction as

$$\Delta H(x,t) = \tfrac{1}{2}\Delta[\ln\{\det \mathbf{W}_c(\mathbf{D},t)\}]$$

This shows that the entropy reduction which is same as uncertainty reduction is dependent on Controllability Gramian only. Other term being constant for constant n, gets canceled.

Therefore, $\Delta H(x,t) = H(x(t_1),t_1) - H(x(t_2),t_2)$

$$= \tfrac{1}{2}\ln\{\det \mathbf{W}_{c1}(\mathbf{D}_1,t_1)\} - \tfrac{1}{2}\ln\{\det \mathbf{W}_{c2}(\mathbf{D}_2,t_2)\}$$

$$\Rightarrow \Delta H(x,t) = \frac{1}{2}\ln\left\{\frac{\det \mathbf{W}_{c1}(\mathbf{D}_1,t_1)}{\det \mathbf{W}_{c2}(\mathbf{D}_2,t_2)}\right\} \tag{11}$$

For simplicity we denote $\Delta H(x,t)$ by ΔH, $\mathbf{W}_{c1}(\mathbf{D}_1,t_1)$ by $\mathbf{W}_{c1}$ and $\mathbf{W}_{c2}(\mathbf{D}_2,t_2)$ by $\mathbf{W}_{c2}$.

Therefore, $\Delta H = \frac{1}{2}\ln\left\{\det(\mathbf{W}_{c1}\mathbf{W}_{c2}^{-1})\right\}$

$$\Rightarrow \det(\mathbf{W}_{c1}\mathbf{W}_{c2}^{-1}) = e^{2(\Delta H)}$$

Using the above expression along with the concept of mutual information being the difference of the entropy and the residual conditional entropy i.e. $I(X;U) = H(X) - H(X|U)$ (gain in information is reduction in entropy), we can conclude that Mutual Information $I(X;U)$ between the state X and control input U denoted simply by Shannon Information I_{sh} is given by this ΔH which can be expressed further as

Finally,

$$\det(\mathbf{W}_{c1}\mathbf{W}_{c2}^{-1}) = e^{2(\Delta H)} = e^{2I_{sh}} \tag{12}$$

We may conclude that the uncertainty reduction which is directly related to the $\Delta H(x,t)$ will reduce the variance of the state with respect to the steady-state if $\Delta H(x,t)$ converges to zero. The only influence we have on the control signal is related to that of feedback gain ans scheduling of control signals, to be chosen such that the norm of gramians, represented by $\det(\mathbf{W}_c(D_i,t))$ converge rapidly to their norm to infinity $\det(\mathbf{W}_c(D_\infty,\infty))$.

We can also conclude from (12) that gramian is directly related to the entropy and in turn with Shannon mutual information.

5 Information Induced by FIM and Controllability Gramian

The relation between controllability gramian, Shannon entropy and Shannon Information has already been established in the previous section(s). Referring to the result (12) of previous section of this chapter we can say that

$$\det(\mathbf{W}_{c1}\mathbf{W}_{c2}^{-1}) = e^{2(\Delta H)} = e^{2I_{sh}} \tag{13}$$

where $\mathbf{W}_{c1}, \mathbf{W}_{c2}$ denote controllability gramians and H and I_{sh} denote the Shannon Entropy and Shannon Information respectively.

Fisher Information is also a measure of the intrinsic accuracy of a distribution and in [19] it is shown that Mutual Information I_{sh} is related to the Fisher Information $\det(\mathbf{FIM})$ as follows :

$I_{sh} = \frac{1}{2} \ln \{\det(\mathbf{FIM})\}$

Therefore,

$$\det(\mathbf{FIM}) = e^{2I_{sh}} \tag{14}$$

Hence, using equations (14) and (13) we can conclude that

$$\det(\mathbf{FIM}) = \det(\mathbf{W}_{c1}\mathbf{W}_{c2}^{-1}) \tag{15}$$

Thus with the reduction of controllability gramian norm from $\mathbf{W}\_c1$ to $\mathbf{W}\_c2$ there is an increase in **FIM** and higher the reduction of controllability gramian, more is the Fisher information associated with. Since Cramer-Rao Bound implies that Fisher information is the reciprocal of the variance as its lower limit and controllability gramian is same as variance for a white Gaussian noise as input, we can conclude that our result is in agreement with the already established theories.

From control theory the control energy expression is given by

$$\epsilon^2\_c \equiv \mathbf{x}^T \mathbf{W}^{-1}\_c\mathbf{x}$$

By controllability one means that with minimum energy the state of a system can be steered towards the target in a finite time. Minimizing the energy means minimizing $\mathbf{W}^{-1}\_c$ i.e. maximizing $\mathbf{W}\_c$ in the sense of a given matrix norm. The use of enriched Fisher information which has got a direct relation with variance and controllability gramian helps us to achieve the target state with greater ease and prudent transfer of information which is of great interest in communication and power constrained embedded systems.

An important observation in the above energy expression (control-theoretic) is that the locus is an ellipsoid with axis direction given by eigenvector of $\mathbf{W}\_c^{-1}$ and length proportional to eigenvalue. Smaller the ellipsoid is, more controllable the system is. Similar analogy can be drawn in information theory [13] as well :

$$\Delta I(Y;X) = \frac{1}{2}(\Delta X)^T \mathbf{FIM}(\Delta X)$$

where superscript T denoting Transpose, I, X and Y being the mutual information, input and output processes respectively with Δ being the change in respective parameters.

It is shown in [22] that the derivative of the mutual information (nats) w.r.t the signal-to-noise ratio (*snr*) is equal to half the Minimum Mean Square Error (MMSE), regardless of the input statistics i.e.

$$\frac{dI}{dsnr}(snr) = \frac{1}{2}MMSE(snr) \tag{16}$$

This relationship holds for both scalar and vector signals, as well as for discrete-time and continuous-time non-causal MMSE estimation. MMSE reduces with *snr* and Shannon Information increases with *snr* for Gaussian and Binary Channel as well. Hence, by

reducing quantization noises (in terms of Quantization Precision /Accuracy) in [23] at the time when the system approaches steady-state we can increase the *snr* and with increase in *snr* we can have increased Mutual information and thus less uncertainty. Here, we provide the theoretical justification of the simulation work done before in [23].

6 Conclusions

This chapter has addressed some new ideas based on our paper [2] concerning the relation between control design and information theory. Since the distributed embedded system or networked control system has communication constraints due to limited bandwidth or quantization noises, we must have to adopt a policy of resource allocation which enhances the information transmitted. This may be done possible if we know the characteristics of the networks, the bandwidth constraints and that of the dynamical system under study.

As demonstrated here, the controllability gramian constitutes a metric of information theoretic entropy w.r.t the noises induced by quantization. Reduction of these noises is equivalent to the design methods proposing a reduction of the controllability gramian norm.

Since zoom - in (near the target) is equivalent to reducing the entropy (contracting the uncertainty domain) and zoom - out (away from the target) is equivalent to increasing the entropy (expanding the uncertainty domain), work in this direction to propose an information-theoretic analysis of the zooming algorithm proposed in [23] would be dealt in future.

Controllability gramian is, in some sense, a measure of energy. Higher the controllability gramian ($\mathbf{W}_c$), lower is the energy associated with it and hence lower the uncertainty (randomness, like the Brownian motion or the physical notion of energy and entropy). Hence, entropy reduction is associated with controllability gramian reduction and entropy reduction is at the cost of gain in Shannon mutual information which is established in this chapter vide equation (12). This chapter has addressed some new ideas (based on our work [24]) concerning the relation between Control Theoretic parameter (Controllability Gramian) and **FIM** of Estimation Theory which, in turn, linked to Information Theory. Since the DES or Networked Embedded Control System (NECS) has communication constraints due to limited bandwidth or noises, we must have to adopt a policy of resource allocation [25] which enhances the information transmitted.

It is demonstrated that the controllability gramian norm constitutes a metric of information-theoretic entropy w.r.t the noises induced by quantization. In fact, physical interpretation of controllability Gramian is : "If the input to the system is white Gaussian noise, then controllability gramian is the covariance of the state". Reduction of these noises is equivalent to the design methods proposing a reduction of the controllability gramian norm which is related to the Fisher information matrix as well.

7 Future Work

Future work in this direction would be simulation and / or experimental verification of the theoretical approaches we attempted to project in this chapter.

References

1. Sandberg, H., Bernhardsson, B.: A Bode Sensitivity Integral for Linear Time-Periodic Systems. IEEE Trans. Autom. Control 50(12) (December 2005)
2. Roy, P., Çela, A., Hamam, Y.: An Information-Theoretic View of Control. In: 6th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2009), Milan, Italy, Milan (July 2009)
3. Zang, G., Iglesias, P.: Nonlinear extension of Bode's integral based on an information theoretic interpretation. Systems and Control Letters 50, 11–29 (2003)
4. Mehta, P., Vaidya, U., Banaszuk, A.: Markov Chains, Entropy, and Fundamental Limitations in Nonlinear Stabilization. In: 45th IEEE Conference on Decision & Control, San Diego, USA (December 2006)
5. Antsaklis, P., Baillieul, J.: Special Issue on the Technology of Networked Control Systems. Proceedings of the IEEE 95(1) (January 2007)
6. Nair, G., Evans, R.: Stabilizability of Stochastic Linear Systems with Finite Feedback Data Rates. SIAM J. Control Optim. 43(2), 413–436 (2004)
7. Tatikonda, S., Mitter, S.: Control under Communication Constraints. IEEE Trans. Autom. Control 49(7), 1056–1068 (2004)
8. Martins, N., Dahleh, M., Doyle, J.: Fundamental Limitations of Disturbance Attenuation in the presence of Side Information. IEEE Trans. Autom. Control 52(1), 56–66 (2007)
9. Bode, H.: Network Analysis and Feedback Amplifier Design, Van Nostrand D (1945)
10. Iglesias, P.: Trade-offs in Linear Time-varying Systems: An Analogue of Bode's Sensitivity Integral. Automatica 37(10), 1541–1550 (2001)
11. Shannon, C.: The Mathematical Theory of Communication. Bell Systems Tech. J. 27, 379–423 (1948); 623– 656 (October 1948)
12. Cover, T.: Elements of Information Theory, 2nd edn. John Wiley and Sons, Chichester (2006)
13. Middleton, D.: An Introduction to Statistical Communication Theory. McGraw-Hill Publication, New York (1960)
14. Okano, K., Hara, S., Ishii, H.: Characterization of a complementary sensitivity property in feedback control: An information theoretic approach. In: Proc. of the 17th IFAC-World Congress, Seoul, Korea, July 6-11, 2008, pp. 5185–5190 (2008)
15. Wu, B., Jonckheere, E.: A Simplified Approach to Bode's Theorem for Continuous-time and Discrete-time Systems. IEEE Trans. Autom. Control 37(11) (November 1992)
16. Iglesias, P.: Logarithmic integrals and system dynamics: An analogue of Bode's sensitivity integral for continuous-time time-varying systems. Linear Alg. Appl. 343-344(2), 451–471 (2002)
17. Rao, C.: Linear Statistical Inference and its Applications. John Wiley & Sons, Chichester (1965)
18. Plastino, A., Casas, M., Plastino, A.: Fisher's information, Kullback's measure and H-theorems. Physics Letters A 24(6), 498–504 (1998)
19. Liu, J., Elia, N.: Convergence of Fundamental Limitations in Information, Estimation and Control. In: Proc. of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA (December 2006)
20. Kalman, R., Ho, Y., Narendra, K.: Controllability of linear dynamical systems. Contributions to Differential Equations 1(2), 189–213 (1963)
21. Mitra, D.: W-matrix and the Geometry of Model Equivalence and Reduction. Proc. of the IEE 116(6), 1101–1106 (1969)

22. Guo, D., Shamai(Shitz), S., Verdú, S.: Mutual Information and Minimum Mean-Square Error in Gaussian Channels. IEEE Trans. on Information Theory 51(4) (April 2005)
23. Ben Gaid, M., Çela, A.: Trading Quantization Precision for Sampling Rates in Networked Systems with Limited Communication. In: Proc. of the 45th IEEE Conference on Decision & Control, San Diego, CA, USA (December 2006)
24. Roy, P., Çela, A., Hamam, Y.: On the Relation of FIM and Controllability Gramian. In: Proc. of IEEE International Symposium on Industrial Embedded Systems (SIES 2009), EPFL, Lausanne, Switzerland, July 8–10 (2009)
25. Roy, P.: Analysis & Design of Control for Distributed Embedded Systems under Communication Constraints. Ph.D. Thesis, Université Paris-Est. (December 2009)

VAR Based State–space Structures: Realization, Statistics and Spectral Analysis

Vasilis K. Dertimanis and Dimitris V. Koulocheris

National Technical University of Athens, Mechanical Design & Automatic Control Department
Iroon Politechniou 9, 157 80, Athens, Greece

Abstract. The effective representation of multivariate time–series in terms of their structural indices is faced in this study, leading to the development of a corresponding scheme that is based in the interconnection between VAR and state–space models. To this, a specific state equation that retains a state matrix identical to the companion matrix of the original VAR polynomial is realized, uncovering many "hidden" information of the initial process. In light of this realization, it is shown how closed form expressions for the Green function, the covariance generating function and the spectral density can be derived through the *spectrum* of the state matrix, thus allowing assessment and quantification (via the notion of dispersion analysis) of every structural mode. A Yule–Walker based estimate is also provided, which applies directly to the state equation. A structural system with two degrees of freedom and closely spaced modes serves as an application of the novel scheme, using Monte Carlo analysis.

Keywords: Time–series, Vector autoregressive, State–space, Green function, Covariance function, Dispersion analysis, Spectral density, Estimation.

1 Introduction

The analysis of vector time–series, generally referring to the determination of the dynamics that govern the performance of a system under unobservable excitations/disturbances, has been a subject of constant development for many years, as part of the broader system identification framework. Relative applications are extended from econometrics [3,16], to dynamics [15,11], vibration and modal analysis [18,10], control [9,12] and fault diagnosis [4].

The study of vector time–series can be assessed from a variety of viewpoints, with respect to the application of interest. These include simulation, prediction and extraction of structural information. Yet, while in the first two areas the interrelation of the corresponding time–series structures, such as the VAR one (or the VARX and the VARMAX, under the availability of input information), to equivalent state–space models has been studied extensively [7,2,16], not much have been done in the third [13,5], from where it appears that state–space realizations may provide significant advantages, regarding structural estimation, with respect to other approaches [8].

This work attempts to provide a unified framework for the structural representation of vector time–series, by means of VAR models and their corresponding state–space realizations. Based on the fact that every VAR structure of order n (notated from now on as VAR(n) structure) can be expressed as an equivalent (and non–unique) VAR(1) one,

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 301–315.
springerlink.com

a corresponding state–space model is developed. This specific model qualifies, over other possible realizations, for having a state matrix that coincides with the VAR(n) matrix polynomial. It turns out that the spectrum of this matrix (not to be confused with the spectral density of a time–series) has all the structural information about the system that generates the time–series "hidden" in its spectrum. Consequently, by taking advantage of standard results of matrix algebra, closed form expressions for the Green function, the covariance generating function and the spectral density of the state and output vectors are derived, which reveal the dependence to the structural modes. In addition, the impact of every such mode in the total energy of the state and output equations is assessed, by following a methodology that has been recorded in the literature as *dispersion analysis* [14]. Yet, unlike other dispersion analysis estimators [6] that rely on the calculation of the residues of an estimated transfer function, the one addressed here is also based on the spectrum of the state matrix and includes the effect of the stochastic excitation's covariance. The applied framework is independent to the estimation method selected, however, a state matrix estimate that is based on the Yule–Walker equations (resulting thus in a minimum phase (stable) estimator) is provided, the effectiveness of which is investigated through a simple structural system, with two degrees of freedom, that is characterized by a pair of closely spaced modes.

The rest of the text is organized as follows: in Sec. 2 the VAR model is outlined and the reduction to state–space is introduced. Section 3 investigates the properties of the VAR(n) state–space realization, regarding the Green function, the covariance generating function and the spectral density matrix, including dispersion analysis, and formulates the relations with the original VAR(n) model. Section 4 deals with the estimation of the state matrix. The Monte Carlo analysis of the structural system takes place in section 5 and the main characteristics of the introduced method are concluded in section 6.

2 The VAR State–space Realization

2.1 Preliminaries

Let $\mathbf{Y}[t] = \left[y_1[t]\ y_2[t]\ ...\ y_s[t]\right]^T$ denote a s–dimensional vector time–series of zero mean random variables[1]. Under the stationarity assumption [1], $\mathbf{Y}[t]$ can be described by a finite order VAR(n) model of the following form:

$$\mathbf{Y}[t] + \mathbf{A}_1 \cdot \mathbf{Y}[t-1] + ... + \mathbf{A}_n \cdot \mathbf{Y}[t-n] = \mathbf{Z}[t] \tag{1}$$

In the above equation n is the order of the VAR process, $\mathbf{A}_j$ designate the $[s \times s]$ AR matrices and $\mathbf{Z}[t]$ describes a vector white noise process with zero mean and covariance function,

$$\boldsymbol{\Gamma}_{\mathbf{Z}}[h] \equiv E\left\{\mathbf{Z}[t+h] \cdot \mathbf{Z}^T[t]\right\} = \begin{cases} \boldsymbol{\Sigma}_{\mathbf{Z}} & h = 0 \\ \mathbf{0} & h \neq 0 \end{cases} \tag{2}$$

where $\boldsymbol{\Sigma}_{\mathbf{Z}}$ is a non–singular (and generally non–diagonal) matrix.

[1] Throughout the text, quantities in the brackets shall notate discrete–time units (or time lags, in the case of covariance functions) and hats shall notate estimators / estimates. $E\{\cdot\}$ shall notate expectation.

Taking advantage of the backshift operator q, defined such that $q^{-k}\cdot\mathbf{Y}[t]=\mathbf{Y}[t-k]$, the VAR($n$) structure can be compactly written as,

$$\mathbf{A}(q)\cdot\mathbf{Y}[t]=\mathbf{Z}[t] \tag{3}$$

where $\mathbf{A}(q)$ is the $[s\times s]$ AR matrix polynomial:

$$\mathbf{A}(q)=\mathbf{I}_s+\mathbf{A}_1\cdot q^{-1}+...+\mathbf{A}_n\cdot q^{-n} \tag{4}$$

2.2 Reduction to State–space

Any VAR(n) process of Eq. 1 can be transformed to an equivalent VAR(1) structure [16]. By defining the $[n\cdot s\times 1]$ vectors,

$$\boldsymbol{\Xi}[t]=\begin{bmatrix}\mathbf{Y}[t-n+1]\\ \vdots\\ \mathbf{Y}[t-1]\\ \mathbf{Y}[t]\end{bmatrix}\quad \boldsymbol{N}[t]=\begin{bmatrix}\mathbf{0}\\ \vdots\\ \mathbf{0}\\ \mathbf{Z}[t]\end{bmatrix} \tag{5}$$

and the $[n\cdot s\times n\cdot s]$ and $[s\times n\cdot s]$ matrices,

$$\mathbf{F}=\begin{bmatrix}\mathbf{O}_s & \mathbf{I}_s & \dots & \mathbf{O}_s\\ \mathbf{O}_s & \mathbf{O}_s & \dots & \mathbf{O}_s\\ \dots & \dots & \ddots & \vdots\\ \mathbf{O}_s & \mathbf{O}_s & \dots & \mathbf{O}_s\\ -\mathbf{A}_n & -\mathbf{A}_{n-1} & \dots & -\mathbf{A}_1\end{bmatrix} \tag{6}$$

$$\mathbf{C}=\begin{bmatrix}\mathbf{O}_s\ \mathbf{O}_s\ \dots\ \mathbf{I}_s\end{bmatrix} \tag{7}$$

respectively, Eq. 1 can take the following form:

$$\boldsymbol{\Xi}[t]=\mathbf{F}\cdot\boldsymbol{\Xi}[t-1]+\boldsymbol{N}[t] \tag{8}$$

$$\boldsymbol{Y}[t]=\mathbf{C}\cdot\boldsymbol{\Xi}[t] \tag{9}$$

Equations 8–9 illustrate the state–space realization of the VAR(n) structure of Eq. 1. Naturally, the state–space model consists of a state equation (Eq. 8), in which $\mathbf{F}$ is the state matrix, and an observation equation (Eq. 9) that relates the original s–variate time–series $\mathbf{Y}[t]$ to the state vector, $\boldsymbol{\Xi}[t]$, by means of the output matrix $\mathbf{C}$. Obviously, the state equation can be viewed as a VAR(1) model, in which $\boldsymbol{\Xi}[t]$ is a well–defined stationary stochastic process and $\mathbf{N}[t]$ has properties similar to that of $\mathbf{Z}[t]$, as it will become clear at the following.

It must be noted that the state–space realization of Eq. 1 is not unique [13]. In fact they exist infinitely many pairs $\{\mathbf{F},\mathbf{C}\}$ that can describe $\mathbf{Y}[t]$ in terms of Eqs. 8–9,

since any transformation of the state vector by a non–singular $[n{\cdot}s \times n{\cdot}s]$ matrix $\mathbf{T}$ leads to new state equation, in which the state matrix $\mathbf{T}{\cdot}\mathbf{F}{\cdot}\mathbf{T}^{-1}$ is similar to $\mathbf{F}$ and preserves its eigenvalues [17]. Yet, Eq. 8 has a very important property: the state matrix $\mathbf{F}$, as defined in Eq. 6, is the block companion matrix of the AR polynomial described by Eq. 4 and includes all the structural information of interest, regarding the process that generates the s–variate time–series $\mathbf{Y}[t]$.

3 State–space Realization Properties

3.1 Noise

From Eqs. 5, 7, it holds that,

$$\mathbf{N}[t] = \mathbf{C}^T{\cdot}\mathbf{Z}[t] \tag{10}$$

thus, $\mathbf{N}[t]$ is characterized by zero mean and covariance matrix given by,

$$\begin{aligned} \boldsymbol{\Gamma}_{\mathbf{N}}[h] &\equiv E\Big\{\mathbf{N}[t+h]{\cdot}\mathbf{N}^T[t]\Big\} = E\Big\{\mathbf{C}^T{\cdot}\mathbf{Z}[t+h]{\cdot}\mathbf{Z}^T[t]{\cdot}\mathbf{C}\Big\} \\ &= \mathbf{C}^T{\cdot}E\Big\{\mathbf{Z}[t+h]{\cdot}\mathbf{Z}^T[t]\Big\}{\cdot}\mathbf{C} = \mathbf{C}^T{\cdot}\boldsymbol{\Gamma}_{\mathbf{Z}}[h]{\cdot}\mathbf{C} \end{aligned} \tag{11}$$

leading to,

$$\boldsymbol{\Gamma}_{\mathbf{N}}[h] = \begin{cases} \boldsymbol{\Sigma}_{\mathbf{N}} & h = 0 \\ \mathbf{0} & h \neq 0 \end{cases} \tag{12}$$

where $\boldsymbol{\Sigma}_{\mathbf{N}} = \mathbf{C}^T{\cdot}\boldsymbol{\Sigma}_{\mathbf{Z}}{\cdot}\mathbf{C}$.

3.2 State Vector

The Green Function. In view of Eq. 8, the state equation can be written as an infinite VMA,

$$\boldsymbol{\Xi}[t] = \sum_{k=0}^{\infty} \mathbf{F}^k{\cdot}\boldsymbol{N}[t-k] \tag{13}$$

which is a multivariate generalization of Wold's theorem [1]. The Green function is defined as:

$$\mathbf{H}_{\boldsymbol{\Xi}}(q) \equiv \sum_{k=0}^{\infty} \mathbf{H}_{\boldsymbol{\Xi}}[k]{\cdot}q^{-k} = \sum_{k=0}^{\infty} \mathbf{F}^k{\cdot}q^{-k} \tag{14}$$

Under the assumption that $\mathbf{F}$ retains a complete set of eigenvalues $\big\{\lambda_1, \lambda_2, ..., \lambda_{n\cdot s}\big\}$, it can be expressed as,

$$\mathbf{F} = \sum_{j=1}^{n\cdot s} \mathbf{G}_j{\cdot}\lambda_j \tag{15}$$

where $\mathbf{G}_j$ are the, so called, spectral projectors of $\mathbf{F}$ (refer to Meyer [17], Chapter 7, for a rigorous analysis on the spectral properties of square matrices). The substitution of Eq. 15 to Eq. 14, using the fact that $\mathbf{G}_j^k = \mathbf{G}_j$ and $\mathbf{G}_i \cdot \mathbf{G}_j = 0, \;\; i \neq j$, yields,

$$\mathbf{H}_{\boldsymbol{\Xi}}(q) = \sum_{k=0}^{\infty} \Big[\sum_{j=1}^{n \cdot s} \mathbf{G}_j \cdot \lambda_j \Big]^k \cdot q^{-k} = \sum_{j=1}^{n \cdot s} \frac{1}{1 - \lambda_j \cdot q^{-1}} \cdot \mathbf{G}_j \tag{16}$$

so that the Green function is decomposed into $n \cdot s$ first order terms. In addition, from Eq. 14, the coefficients (weights) of the Green function can be expressed in a closed form as,

$$\mathbf{H}_{\boldsymbol{\Xi}}[k] \equiv \mathbf{F}^k = \sum_{j=1}^{n \cdot s} \mathbf{G}_j \cdot \lambda_j^k \tag{17}$$

in terms of the spectrum of $\mathbf{F}$. In general, $\mathbf{H}_{\boldsymbol{\Xi}}[k]$ has a decaying performance, characterized by a mixture of damped exponentials and cosines, as for example in vibrating systems, where the eigenvalues λ_j often appear in complex conjugate pairs. Furthermore it holds that [17]:

$$\mathbf{H}_{\boldsymbol{\Xi}}[0] = \sum_{j=1}^{n \cdot s} \mathbf{G}_j = \mathbf{I} \tag{18}$$

The Covariance Generating Function. The covariance generating function is defined as,

$$\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}(q) \equiv \sum_{k=-\infty}^{\infty} \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[k] \cdot q^{-k} \tag{19}$$

where $\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[k]$ is the covariance matrix related to the Wold decomposition of the state equation (as expressed in VMA form, Eq. 13), given by:

$$\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[k] = \sum_{i=0}^{\infty} \mathbf{H}_{\boldsymbol{\Xi}}[i+k] \cdot \boldsymbol{\Sigma}_{\mathbf{N}} \cdot \mathbf{H}_{\boldsymbol{\Xi}}^T[i] \tag{20}$$

Substitution in Eq. 19 implies:

$$\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}(q) = \sum_{k=-\infty}^{\infty} \sum_{i=0}^{\infty} \mathbf{H}_{\boldsymbol{\Xi}}[i+k] \cdot \boldsymbol{\Sigma}_{\mathbf{N}} \cdot \mathbf{H}_{\boldsymbol{\Xi}}^T[i] \cdot q^{-k} \tag{21}$$

Setting $i + k = j$ and assuming causality, so that $\mathbf{H}_{\boldsymbol{\Xi}}[k] = \mathbf{O}_s$ for $k < 0$, Eq. 21 becomes,

$$\begin{aligned} \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}(q) &= \sum_{j=-\infty}^{\infty} \sum_{i=0}^{\infty} \mathbf{H}_{\boldsymbol{\Xi}}[j] \cdot \boldsymbol{\Sigma}_{\mathbf{N}} \cdot \mathbf{H}_{\boldsymbol{\Xi}}^T[i] \cdot q^{i-j} = \sum_{j=-\infty}^{\infty} \mathbf{H}_{\boldsymbol{\Xi}}[j] \cdot \boldsymbol{\Sigma}_{\mathbf{N}} \cdot \sum_{i=0}^{\infty} \mathbf{H}_{\boldsymbol{\Xi}}^T[i] \cdot q^{i-j} \\ &= \sum_{j=0}^{\infty} \mathbf{H}_{\boldsymbol{\Xi}}[j] \cdot q^{-j} \cdot \boldsymbol{\Sigma}_{\mathbf{N}} \cdot \sum_{i=0}^{\infty} \mathbf{H}_{\boldsymbol{\Xi}}^T[i] \cdot q^{i} = \mathbf{H}_{\boldsymbol{\Xi}}(q) \cdot \boldsymbol{\Sigma}_{\mathbf{N}} \cdot \mathbf{H}_{\boldsymbol{\Xi}}^T(q^{-1}) \end{aligned} \tag{22}$$

or, using the result of Eq. 16,

$$\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}(q) = \sum_{j=1}^{n\cdot s} \frac{1}{1-\lambda_j \cdot q^{-1}} \cdot \mathbf{G}_j \cdot \boldsymbol{\Sigma}_{\mathbf{N}} \cdot \sum_{j=1}^{n\cdot s} \frac{1}{1-\lambda_j \cdot q} \cdot \mathbf{G}_j^T \tag{23}$$

so that the covariance generating function of the state equation is also expressed in terms of the state and the noise covariance matrices.

Of particular importance is the covariance matrix of the state vector, $\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[k]$. Combining Eqs. 17 and 20,

$$\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[h] = \sum_{k=0}^{\infty} \mathbf{F}^{k+h} \cdot \boldsymbol{\Sigma}_{\mathbf{N}} \cdot \left[\mathbf{F}^k\right]^T \tag{24}$$

that leads to (the proof is given in [5]),

$$\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[h] = \sum_{j=1}^{n\cdot s} \mathbf{D}_j \cdot \lambda_j^h \tag{25}$$

where,

$$\mathbf{D}_j = \mathbf{G}_j \cdot \boldsymbol{\Sigma}_{\mathbf{N}} \cdot \sum_{m=1}^{n\cdot s} \frac{\mathbf{G}_m^T}{1-\lambda_j \cdot \lambda_m} \tag{26}$$

Equation 25 has some important features. Firstly, as becomes directly evident, it has the same form as the weights of the Green function. Secondly, it describes the covariance matrix in terms of the spectral properties of the state matrix (plus the noise covariance), which, as already mentioned, contains all the information about the dynamics that produce the state vector and, thus, $\mathbf{Y}[t]$. This fact leads to a third crucial feature: for $h = 0$, Eq. 25 yields:

$$\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[0] = \mathbf{D}_1 + \mathbf{D}_2 + \cdots + \mathbf{D}_{n\cdot s} \tag{27}$$

Recalling that $\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[0]$ can be treated as the multivariate equivalent of the variance (in fact its diagonal elements are the variances of each entry of $\boldsymbol{\Xi}[t]$), Eq. 27 can be used as a direct measure of the significance that every eigenvalue has, in the *total energy* of the vector time–series. This leads to the notion of *dispersion analysis*, originated in the work of Lee and Fassois [14], in which the estimated modal characteristics of a vibrating system are qualified against some predefined thresholds, under a broadband stochastic excitation environment. Since the eigenvalues of the state matrix may be both real (say k_1) and complex conjugate pairs (say $2 \cdot k_2$, so that $k_1 + 2 \cdot k_2 = n \cdot s$), the same applies to the matrices $\mathbf{D_j}$ (see Eq. 26). Thus, by following a similar procedure to that described in Fassois [6], Eq. 27 can be written as,

$$\begin{aligned} \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[0] &= \sum_{k=1}^{k_1} \mathbf{D}_k + \sum_{k=k_1+1}^{k_1+k_2} \mathbf{D}_k + \mathbf{D}_k^* \\ &= \sum_{k=1}^{k_1} \mathbf{M}_k^r + \sum_{k=k_1+1}^{k_1+k_2} \mathbf{M}_k^c \end{aligned} \tag{28}$$

where $\mathbf{M}_k^r$ and $\mathbf{M}_k^c$ denote the real and complex dispersions, respectively. Correspondingly, within the content of the $\{i, j\}$ entry of the zero lag covariance matrix, the percentage dispersion matrices are defined as,

$$\boldsymbol{\Delta}_k^* = [\delta_{k_{ij}}^*], \quad {}^* : r, c \tag{29}$$

where,

$$\delta_{k_{ij}}^* = 100 \cdot \frac{\mathbf{M}_{k_{ij}}^*}{\boldsymbol{\Gamma}_{\boldsymbol{\Xi}_{ij}}[0]}\% \tag{30}$$

while any matrix norm can be utilized, in order to assess the significance of a specific mode in the total vector time–series energy.

The Spectral Density. For time–series sampled at a frequency F_s, the spectral density matrix of the state vector is given by the Fourier transform of the covariance matrix,

$$\mathbf{S}_{\boldsymbol{\Xi}}(f) = \sum_{k=-\infty}^{\infty} \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[k] \cdot e^{-j \cdot \frac{2 \cdot \pi \cdot f}{F_s} \cdot k} \tag{31}$$

where f is the frequency in Hz. It is worth noting that the above expression for the spectral density matrix retains a strong relation to the one of the covariance generating function. In fact, using Eq. 19,

$$\mathbf{S}_{\boldsymbol{\Xi}}(f) = \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}(q = e^{j \cdot \frac{2 \cdot \pi \cdot f}{F_s}}) \tag{32}$$

thus, by substituting the resultant formula for $\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}(q)$, as described by Eq. 23, the following relation occurs:

$$\mathbf{S}_{\boldsymbol{\Xi}}(f) = \sum_{j=1}^{n \cdot s} \frac{1}{1 - \lambda_j \cdot e^{-j \cdot \frac{2 \cdot \pi \cdot f}{F_s}}} \cdot \mathbf{G}_j \cdot \boldsymbol{\Sigma}_{\mathbf{N}} \cdot \sum_{j=1}^{n \cdot s} \frac{1}{1 - \lambda_j \cdot e^{j \cdot \frac{2 \cdot \pi \cdot f}{F_s}}} \cdot \mathbf{G}_j^T \tag{33}$$

A more efficient expression for the spectral density matrix can be derived directly from Eq. 31, by replacing the covariance matrix via Eq. 25:

$$\mathbf{S}_{\boldsymbol{\Xi}}(f) = \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[0] + \sum_{k=-\infty}^{-1} \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[k] \cdot e^{-j \cdot \frac{2 \cdot \pi \cdot f}{F_s} \cdot k} + \sum_{k=1}^{\infty} \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[k] \cdot e^{-j \cdot \frac{2 \cdot \pi \cdot f}{F_s} \cdot k} \tag{34}$$

Since $\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[-k] = \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}^T[k]$, Eq. 34 becomes,

$$\mathbf{S}_{\boldsymbol{\Xi}}(f) = \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[0] + \sum_{k=1}^{\infty} \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}^T[k] \cdot e^{j \cdot \frac{2 \cdot \pi \cdot f}{F_s} \cdot k} + \sum_{k=1}^{\infty} \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[k] \cdot e^{-j \cdot \frac{2 \cdot \pi \cdot f}{F_s} \cdot k} \tag{35}$$

Using Eq. 25,

$$\begin{aligned} \mathbf{S}_{\boldsymbol{\Xi}}(f) = \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[0] + \sum_{k=1}^{\infty} \sum_{m=1}^{n \cdot s} \mathbf{D}_m^T \cdot \lambda_m^k \cdot e^{j \cdot \frac{2 \cdot \pi \cdot f}{F_s} \cdot k} + \\ \sum_{k=1}^{\infty} \sum_{m=1}^{n \cdot s} \mathbf{D}_m \cdot \lambda_m^k \cdot e^{-j \cdot \frac{2 \cdot \pi \cdot f}{F_s} \cdot k} \end{aligned} \tag{36}$$

Setting $a = e^{j\cdot\frac{2\cdot\pi\cdot f}{F_s}}$ for ease in manipulation, it follows that:

$$\begin{aligned}\mathbf{S}_{\boldsymbol{\Xi}}(f) &= \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[0] + \sum_{m=1}^{n\cdot s}\mathbf{D}_m^T\cdot\sum_{k=1}^{\infty}\lambda_m^k\cdot a^k + \sum_{m=1}^{n\cdot s}\mathbf{D}_m^T\cdot\sum_{k=1}^{\infty}\lambda_m^k\cdot a^{-k}\\ &= \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[0] + \sum_{m=1}^{n\cdot s}\mathbf{D}_m^T\cdot(\lambda_m\cdot a)\cdot\sum_{k=0}^{\infty}(\lambda_m\cdot a)^k\\ &\quad + \sum_{m=1}^{n\cdot s}\mathbf{D}_m\cdot(\lambda_m\cdot a^{-1})\cdot\sum_{k=0}^{\infty}(\lambda_m\cdot a^{-1})^k\\ &= \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[0] + \sum_{m=1}^{n\cdot s}\mathbf{D}_m^T\cdot\frac{\lambda_m\cdot a}{1-\lambda_m\cdot a} + \mathbf{D}_m\cdot\frac{\lambda_m\cdot a^{-1}}{1-\lambda_m\cdot a^{-1}}\end{aligned} \tag{37}$$

By replacing a,

$$\mathbf{S}_{\boldsymbol{\Xi}}(f) = \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[0] + \sum_{m=1}^{n\cdot s}\frac{\lambda_m\cdot e^{-j\cdot\frac{2\cdot\pi\cdot f}{F_s}}}{1-\lambda_m\cdot e^{-j\cdot\frac{2\cdot\pi\cdot f}{F_s}}}\cdot\mathbf{D}_m + \frac{\lambda_m\cdot e^{j\cdot\frac{2\cdot\pi\cdot f}{F_s}}}{1-\lambda_m\cdot e^{j\cdot\frac{2\cdot\pi\cdot f}{F_s}}}\cdot\mathbf{D}_m^T \tag{38}$$

with $\mathbf{D}_m$ calculated by Eq. 26. Since the spectral density matrix is Hermitian, $\mathbf{S}_{\boldsymbol{\Xi}}(-f) = \mathbf{S}_{\boldsymbol{\Xi}}^T(f)$, it is sufficient to calculate and illustrate its auto and cross spectral density entries in the $[0 \quad F_s/2]$ frequency interval, according to:

$$\mathbf{S}_{\boldsymbol{\Xi}}^*(f) = \begin{cases}2\cdot\mathbf{S}_{\boldsymbol{\Xi}}(f) & ,0 < f \leq \frac{F_s}{2}\\ \mathbf{S}_{\boldsymbol{\Xi}}(f) & ,f = 0\end{cases} \tag{39}$$

3.3 Output Time–Series

The preceding analysis explored the advantages of the state equation and led to closed form representations for the Green function, the covariance generating function and the spectral density matrix, which are exclusively depend on the spectrum of the state matrix $\mathbf{F}$ and, where applicable, the state noise covariance $\boldsymbol{\Sigma}_{\mathbf{N}}$. Naturally, there exist strong connections between these quantities and the corresponding ones of the s–variate time–series $\mathbf{Y}[t]$, via the output equation of the state–space model. These are explored in the following.

Green Function. The substitution of Eq. 13 to Eq. 9, in view of Eq. 17, produces,

$$\begin{aligned}\mathbf{Y}[t] = \mathbf{C}\cdot\boldsymbol{\Xi}[t] &= \mathbf{C}\cdot\sum_{k=1}^{\infty}\mathbf{H}_{\boldsymbol{\Xi}}[k]\cdot\boldsymbol{N}[t-k]\\ &= \mathbf{C}\cdot\sum_{k=1}^{\infty}\mathbf{H}_{\boldsymbol{\Xi}}[k]\cdot q^{-k}\cdot\boldsymbol{N}[t]\\ &= \mathbf{C}\cdot\mathbf{H}_{\boldsymbol{\Xi}}(q)\cdot\mathbf{C}^T\cdot\boldsymbol{Z}[t] \equiv \mathbf{H}_{\mathbf{Y}}(q)\cdot\boldsymbol{Z}[t]\end{aligned} \tag{40}$$

thus, the Green function of original VAR(n) model is,

$$\mathbf{H}_{\mathbf{Y}}(q) \equiv \sum_{k=1}^{\infty} \mathbf{H}_{\mathbf{Y}}[k] \cdot q^{-k} \cdot \boldsymbol{Z}[t] = \sum_{j=1}^{n \cdot s} \frac{1}{1 - \lambda_j \cdot q^{-1}} \cdot \mathbf{C} \cdot \mathbf{G}_j \cdot \mathbf{C}^T \tag{41}$$

while its weights are calculated by:

$$\begin{aligned} \mathbf{H}_{\mathbf{Y}}[k] &= \mathbf{C} \cdot \mathbf{H}_{\boldsymbol{\Xi}}[k] \cdot \mathbf{C}^T \\ &= \sum_{j=1}^{n \cdot s} \mathbf{C} \cdot \mathbf{G}_j \cdot \mathbf{C}^T \cdot \lambda_j^k \\ &= \sum_{j=1}^{n \cdot s} \boldsymbol{\Omega}_j \cdot \lambda_j^k \end{aligned} \tag{42}$$

Covariance Generating Function. By definition,

$$\boldsymbol{\Gamma}_{\mathbf{Y}}(q) \equiv \sum_{k=-\infty}^{\infty} \boldsymbol{\Gamma}_{\mathbf{Y}}[k] \cdot q^{-k} \tag{43}$$

The covariance matrix of $\mathbf{Y}[t]$ is:

$$\boldsymbol{\Gamma}_{\mathbf{Y}}[h] \equiv E\Big\{\mathbf{Y}[t+h] \cdot \mathbf{Y}^T[t]\Big\} = \sum_{k=0}^{\infty} \mathbf{H}_{\mathbf{Y}}[k+h] \cdot \boldsymbol{\Sigma}_{\mathbf{Z}} \cdot \mathbf{H}_{\mathbf{Y}}^T[k] \tag{44}$$

Recalling that,

$$\boldsymbol{\Sigma}_{\mathbf{N}} = \mathbf{C}^T \cdot \boldsymbol{\Sigma}_{\mathbf{Z}} \cdot \mathbf{C} \tag{45}$$

and

$$\mathbf{H}_{\mathbf{Y}}[k] = \mathbf{C} \cdot \mathbf{H}_{\boldsymbol{\Xi}}[k] \cdot \mathbf{C}^T \tag{46}$$

the following apply:

$$\begin{aligned} \boldsymbol{\Gamma}_{\mathbf{Y}}[h] &= \sum_{k=0}^{\infty} \mathbf{C} \cdot \mathbf{H}_{\boldsymbol{\Xi}}[k+h] \cdot \mathbf{C}^T \cdot \boldsymbol{\Sigma}_{\mathbf{Z}} \cdot \mathbf{C} \cdot \mathbf{H}_{\boldsymbol{\Xi}}^T[k] \cdot \mathbf{C}^T \\ &= \mathbf{C} \cdot \left\{ \sum_{k=0}^{\infty} \mathbf{H}_{\boldsymbol{\Xi}}[k+h] \cdot \boldsymbol{\Sigma}_{\mathbf{N}} \cdot \mathbf{H}_{\boldsymbol{\Xi}}^T[k] \right\} \cdot \mathbf{C}^T \\ &= \mathbf{C} \cdot \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[h] \cdot \mathbf{C}^T \end{aligned} \tag{47}$$

Putting this result in Eq. 43 implies,

$$\boldsymbol{\Gamma}_{\mathbf{Y}}(q) = \mathbf{C} \cdot \boldsymbol{\Gamma}_{\boldsymbol{\Xi}}(q) \cdot \mathbf{C}^T \tag{48}$$

thus the covariance generating function and the covariance matrix of the original VAR(n) model take the form

$$\begin{aligned}\boldsymbol{\Gamma}_{\mathbf{Y}}(q) &= \mathbf{C}\cdot\mathbf{H}_{\boldsymbol{\Xi}}(q)\cdot\boldsymbol{\Sigma}_{\mathbf{N}}\cdot\mathbf{H}^T_{\boldsymbol{\Xi}}(q^{-1})\cdot\mathbf{C}^T \\ &= \mathbf{C}\cdot\mathbf{H}_{\boldsymbol{\Xi}}(q)\cdot\mathbf{C}^T\cdot\boldsymbol{\Sigma}_{\mathbf{Z}}\cdot\mathbf{C}\cdot\mathbf{H}^T_{\boldsymbol{\Xi}}(q^{-1})\cdot\mathbf{C}^T \\ &= \mathbf{H}_{\mathbf{Y}}(q)\cdot\boldsymbol{\Sigma}_{\mathbf{Z}}\cdot\mathbf{H}^T_{\mathbf{Y}}(q^{-1})\end{aligned} \tag{49}$$

and,

$$\boldsymbol{\Gamma}_{\mathbf{Y}}[h] = \sum_{j=1}^{n\cdot s}\mathbf{Q}_j\cdot\lambda_j^h \tag{50}$$

respectively, where:

$$\mathbf{Q}_j = \mathbf{C}\cdot\mathbf{D}_j\cdot\mathbf{C}^T \tag{51}$$

while the dispersion analysis of the state vector (Eqs. 27–30) can be straightforwardly applied to Eq. 50 as well.

Spectral Density. the spectral density matrix of the s–variate time–series $\mathbf{Y}[t]$ is given by,

$$\begin{aligned}\mathbf{S}_{\mathbf{Y}}(f) &= \sum_{k=-\infty}^{\infty}\boldsymbol{\Gamma}_{\mathbf{Y}}[k]\cdot e^{-j\cdot\frac{2\cdot\pi\cdot f}{F_s}\cdot k} \\ &= \sum_{k=-\infty}^{\infty}\mathbf{C}\cdot\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[k]\cdot\mathbf{C}^T\cdot e^{-j\cdot\frac{2\cdot\pi\cdot f}{F_s}\cdot k} \\ &= \mathbf{C}\cdot\mathbf{S}_{\boldsymbol{\Xi}}(f)\cdot\mathbf{C}^T\end{aligned} \tag{52}$$

so that, in light of Eqs. 38 and 51, the following expression is derived:

$$\mathbf{S}_{\mathbf{Y}}(f) = \boldsymbol{\Gamma}_{\mathbf{Y}}[0] + \sum_{m=1}^{n\cdot s}\frac{\lambda_m\cdot e^{-j\cdot\frac{2\cdot\pi\cdot f}{F_s}}}{1-\lambda_m\cdot e^{-j\cdot\frac{2\cdot\pi\cdot f}{F_s}}}\cdot\mathbf{Q}_m + \frac{\lambda_m\cdot e^{j\cdot\frac{2\cdot\pi\cdot f}{F_s}}}{1-\lambda_m\cdot e^{j\cdot\frac{2\cdot\pi\cdot f}{F_s}}}\cdot\mathbf{Q}_m^T \tag{53}$$

Again, since $\mathbf{S}_{\mathbf{Y}}(f)$ is Hermitian, the calculation of the spectral density can be made for the frequency range between the zero and the Nyquist frequencies:

$$\mathbf{S}^*_{\mathbf{Y}}(f) = \begin{cases}2\cdot\mathbf{S}_{\mathbf{Y}}(f) & ,0 < f \leq \frac{F_s}{2} \\ \mathbf{S}_{\mathbf{Y}}(f) & ,f = 0\end{cases} \tag{54}$$

Equations 42, 50 and 54 show how three fundamental characteristics of the s–variate time–series $\mathbf{Y}[t]$ are related to the state matrix, $\mathbf{F}$. In addition, they retain a form that is easy to implement and allows direct quantification of every structural mode. It is important to note that the above analysis is strictly depended to the spectrum of $\mathbf{F}$ and the noise covariance. Indeed, when a VAR(n) structure described by Eq. 1 is available, all the information about the dynamics of the system that produces $\mathbf{Y}[t]$ can be assessed by the eigenvalue problem of $\mathbf{F}$, which is identical to the block companion matrix of the AR polynomial, $\mathbf{A}(q)$.

4 Estimation

The estimation of VAR(n) structures pertains to the identification of the matrix polynomial order and coefficients, as well as the covariance matrix of the vector noise sequence, given observations of a s–variate times–series $\mathbf{Y}[t],\ t = 1, ..., N$, that has been sampled at a period T_s. To this, the state–space realization may again be utilized, noting that, regardless the selected order n of the original VAR structure, the state equation retains the first order VAR form. Post–multiplying Eq. 8 by $\boldsymbol{\Xi}^T[t+h]$, $h > 0$, and taking expectations on both sides leads to the Yule–Walker equations:

$$\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[h] = \mathbf{F}\boldsymbol{\Gamma}_{\boldsymbol{\Xi}}[h-1] \tag{55}$$

The canonical form of the the state matrix allows its decomposition as,

$$\mathbf{F} = \mathbf{T} - \mathbf{C}^T \cdot \mathbf{f} \tag{56}$$

where,

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \otimes \mathbf{I}_s \qquad \mathbf{f} = \begin{bmatrix} \mathbf{A}_n & \mathbf{A}_{n-1} & \dots & \mathbf{A}_1 \end{bmatrix} \tag{57}$$

and $\mathbf{C}$ is given by Eq. 7. Thus, by selecting $h = 0$, the following estimate can be derived,

$$\hat{\mathbf{f}} = \mathbf{C} \cdot \left[\mathbf{T} - \widehat{\boldsymbol{\Gamma}}_{\boldsymbol{\Xi}}[1] \cdot \widehat{\boldsymbol{\Gamma}}_{\boldsymbol{\Xi}}^{-1}[0] \right] \tag{58}$$

where $\widehat{\boldsymbol{\Gamma}}_{\boldsymbol{\Xi}}[0]$ and $\widehat{\boldsymbol{\Gamma}}_{\boldsymbol{\Xi}}[1]$ are the zero and first lag sample covariance matrices of $\boldsymbol{\Xi}[t]$, respectively. Correspondingly, the state noise covariance matrix is estimated as $\widehat{\boldsymbol{\Sigma}}_{\mathbf{N}} = \widehat{\boldsymbol{\Gamma}}_{\boldsymbol{\Xi}}[0] - \widehat{\mathbf{F}} \cdot \widehat{\boldsymbol{\Gamma}}_{\boldsymbol{\Xi}}^T[1]$.

It can be shown that the estimator of Eq. 58 is of minimum phase, sharing similar convergence properties to the least–squares one (the reader is referred to [5] and [16], Chapter 3, for details about convergence). Having the state equation estimated, the transition to the original VAR(n) structure is designated by the matrix $\mathbf{C}$ of the state–space realization's output equation. To this, the transformation methods that were implied in Sec. 3 can be applied. Notice that in fitting VAR models to data, it is interest to keep in mind the procedure of fitting higher–order VAR models and testing the significance of the last coefficient matrix, as a direct extension of the partial autocorrelation function's use. It is easy to observe that this procedure can be also reflected back to the eigenvalue problem of the state matrix: when the last coefficient approaches zero, $\widehat{\mathbf{F}}$ approaches singularity and some of its eigenvalues produce low norm percentage dispersions. Regarding the latter, their significance can be quantified using the following steps:

Step 1**:** under the availability of an estimated VAR(n) model, apply a state–space realization (Eqs. 8–9), construct the state matrix $\widehat{\mathbf{F}}$ and calculate its eigenvalues. Say that

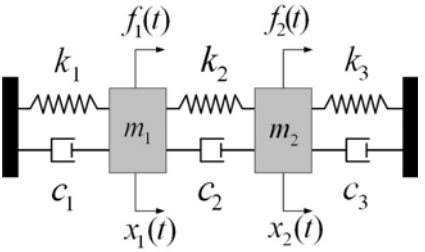


Fig. 1. A structural system with two degrees of freedom: $m_1 = m_2 = 4.5\ kg$, $c_1 = 45\ Ns/m$, $c_2 = 35\ Ns/m$, $c_3 = 15\ Ns/m$, $k_1 = k_3 = 17500\ N/m$, $k_2 = 100\ N/m$

these are split into k_1 real and k_2 complex conjugate pairs ($k_1 + 2 \cdot k_2 = n \cdot s$, s the dimension of the vector time–series).

Step 2**:** calculate the coefficients $\widehat{\mathbf{Q}}_j$ of the covariance matrix $\widehat{\boldsymbol{\Gamma}}_{\mathbf{Y}}[h]$ (Eqs. 51–50, respectively).

Step 3**:** extend the procedure described by Eqs. 27–30 to the $\widehat{\mathbf{Q}}_j$'s and calculate the, real and complex, percentage dispersion matrices $\widehat{\boldsymbol{\Delta}}_k$. Use any matrix norm to quantify the significance of the k^{th} mode (the maximum absolute element of $\widehat{\boldsymbol{\Delta}}_k$ is used in the following).

Table 1. Theoretical / identified natural frequencies (Hz) and damping ratios, and dispersions of the identified VAR(2) models (estimated values appear as mean(standard deviation) of the Monte Carlo experiments)

			Dispersion matrices (%)	
	Theoretical	Identified	$1^{st} Mode$	$2^{nd} Mode$
w_n	9.98	9.99(0.13)		
	9.93	9.93(0.05)	$\begin{bmatrix} 37.50(5.40) & -32.58(8.83) \\ -36.31(8.71) & 9.53(2.50) \end{bmatrix}$	$\begin{bmatrix} 62.50(5.40) & 132.58(8.83) \\ 136.31(8.71) & 90.77(2.50) \end{bmatrix}$
ζ	0.18	0.18(0.02)		
	0.05	0.05(0.00)		

5 Application Study

The method's performance was examined through the structural identification problem of a vibrating system with two degrees of freedom, presented in Fig. 1. The system is characterized by a pair of closely spaced modes, as indicated in Tab. 1 and the vector time–series used for the identification tasks was the vibration displacement of the masses. The statistical consistency of the method was investigated via Monte Carlo analysis that consisted of 20 data records of vibration displacement time–series, with each such record having 1000 samples, obtained with different white excitations and noise–corrupted at 5% noise to signal (N/S) ratio. Regarding the simulation, the continuous system was discretized using the impulse–invariant transformation, at a sampling period $T_s = 0.025\ s$.

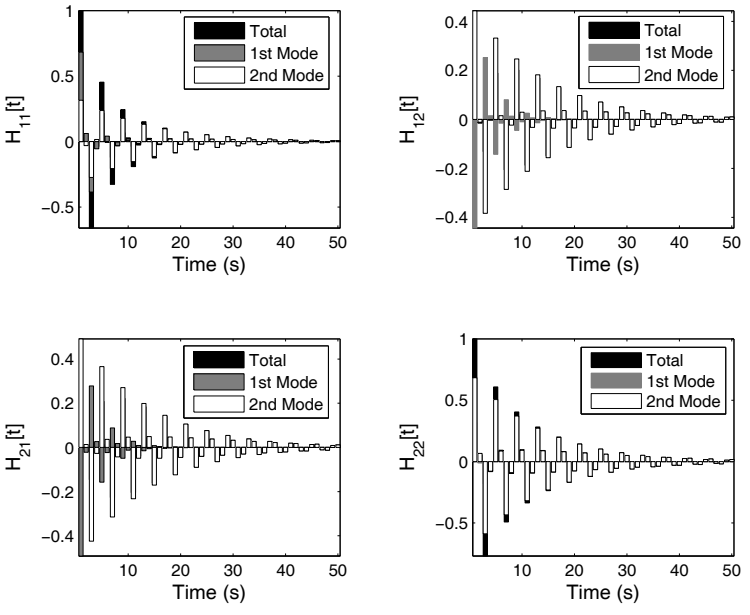


Fig. 2. VAR(2) impulse responses and relative contributions of every mode

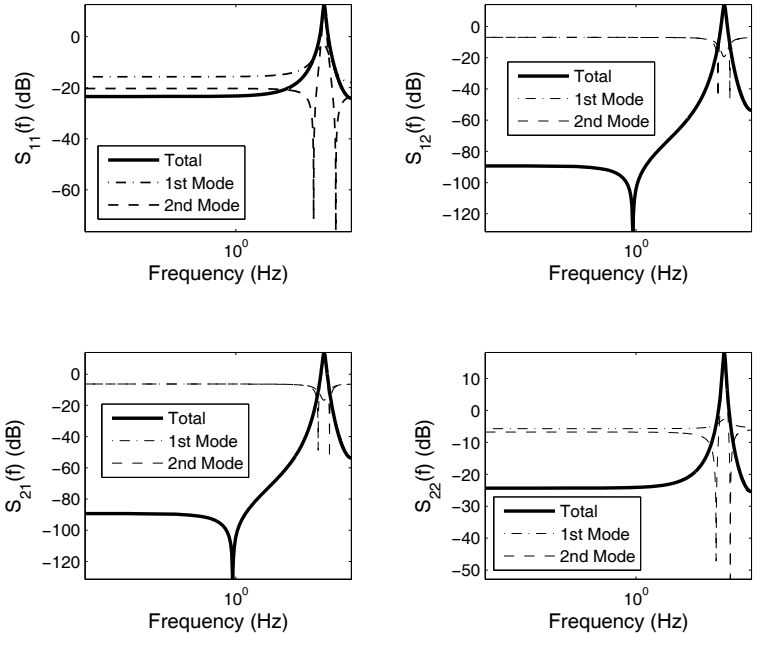


Fig. 3. VAR(2) spectral density matrix and relative contributions of every mode

Following the estimation procedure described in Sec. 4 and utilizing statistical order selection via the BIC criterion [5], VAR(2) models were found adequate to describe system dynamics. Table 1 illustrates the estimates of the natural frequencies and the damping ratios, together with the theoretical ones, from where it is clear that the method performed satisfactory and identified the relative quantities, even in the absence of the input excitations. Table 1 further displays the percentage dispersion matrices for each mode of vibration, showing that the second mode is a heavier contributor in the total energy of the system.

For further validation of the results, Figs. 2–3 display the Green function (impulse response) and the spectral density matrix, respectively, for one VAR(2) model out of the estimated set, together with the contributions from the two modes of interest. Clearly, the second mode dominates in both time and frequency domains, apart from the $H_{11}[t]/S_{11}(f)$ impulse/frequency responses, where the first mode determines the performance.

6 Conclusions

A novel method for the representation of vector time–series, by means of VAR(n) structures, was presented in this paper. Focusing on the estimation of structural information, the method takes advantage of the fact that every VAR(n) structure can be turn into a VAR(1) counterpart and is led to a state–space realization, whose state matrix coincides with the block companion matrix of the VAR polynomial. Consequently, it was shown how important quantities of the original VAR(n) structure, such as the Green function, the covariance generating function and the spectral density matrix, can be qualified and assessed only in terms of the spectrum of the transition matrix (and the noise covariance, where applicable). This fact provides the user with the ability to accurately evaluate the significance of every structural mode in the total vector time–series energy (a technique referred to as dispersion analysis). Of the advantages of the method is its ability to use the state equation for identification purposes, by implementing a Yule–Walker based estimate that results in (unique, for a given data set) minimum phase estimators.

The encouraging results (reduced data acquisition, statistical consistency, accurate structural identification, no overdetermination, unique estimate) suggest the further research into this field. Extension of the method to vector time–series with structural indices governed by multiple eigenvalues, probably by means of Jordan canonical forms, as well as the investigation of VARMA models, ensues straightly. Of main interest is also the application of the method under the availability of input excitation and the expansion of its framework to non–stationary vector time–series, to closed–loop operations, as well as to fault diagnosis schemes.

References

1. Box, G.E.P., Jenkins, G.M., Reinsel, G.C.: Time Series Analysis, Forecasting and Control, 4th edn. Prentice-Hall, Inc., New Jersey (2008)
2. Brockwell, P.J., Davis, R.A.: Introduction to Time Series and Forecasting, 2nd edn. Springer, New York (2002)

3. Clements, M.P., Henry, D.F.: Forecasting Economic Time Series. Cambridge University Press, Cambridge (1998)
4. Dertimanis, V.K.: Fault Modeling and Identification in Mechanical Systems. Ph.D. thesis, School of Mechanical Engineering, Athens, Greece (2006) (in greek)
5. Dertimanis, V.K., Koulocheris, D.V.: On the state-space realization of vector autoregressive structures: an assessment. In: Proceedings of the 6th ICINCO Conference. INSTICC, Milan, Italy (2009)
6. Fassois, S.D.: MIMO LMS-ARMAX identification of vibrating structures-Part I: The method. Mechanical Systems and Signal Processing 15(4), 737–758 (2001)
7. Hannan, E.J.: The identification and parameterization of ARMAX and state space forms. Econometrica 44(4), 713–723 (1976)
8. He, J., Fu, Z.F.: Modal Analysis. Butterworth-Heinemann, Oxford (2001)
9. Hendricks, E., Jannerup, O., Sørensen, P.H.: Linear Systems Control: Deterministic and Stochastic Methods. Springer, Berlin (2008)
10. Huang, C.S.: Structural identification from ambient vibration using the multivariate AR model. Journal of Sound and Vibration 241(3), 337–359 (2001)
11. Koulocheris, D.V., Dertimanis, V.K., Spentzas, C.N.: Parametric identification of vehicle structural characteristics. Forschung im Ingenieurwesen 72(1), 39–51 (2008)
12. Landau, I.D., Zito, G.: Digital Control Systems: Design, Identification and Implementation. Springer, London (2006)
13. Lardies, J.: Relationship between state-space and ARMAV approaches to modal parameter identification. Mechanical Systems and Signal Processing 22(3), 611–616 (2008)
14. Lee, J.E., Fassois, S.D.: On the problem of stochastic experimental modal analysis based on multiple-excitation multiple-response data, Part I: Dispersion analysis of continuous-time systems. Journal of Sound and Vibration 161(1), 57–87 (1993)
15. Ljung, L.: System Identification: Theory for the User, 2nd edn. Prentice-Hall, Inc, New Jersey (1999)
16. Lütkepohl, H.: New Introduction to Multiple Time Series Analysis. Springer, Berlin (2005)
17. Meyer, C.D.: Matrix Analysis and Applied Linear Algebra. Society for Industrial and Applied Mathematics, Philadelphia (2000)
18. Papakos, V., Fassois, S.D.: Multichannel identification of aircraft skeleton structures under unobservable excitations: A vector AR/ARMA framework. Mechanical Systems and Signal Processing 17(6), 1271–1290 (2003)

Strain Analysis of the Sciara del Fuoco (Stromboli Volcano)

G. Nunnari[1], A. Spata[1], G. Puglisi[2], A. Bonforte[2], and F. Guglielmino[2]

[1] Università degli Studi di Catania, viale A. Doria 6, 95125 Catania, Italy
[2] Istituto Nazionale di Geofisica e Vulcanologia - Sezione di Catania
Piazza Roma 2, 95125 Catania, Italy
gnunnari@diees.unict.it,alessandro_spata@yahoo.it
{puglisi-g,bonforte,guglielmino}@ct.ingv.it

Abstract. In this paper we describe the pre-processing techniques and the theory of classical mechanics we have adopted in order to computed both 3D motion maps and 3D strain tensor over the Sciara del Fuoco (SdF), the Stromboli volcano (Italy) steep flank subjects to dangerous landslide events. Movements affecting the SdF are measured by the monitoring system known as THEODOROS (THEOdolite and Distancemeter Robot Observatory of Stromboli).

1 Introduction

Stromboli is an active volcano, about 2500 m high above the sea floor. Roughly only the last kilometre of this volcano emerges from the sea, forming an island whose diameter ranges from 2.4 to 5 km. It belongs to the Aeolian Islands and represents the most active volcano of this archipelago. Its conic shape is evidently characterized by a big depression that marks the northwestern flank of the edifice: the Sciara del Fuoco (SDF). On December 28th, 2002, lava flows outpoured from the northern wall of the NE crater and descended into the Sciara del Fuoco, a deep depression marking the NW flank of the volcano edifice. On December 30th, 2002, two landslides occurred on the northern part of the Sciara del Fuoco; they moved a mass in the order of tens of millions of cubic meters both above and below sea level. The landslide produced a tsunami causing significant damage to the eastern coast of the island, reaching the other Aeolian Islands and the Sicilian and southern Italian coasts. This event led to the upgrading of the ground deformation monitoring system, already existing on the island; the new requirement was the real-time detection of the deformations related to potential slope failures of the SdF. To this purpose the chosen instrument was the Leica TCA 2003 Total Station (TS) equipped with GeoMos software [4] that allows remote sensor control. The acronym of this system is THEODOROS (THEOdolite and Distance-meter Robot Observatory of Stromboli) [7]. The rest of this paper is organized in the following way. In Sec. II a brief description of the current THEODOROS configuration is given, the interested reader can found more detailed information in [4,7]. Sec. III reports the approach adopted to pre-processing data; Sec. IV shows the methodology used to compute the strain field; Sec. V reports the case study; finally Sec. VI draws the conclusions of this study.

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 317–323.
springerlink.com

2 Brief Introduction to the THEODOROS System

The THEODOROS system consists of a remote-controlled Total Station that can be programmed to measure slope distances and angles between the sensor and benchmarks appropriately installed in the SDF area at a specific sampling rate. To ensure a continuous stream of data from the instrument, it requires a constant power supply and a continuous link with the PC controlling the Total Station's activities, installed on the S. Vincenzo Observatory, where the National Department of Civil Protection (DPC) control room is located. The Stromboli volcano eruption of the 27 February 2007 destroyed the THEODOROS benchmarks inside the SDF. A new configuration was designed and new benchmarks were installed on the new fan produced by the lava flow entering the sea. This new topology consists of six reflectors installed outside the SdF, around the Total Station, for the reference system and atmospheric corrections (SENT, BORD, SEMF, SPLB2, CIST and ELIS), nine reflectors for monitoring movements of the lava fan inside the SdF (SDF18, SDF19, SDF20, SDF21, SDF22, SDF23, SDF24, SDF25 and SDF26), two reflectors to monitor the northern border of the SdF (400 and BASTI) and two further reflectors on stable sites to check the stability of the measurements both on short and very long distance measurements (CURV and CRV). Currently the reflectors SDF20 and SDF21 are not working. The sample time indicated as t_c hereafter is set to be $t_c = 10$ minutes. Each measurement for each target or reference point provides the instantaneous values of three relevant pieces of information: the slope distance (sd), the horizontal (hz) angle and the vertical angle (ve). Starting from this information, the GeoMos system is able to transform the TS measurement vectors (whose components are sd, hz, ve) into an equivalent vector whose components are expressed in terms of North (N), South (S) and Up (U) with respect to the assumed reference system. In this computation, GeoMos is able to take into account the constraints imposed by the assumption of the reference system. Despite the availability of real-time information, this is not enough to automatically evaluate the state of ground deformation. Indeed the acquired measures are affected by offsets, spikes and noise sources that strongly compromised their interpretation. These drawbacks must be necessary overcome before that suitable quantities related to the ground deformation dynamic can be efficiently computed. In particular in this paper we focus our attention on the problems of offsets and spikes removal, smoothing noisy data and strain tensor evaluation.

3 Pre-processing Data

The algorithm we propose to remove both spikes and offsets consists of two steps. First the spikes are removed, then attention is focused on offsets. Since the single displacement components (North, East, Up) of each benchmark in the period June 2006 - December 2008 are characterized by a normal distribution, the problem to remove the spikes affecting observations, i.e. the sharp variations of the time series which are generally due to either periodical maintenance or instrument malfunctions, is well solved adopting the standard deviation of observations as reference. Indeed let $T_{SdF_x}(t)$ be a generic component of the benchmark SdF_x at time t, let $\Delta T_{SdF_x}(t)$ be the difference between two subsequent measures and denoting as σ its standard deviation, the

experience gained through the daily monitoring of the SdF suggest us to consider as spikes the $\Delta T_{SdF_x}(t)$ values falling outside the range covered by one σ.

The offsets affecting observations are essentially due to the maintenances of the THEODORO system. Here it is necessary to distinguish two types of maintenances: periodical maintenance usually carried out every six months, and extra maintenance due to unexpected crash of the system. The offsets related to the periodical maintenance are simply adjusted taking into account the marked sharp variation (jump) visible when the system begins to work. This approach is also suitable for offset due to the crash of the system if the normal functioning of the system is promptly restored. Instead, if the extra maintenance is performed after a sufficiently long time the system crashed, then the offsets removal is not trivial. Indeed, in this case, in order to perform a reliable offsets correction the estimation of the trend of each ground deformation component during the period in which the system was crashed is needed. In order to adjust these kinds of offsets we use the linear trend as shown in figure 1.

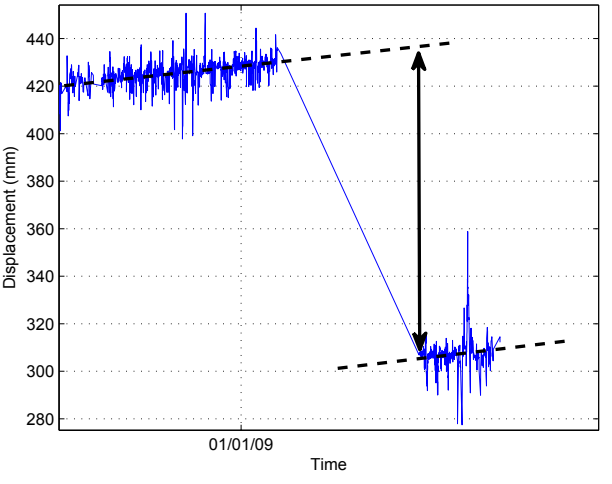


Fig. 1. Offsets correction approach based on linear trend

Although both spikes and offsets removal makes ground deformations more readable, further processing is needed to reduce noise source affecting data, in particular the thermoelastic effects on ground deformation due to the temperature. To this purpose we have smoothed noisy data with spline functions [1]. A spline function $s(t)$ is a function defined piecewise by polynomials. This function takes values from an interval $[a, b]$ and maps them to R, the set of real numbers. The interval $[a, b]$ is divided into k disjoint subintervals $[t_i, t_{i+1}]$ with $0 \leq i \leq k-1$ such that $[a, b] = [t_0, t_1]U...U[t_{k-2}, t_{k-1}]$. The given k points t_i are called knots. The vector $t = (t_0, .., t_{k-1})$ is called a knot vector for the spline. If the knots are equidistantly distributed in the interval $[a, b]$ the spline is uniform, otherwise it is non-uniform. On each of this subintervals a nth polynomial is defined and joined with others polynomials at the knot points in such a way that all derivatives up to the $(n-1)th$ degree are continuous. Within these constraints, the function $s(t)$ is selected which minimizes:

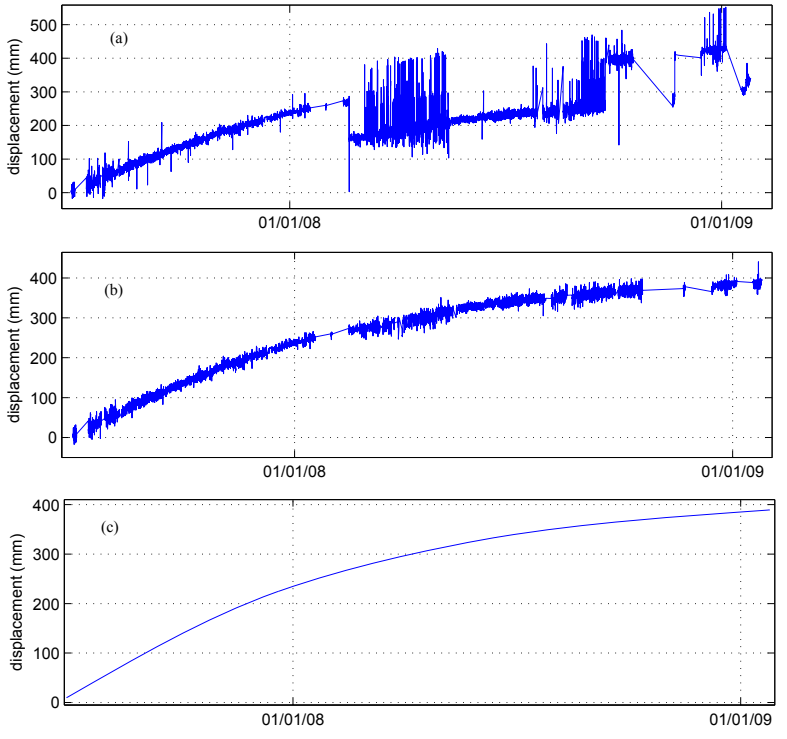


Fig. 2. (a) Original SDF26 North component; (b) Spikes and offset removed; (c) Smoothing noise

$$\sum (s(t_i) - x_i)^2 + p \int (s^{(\frac{n+1}{2})}(t))^2 dt \tag{1}$$

where (t_i, x_i) are the raw data samples and $s(k)$ denotes the kth derivative of $s(t)$. The weight factor p is the smoothing parameter whose value must be opportunely chosen to obtain a good compromise between good fit and the smoothness. In figure 2 are shown, respectively, the raw data of the benchmark SdF26 (North component), the data after removing spikes and offsets and finally the spline-smoothing.

4 Strain Interpolation

In order to compute both 3D displacements map and strain tensor in the area of SdF covered by the THEODOROS system we use the modified least-square approach introduced by [8] and also used by [5] and [9]. Given a point P having position $x_0 = (x_{10}, x_{20}, x_{30})$ surrounded by N experimental points (EPs) whose positions and displacements are respectively $x(n) = (x_{1(n)}, x_{2(n)}, x_{3(n)})$ and $u(n) = (u_{1(n)}, u_{2(n)}, u_{3(n)})$ where $n = 1..N$, the problem of estimating both the displacements gradient tensor H and the displacement components $U_i (i = 1..3)$ of the point P, according to the infinitesimal strain theory, can be modelled in terms of the following strain interpolation equations:

$$u_{i(n)}(x) = H_{ij} \Delta x_{j(n)} + U_i \quad (i, j = 1..3) \tag{2}$$

where $\Delta x_{j(n)} = x_{j(n)} - x_{j0}$ are the relative positions of the nth EP experimental points and the arbitrary point P and $H_{ij} = \frac{\partial u_i}{\partial x_j}$ are the elements of the displacement gradient tensor. It can be decomposed in a symmetric and an anti-symmetric part as $H = E + \Omega$, where E is the strain tensor defined as:

$$E = \frac{1}{2}(H_{ij} + H_{ji})e_i \oplus e_j \tag{3}$$

and Ω is the rigid body rotation tensor defined as:

$$\Omega = \frac{1}{2}(H_{ij} - H_{ji})e_i \oplus e_j \tag{4}$$

Here e_i is the base vector of the Cartesian reference system and $\oplus$ is the tensor product. In a compact form the undetermined system of equations (2) can be written as $Al = u$ where A is the design matrix simply derivable from equation (2), $l = [U_1\ U_2\ U_3\ \epsilon_{11}\ \epsilon_{12}\ \epsilon_{13}\ \epsilon_{22}\ \epsilon_{23}\ \epsilon_{33}\ \omega_1\ \omega_2\ \omega_3]$ is the vector of unknown parameters and $u = [u(1)\ u(2)\ \ u(n)]^T$ is the observation vector. Assuming a uniform strain field and re-writing the previous linear equation (4) as $Al = u + e$, where e is the residual vector which model the stochastic nature of the estimation problem, a suitable method to solve the system is the Weighted Least Squares (WLS) which gives the expression (5) as a suitable formula to estimate the unknown vector l

$$\hat{l} = (A^T W A)^{-1} A^T W u \tag{5}$$

In (5) W is the data covariance matrix. Usually W is assumed to be diagonal (uncorrelated data), i.e. of the form

$$W = diag(\sigma^{-2}_{1(1)}, \sigma^{-2}_{2(1)}, \sigma^{-2}_{3(1)}, ..., \sigma^{-2}_{1(n)}, \sigma^{-2}_{2(n)}, \sigma^{-2}_{3(n)}) \tag{6}$$

where the quantities $\sigma_{j(n)}$'s are the standard deviations of the measurements. According to the modified least squares (MLS) approach proposed by [8], based on the adjustment of the covariance matrix W, we use the matrix W' which is a weighted version of the matrix W of experimental data. Following the suggestion given by [8] and [9] the weight function considered here is:

$$W' = We^{-\frac{d_{(n)}}{d_0}} \tag{7}$$

where $d_{(n)}$ is the distance between the nth EP and the arbitrary point P, and d_0 is a distance-decaying constant defining the "level of locality" of the estimation.

This method, likewise most previous methods [2] and [6] is used to interpolate the strain among benchmarks of geodetic networks where ground deformations are measured by comparing geodetic surveys.

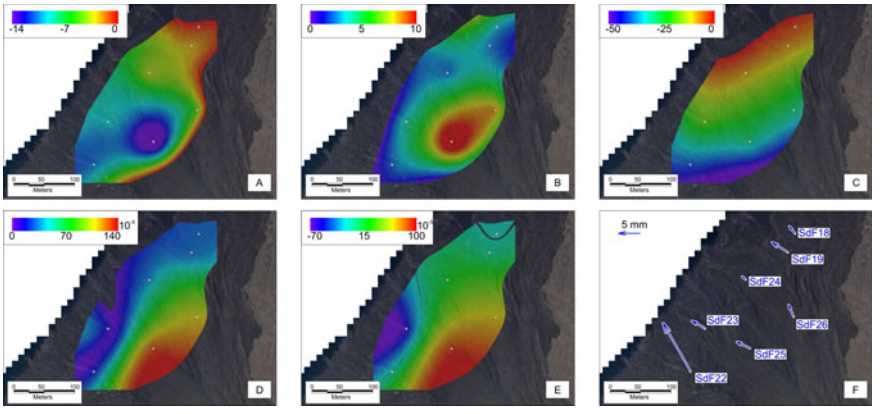


Fig. 3. (a) In the frames (a), (b) and (c) are reported the calculated East, North and Up components of displacements respectively. Frames (d) and (e) report the maximum shear strain and the volume variation. Finally, in the frame (f) the displacement vectors of the benchmarks are shown.

5 An Application to the Sciara del Fuoco

The benchmarks placed on the lava fan show a general NW-ward motion following the maximum slope of the SdF, with an increasing magnitude from NE to SW (Fig. 3). This kind of deformation is in good agreement with a seawards motion of the new lava fan, driven by a mainly gravitational dynamics. However, the ground motion is not uniform above the investigated area, showing significant differences in the displacements measured on different benchmarks. In order to analyze the ground deformation pattern recorded from December 14, 2008 to January 3, 2009 above the deforming lava body, we performed a strain interpolation. Unfortunatly the corresponding linear system is undetermined since it implies more unknows ($n = 9$) that equations ($m = 7$). Therefore the solution is never unique. To this reason we have calculated a basic solution with almost m non zero components by using a MATLAB algorithm based on the QR factorization with column pivoting. Results are reported in Fig. 3, where the decrease of the horizontal motion (Fig. 3a and b) is evident from benchmark SdF25, that is located on the upper and central area of the fan, towards the coastline and towards North, reaching the minimum values at SdF18 benchmark, located close to the SdF northern rim. The vertical motion (Fig. 3c) shows a more uniform gradient, from a maximum down-lift of about 50 mm at the S-Westernmost benchmark (SdF22) to near 0 at SdF18. A deeper analysis of the overall deformation of the lava fan is allowed by the interpolation of the strain tensor. In Fig. 3d, the distribution of the maximum 3D shear strain is reported, confirming the strongest deformation on the upper area of the lava fan; this is mainly due to the stronger magnitude of horizontal displacements of the southernmost SdFS22, SdFS25 and SdFS26 benchmarks with respect to the northern half of the fan, but also to the relative vertical motion of the two halves of the body. On the upper area, also the volumetric dilatation evidences a maximum expansion (Fig. 3e), mainly imputable to the divergent directions of the displacements affecting SdF25 and SdFS26 benchmarks. In addition, a contracting area is detected on the southern coastline of the fan, due to the

smaller displacements of the SdF23 and SdF24 benchmarks with respect to the upper ones, while all the northern half of the lava body shows no significant volumetric strain variation confirming the higher stability of this portion of the fan that is buttressed by the stable northern wall of the SdF.

6 Conclusions

In this paper we have first shown the pre-processing techniques adopted to reduce noise sources affecting ground deformation measures acquired at the Sciara del Fuoco by the automatic monitoring system referred to as THEODOROS. In particular, due to the gaussian distribution of acquisitions, the problem of spikes removal was simply solved taking into account their standard deviations. The offsets due to the crash of the system have been adjusted based on the evaluation of the linear trend of observations. Finally spline functions have been used to reduce the thermoelastic effects. After these pre-processing steps we have shown the based on infinitesimal strain theory method used to compute both displacements maps and strain field over the area covered by the THEODOROS system. Finally a case study related to the ground motion observed in the period December 2008 - January 2009 was carried out in order to test the proposed methodology.

References

1. Biloti, R., Santos, L.T., Martin, T.: Automatic smooting by optimal splines. Rev. Bras. Geof. 21(2), 173–177 (2008)
2. Frank, F.C.: Deduction of Earth Strains from Survey Data. Bull. Seismol. Soc. Am. 56, 34–42 (1966)
3. Leica. Software Geomos User Manual. LEICA and GEODETICS Inc. (2002)
4. Nunnari, G., Puglisi, G., Spata, A.: A warning system for Stromboli volcano based on statistical analysis. Pageoph. 165(8), 1619–1641 (2008)
5. Pesci, A., Teza, G.: Strain rate analysis over the central apennines from GPS velocities: the development of a new free sofftware. Bollettino di Geodesia e Scienze Affini. 56, 69–88 (2007)
6. Prescott, W.H.: An extension of Franks method for obtaining crustal shear strains from survey data. Bull. Seismol. Soc. Am. 66(6), 1847–1853 (1976)
7. Puglisi, G., Bonaccorso, A., Mattia, M., Alosis, M., Bonforte, A., Campisi, O., Cantarero, M., Falzone, G., Puglisi, B., Rossi, M.: New integrated geodetic monitoring system at Stromboli volcano (Italy). Engineering Geology 79(1-2), 13–31 (2004)
8. Shen, Z.D., Ge, B.X.: Crustal deformation across and beyond the Los Angels basin from geodetic measurements. Journal of Geophysical Research 101(27), 957–980 (1996)
9. Teza, G., Pesci, A., Galgaro, A.: Gridstrain and Gridstrain3: Software packages for strain field computation in 2D and 3D environments. Computers and Geoscience 34(9), 1142–1153 (2008)

Improving Performance of Generalised Conditioning Technique Anti-windup Compensator

Dariusz Horla

Poznan University of Technology
Institute of Control and Information Engineering
ul. Piotrowo 3a, 60-965 Poznan, Poland
dariusz.horla@put.poznan.pl

Abstract. The paper presents a novel modification to the generalised conditioning technique anti-windup compensator presented as a triplet of its possible modifications. The results are presented from application of pole-placement control, with an analysis of closed-loop system behaviour for a class of second-order minimumphase stable plants of oscillatory and aperiodic characteristics with different dead-times.

1 Introduction

Constraints are ubiquitous in real-world environment. As the result of their presence or the presence of some nonlinearities in the control loops, arises the difference in between computed and applied (i.e. constrained) control signal. In such a case, the performance of the closed-loop system degrades in comparison with the performance of the linear system, when constraints are not active. Such a degradation is defined as windup phenomenon [4,5,6].

This can be also viewed from the point of discrepancy in between internal controller states and its output. When there is no correspondence in between controller's output and its internal controller states, the controller does not have any information what the current value of the constrained control signal is, and windup phenomenon arises.

The windup phenomenon has been, at first, defined for controllers comprising integral terms, as the so-called integrator windup [4]. For such controllers, control constraints may cause excessive integration of the error signal, giving rise to longer settling of the output signal and overshoots. There are two strands in compensating windup phenomenon (in AWC, anti-windup compensation) – taking constraints into account during the design procedure of the controller or assuming the system is linear, designing the controller for the linear case, and, subsequently, imposing constraints and applying AWCs [1,2,3].

The simplest anti-windup compensators have been based on the idea of integrator clamping, i.e. they referred to the controllers comprising integral terms only [7]. The proposed AWCs avoided integration of the error signal whenever some conditions were met, e.g., the control signal saturated, or error exceeded some predefined threshold, etc.

Such an approach was simple enough to be easily implemented, but as it has been already said, applicable to some controllers only.

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 325–336.
springerlink.com

The advanced anti-windup compensators have been designed for the case of general controller, which input-output equation is written in the RST form. Among the proposed AWCs one can find in the literature deadbeat, generalised, conditioning technique, modified conditioning technique and generalised conditioning technique anti-windup compensators [2,3]. The three latter AWCs are based on the idea of back-calculation, i.e. modification of the signal that the output signal of the plant is to track, with respect to current saturation level.

The paper focuses on the generalised conditioning technique AWC (GCT-AWC), being a compromise solution in between the simplicity of the advanced AWC and compensation capabilities of the conditioning algorithm, what will be explained later.

The main idea of the paper is to present a modification of the GCT-AWC that can arise from the idea of integrator clamping methods, and to show that it can result in better control performance than performance of the system with original GCT-AWC. The presented results refer to the research carried for a set of stable minimumphase second-order discrete-time plants and different constraint levels.

There are no remarks in the literature how to improve the performance of the GCT-AWC, apart from [2]. The proposed method limits the number of modifications, with the same excess. By introducing the proposed modifications one can improve the performance of the most appealing AWC technique.

2 Plant Model

Let the discrete-time CARMA model be given

$$A(q^{-1})y_t = B(q^{-1})u_{t-d}\,, \tag{1}$$

where y_t is the plant output, u_t is the constrained control input, $d \geq 1$ is a dead-time and:

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2}\,, \tag{2}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} \tag{3}$$

are relatively prime. The control signal $u_t = \mathrm{sat}(v_t; \alpha)$ is the computed control signal after saturation by symmetrical cut-off function at level $\pm\alpha$.

3 Controller

The plant is controlled by the pole-placement controller that ensures tracking of a given reference signal r_t by the plant output y_t with given dynamics,

$$\begin{aligned} v_t = k_R r_t - k_P y_t + k_I \frac{q^{-1}}{1-q^{-1}}\,(r_t - y_t) - \\ -k_D \frac{1-q^{-1}}{1-\gamma q^{-1}}\,y_t\,, \end{aligned} \tag{4}$$

where $k_R = rk_P, r > 0$. The above controller equation can be obtained by discretisation of a continuous-time PID controller [4], and it can be rewritten into the RST structure [3]

$$R(q^{-1})v_t = -S(q^{-1})y_t + T(q^{-1})r_t\,. \tag{5}$$

Coefficients of polynomials $R(q^{-1})$, $S(q^{-1})$, $T(q^{-1})$ can be determined by solving the following Diophantine equation

$$\begin{aligned} A(q^{-1})R(q^{-1}) + q^{-d}B(q^{-1})S(q^{-1}) = \\ = A_M(q^{-1})A_o(q^{-1})\,, \end{aligned} \tag{6}$$

where polynomials $A_o(q^{-1})$ and $A_M(q^{-1})$ are stable, and given polynomial $A_M(q^{-1})$ is of second order in the chapter.

Controller polynomials $R(q^{-1})$, $S(q^{-1})$, $T(q^{-1})$ are of order $d + nB$, nA, nA_o, respectively, and have forms as follows:

$$\begin{aligned} R(q^{-1}) &= (1 - q^{-1})R'(q^{-1})\,, \\ S(q^{-1}) &= s_0 + s_1q^{-1} + s_2q^{-2}\,, \\ T(q^{-1}) &= k_RA_o(q^{-1})\,. \end{aligned} \tag{7}$$

From the controller equation (5), given structure $R(q^{-1})$, $S(q^{-1})$, $T(q^{-1})$ (7) and (4) it follows that:

$$\begin{aligned} s_0 &= k_P + k_D\,, \\ s_1 &= k_I - 2k_D - k_P(1+\gamma)\,, \\ s_2 &= k_D - \gamma(k_I - k_P)\,, \end{aligned} \tag{8}$$

$$\begin{aligned} A_o(q^{-1}) &= \left(1-\gamma q^{-1}\right)\left(1-q^{-1}\left(1-\frac{k_I}{k_R}\right)\right) = \\ &= 1 + a_{o1}q^{-1} + a_{o2}q^{-2}\,, \end{aligned} \tag{9}$$

where $\gamma = -\frac{b_1}{b_0}$, $k_R = rk_P$, $a_{o1} = \frac{k_I}{k_R} - (1+\gamma)$, $a_{o2} = \gamma\left(1 - \frac{k_I}{k_R}\right)$. As the polynomial $A_o(q^{-1})$ has to be stable, $0 < \frac{k_I}{k_R} < 2$ must hold what can be ensured by a proper choice of r.

The controller algorithm is assumed to be altered by anti-windup compensator presented in the next Section, in order to assure better control performance of the closed-loop system subject to constraints. It is to be borne in mind that the compensation is based on back-calculation, i.e., it does not require the controller to have integral terms in general.

4 Generalised Conditioning Technique AWC

In GCT, the filtered set-point signal is conditioned, instead of the set-point r_t, and given as

$$r_{f,t} = \frac{Q(q^{-1})T_1(q^{-1})}{L(q^{-1})}\,r_t\,, \tag{10}$$

with

$$\begin{aligned} T(q^{-1}) &= T_2(q^{-1})T_1(q^{-1})\,, \\ Q(q^{-1}) &= q_0 + q_1 q^{-1} + \cdots + q_{nQ} q^{-nQ}\,, \\ L(q^{-1}) &= 1 + l_1 q^{-1} + \cdots + l_{nL} q^{-nL} \end{aligned} \tag{11}$$

and $T_2(0) = t_{2,0}$.

Similarly to the conditioning method (see [2]), the modified filtered reference signal is given by

$$r^r_{f,t} = r_{f,t} + \frac{q_0(u_t - v_t)}{t_{2,0}}\,, \tag{12}$$

and the control signal is defined as

$$\begin{aligned} v_t = (1 - Q'(q^{-1})R(q^{-1}))u_t + \frac{t_{2,0}}{q_0}\, r_{f,t} + \\ + \frac{1}{q_0}((T_2(q^{-1})L(q^{-1}) - t_{2,0})r^r_{f,t} - \\ - Q'(q^{-1})S(q^{-1})y_t\,, \end{aligned} \tag{13}$$

where $Q'(q^{-1}) = \frac{Q(q^{-1})}{q_0}$.

The GCT method enables additional tuning of the performance by reference signal filter design. Because its parameters should correspond to model parameters, saturation level and set-point values, a special choice of parameters of the filter (10) for minimumphase second-order model is proposed [2]. Let ϱ_1 and ϱ_2 denote poles of stable $A(z^{-1})$, then

$$\varrho = \max(|\varrho_1|, |\varrho_2|)\,, \tag{14}$$

$$Q(q^{-1}) = 1 + \left((1 - \varrho)^{\xi} - 1\right) q^{-1}\,, \tag{15}$$

$$L(q^{-1}) = 1 - (1 - \varrho)^{\xi} q^{-1}\,, \tag{16}$$

where $0 < \xi \leq 1$ is the damping factor obtained from classical root locus theory for the second-order systems. The suggested filter (14–16) takes into consideration model parameters and set-point values only, forcing the initial values of the filtered reference signal for slow models and reducing the amplitude and rate of transients for oscillatory ones., see Figure 1.

The inherent property of the conditioning technique is the so-called short sightedness phenomenon, resulting in consecutive resaturations of the control signal because of excessive modification of the reference signal. In order to improve the performance of the compensation three modifications will be considered as in the next Section.

5 Modified Generalised Conditioning AWCs

In order to combine classical conditional integration methods that work for controllers comprising integrators with back-calculation AWC presented in the previous Section,

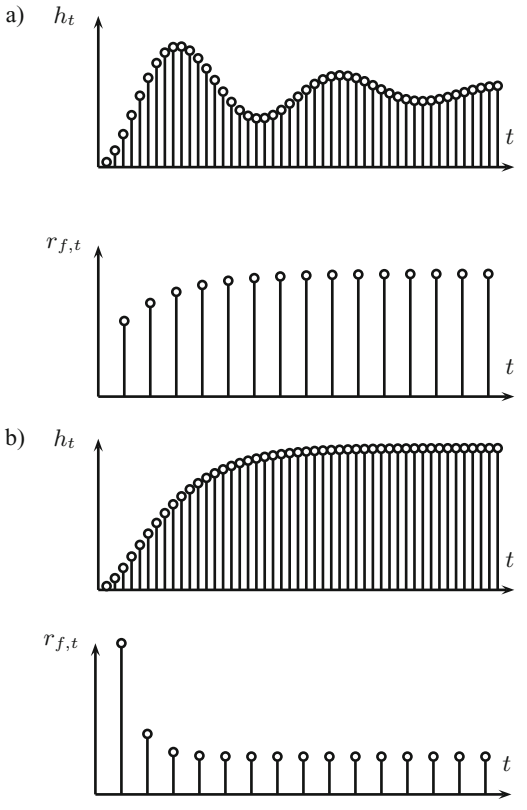


Fig. 1. Plant step response h_t and reference filter step response $r_{f,t}$ for a) oscillatory and b) aperiodic plant

the following three back-calculation modifications have been proposed – the modification of the filtered reference input is applied when:

M1 $|e_t| > e_1$,
M2 $u_{t-1} \neq v_{t-1}$,
M3 $u_{t-1} \neq v_{t-1}$ and $e_t u_{t-1} > 0$,

where e_1 is a threshold value for reference modification clamping.

By applying the modifications to the GCT-AWC one assures that modification of the filtered reference signal is performed only when necessary.

6 Simulated Plants

The simulations have been performed for a set of stable, second-order, minimumphase plants with $B(q^{-1}) = 1 + 0.5q^{-1}$ and:

– P1 type

$$A(q^{-1}) = (1 - q^{-1}(\sigma + \omega i))(1 - q^{-1}(\sigma - \omega i)) ,$$

where:

$$-1 < \sigma < 1\,,$$
$$-1 < \omega < 1\,,$$
$$|\sigma \pm \omega i| < 1\,,$$

what corresponds to oscillatory behaviour of the plant,

– P2 type

$$A(q^{-1}) = (1 - q^{-1}z_1)(1 - q^{-1}z_2)\,,$$

where:

$$0 < z_1 < 1\,,$$
$$0 < z_2 < 1\,,$$

what corresponds to aperiodic behaviour of the plant.

The simulations have been run for square wave reference signal of period 40 samples and symmetrical amplitude ± 3 with $\frac{k_I}{k_R} = 0.5$, $A_M(q^{-1}) = 1 - 0.5q^{-1} + 0.06q^{-2}$ and $e_1 = 3$.

In order to evaluate the quality of regulation process, the performance index is introduced

$$J = \frac{1}{N} \sum_{t=0}^{N} (r_t - y_t)^2\,, \tag{17}$$

where $N = 150$ denotes the simulation horizon.

The simulations have been performed for the same constraint hardness for each of the plants, denoted by relative constraint level α_r (i.e., the multiplicity of the minimum constraint level $\alpha_{\min} = 3\frac{|A(1)|}{|B(1)|}$ allowing asymptotic tracking). The absolute value of the constraint is $\alpha = \alpha_r \alpha_{\min}$ and changes as the plants change.

7 Performance Surfaces

The results of the simulation tests are shown as performance surfaces. Each of the axes has been divided into 101 values, thus all simulation results refer to a grid of 101×101 different plants. The idea of such surfaces is as follows – let J_0 denote the value of the performance index of the control system with some plant and given α_r and no AWC. Let J_1 denote the value of the performance index of the same control system with the same plant but with classical GCT-AWC. Let J_2 denote the value of the performance index of, again, the same control system with the same plant but with modified GCT-AWC (M1, M2 or M3).

For each of the plants and constraints level the following face is plotted:

- (magenta) $J_0 = J_1 = J_2$,
- (red) modification is of the worst performance, $J_0 < J_1 < J_2$ or $J_1 < J_0 < J_2$, the intensity of the red level is proportional to $J_2 - J_0$ or $J_2 - J_1$,
- (white) modification improves the performance of the GCT-AWC, $J_0 < J_2 < J_1$,

- (black) it is not worth to modify GCT, $J_1 < J_2 < J_0$, the intensity of the black level is proportional to $J_2 - J_1$,
- (blue) modification improves the performance where GCT fails to, $J_2 < J_0 < J_1$, the intensity of the blue level is proportional to $J_0 - J_2$,
- (green) modification is of the best performance, the intensity of the green level is proportional to $J_0 - J_2$ or $J_1 - J_2$.

8 Should One Modify GCT?

The performance surfaces have been obtained for P1 and P2 type plants with different dead-times and presented in Figs 1–4, where consecutive rows for different dead-times refer to M1, M2 and M3.

In all the cases of P1 and $d = 1$ it is visible that all modifications can improve the performance of the GCT for slow plants, i.e., with small natural frequency, whereas in the case of M1 and M3 there is an improvement visible for such plants near stability border. In the case of M1 and M3, one can see region of the best improvement (green). By comparing the given surfaces one can say that M3 is of the best AWC performance, because of the green regions and brighter red regions than in other cases, what refers to less performance degradation.

It is not advisable to modify the GCT algorithm when the region is red, it is advisable to improve where it is white and definitely advisable when green.

In the case of $d = 3$ one can see that red regions have almost disappeared and the improvement is best in the case of M3.

For P2 type plants a performance improvement can be observed for slow plants (green) with M1 and M3. Because of the size of white and green regions one can say that the best performance is assured by M1, mainly because of the $\alpha_r > 1$, that is visibility of green regions for greater α_rs. The vast areas of red color suggest that it is inadvisable to modify the original GCT when plant is moderately slow (expressed by absolute values of its poles).

In the case of $d = 3$ because of the area of white region and brightness of the red region, it can be said that M1 is the best choice, then M2 and M3.

9 Summary

It has been shown that it can be advantageous to modify the algorithm of well-known GCT-AWC in order to obtain high control performance. Such a modification can be implemented with the use of lookup table, where the information is stored what GCT algorithm should be used when plant parameters vary in time, e.g. due to aging or set-point change. A similar approach has been presented for continuous system, PID controllers and integrator clamping [7].

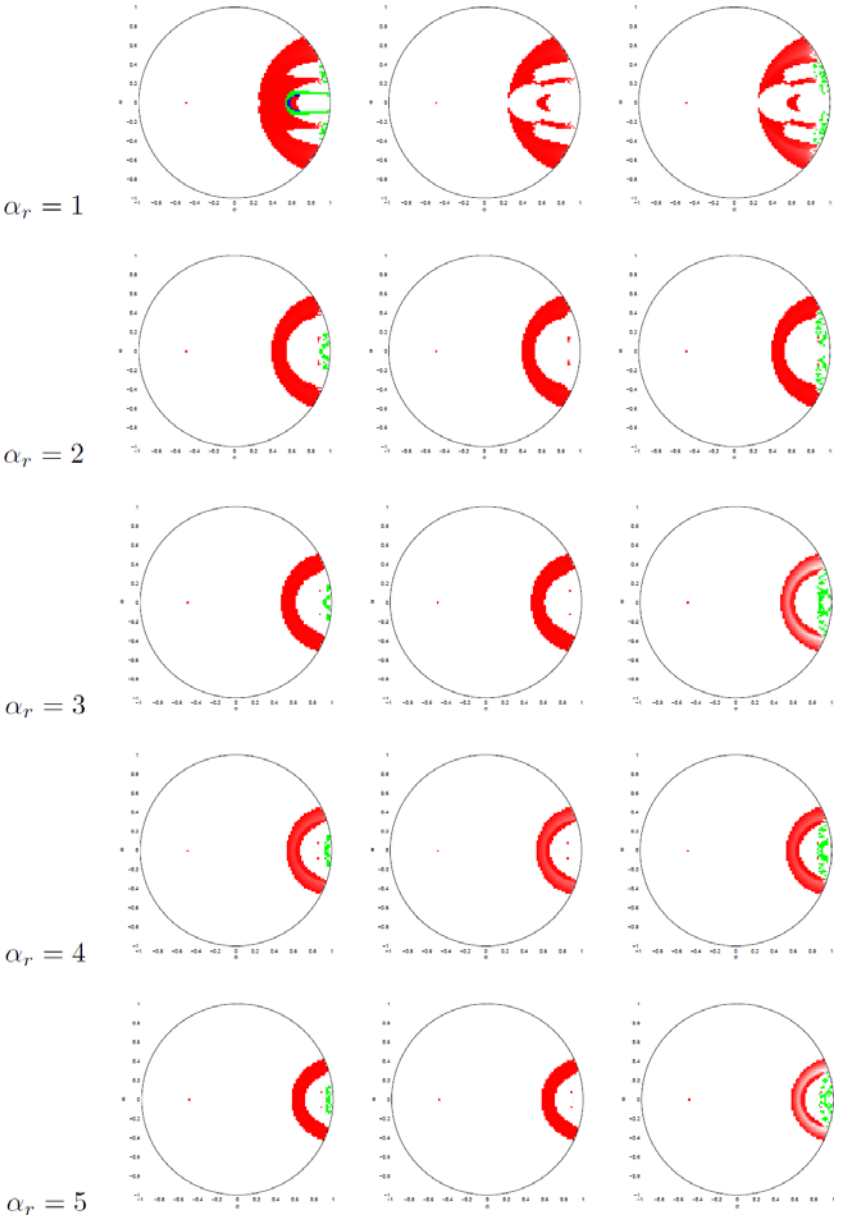


Fig. 2. Performance surfaces for P1, $d = 1$

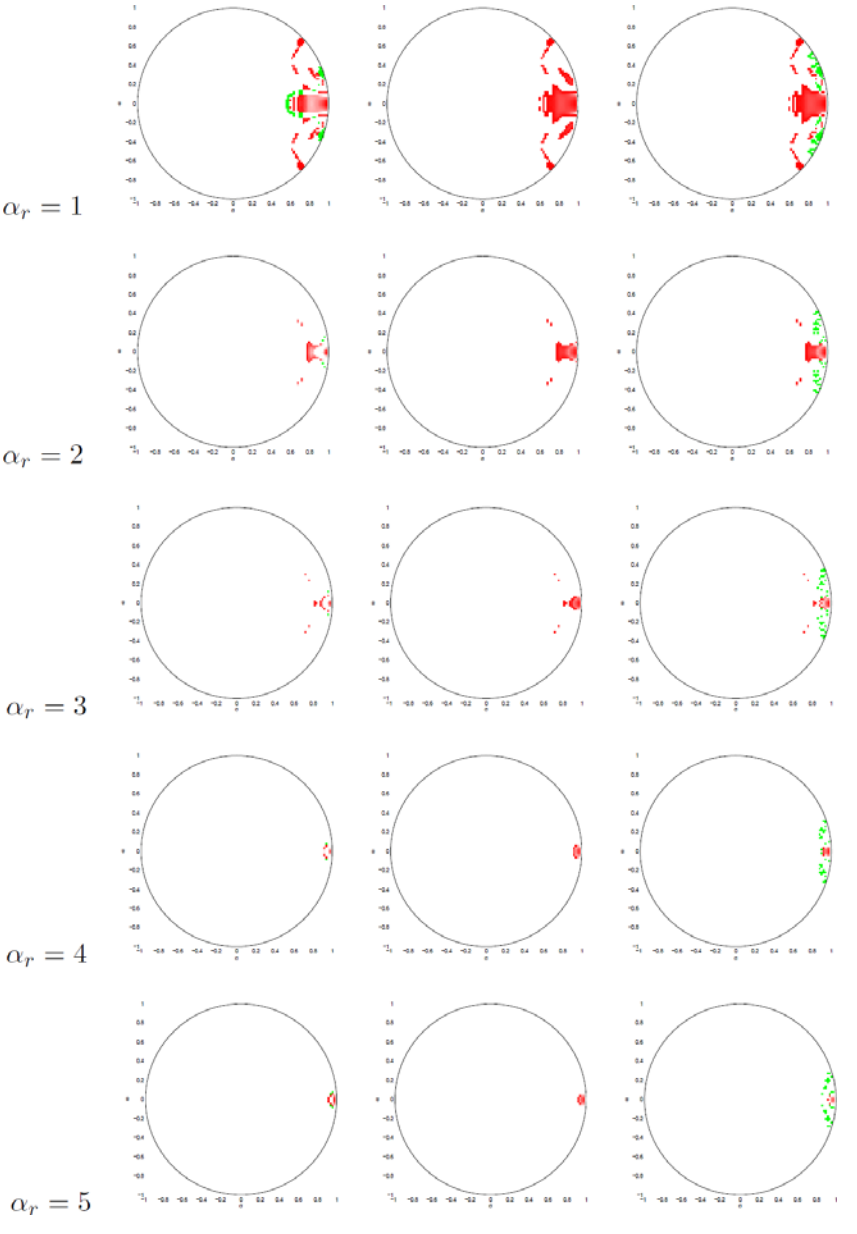


Fig. 3. Performance surfaces for P1, $d = 3$

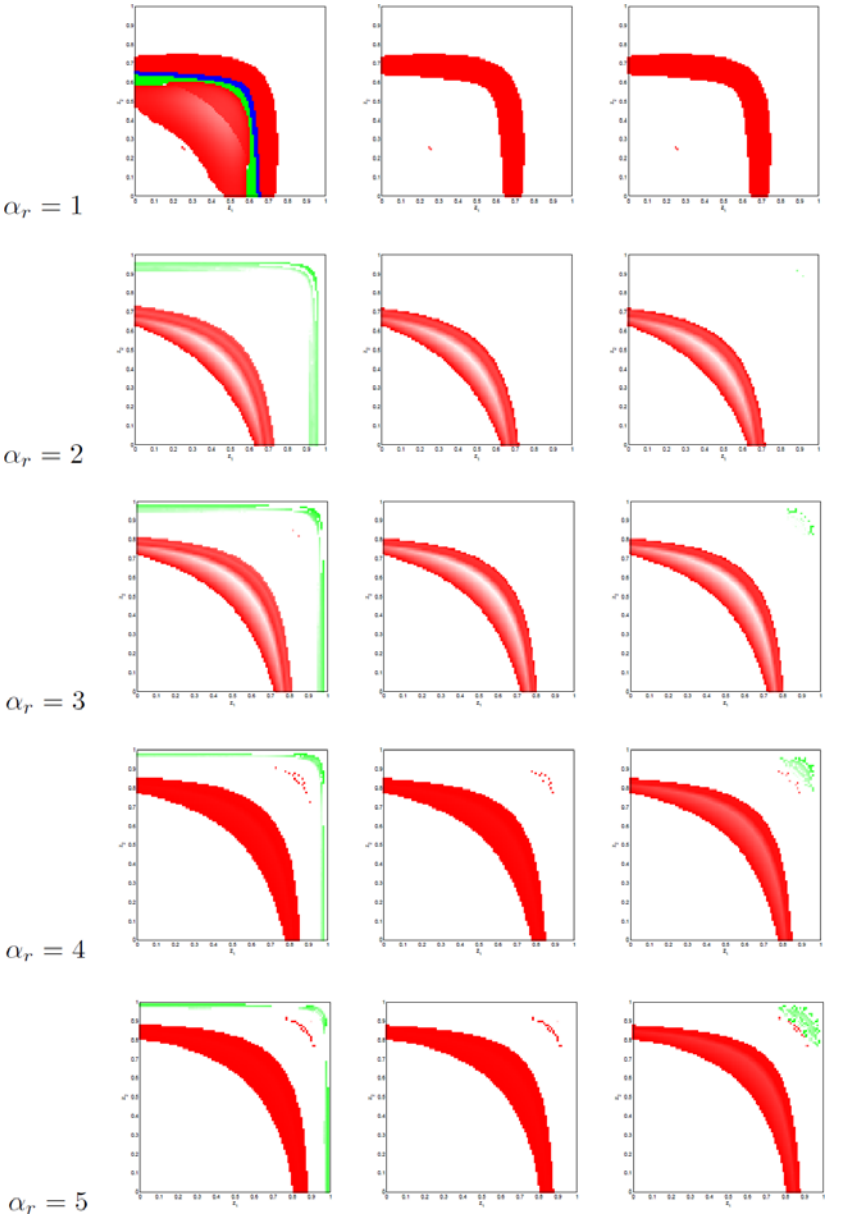


Fig. 4. Performance surfaces for P2, $d = 1$

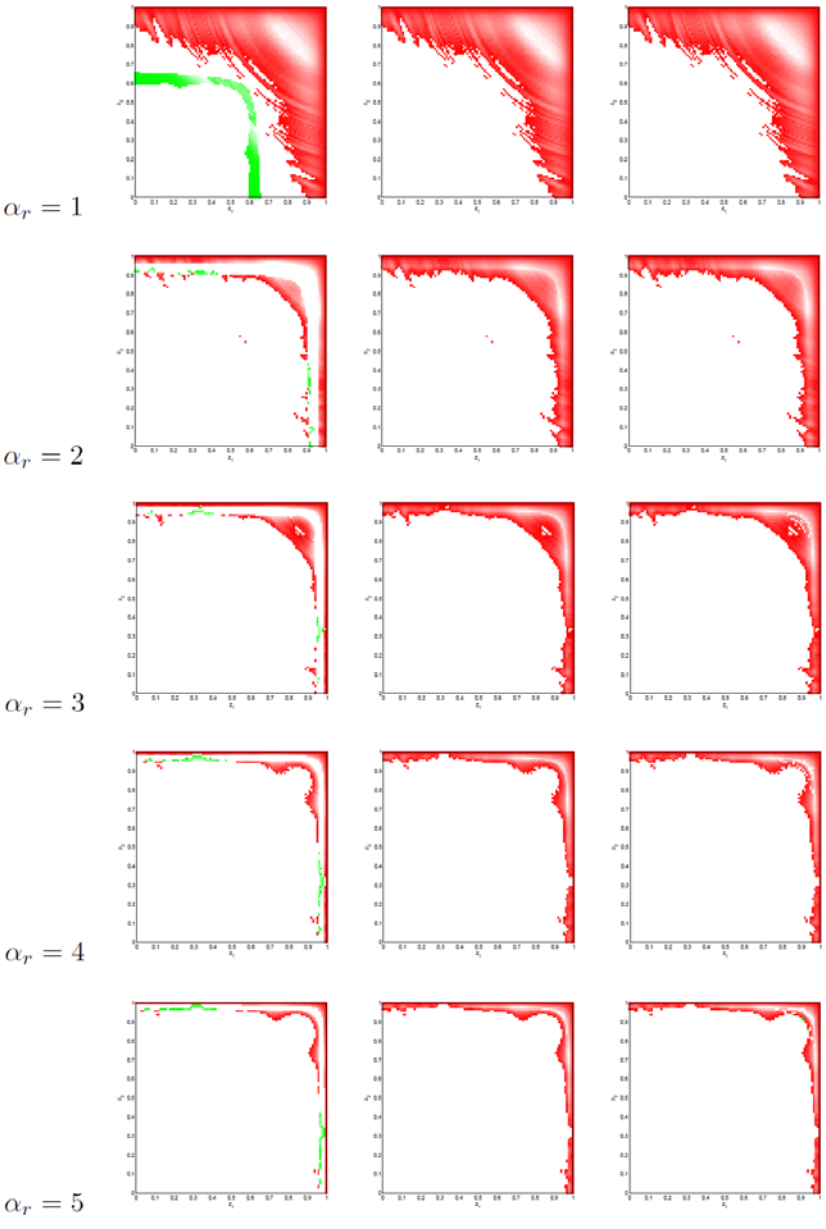


Fig. 5. Performance surfaces for P2, $d = 3$

References

1. Horla, D.: Simple Anti-integrator Windup Compensators – Performance Analysis. Studies in Control and Computer Science 32, 85–102 (2007)
2. Horla, D., Krolikowski, A.: Anti-windup Circuits in Adaptive Pole-placement Control. In: 7th European Control Conference CD-ROM (2003)
3. Horla, D., Krolikowski, A.: Anti-windup Compensators for Adaptive PID Controllers. In: 9th IEEE International Conference on Methods and Models in Automation and Robotics, pp. 575–580 (2003)
4. Rundqwist, L.: Anti-reset Windup for PID Controllers. Ph.D. Thesis, Lund University of Technology (1991)
5. Walgama, K.S., Sternby, J.: Inherent Observer Property in a Class of Anti-windup Compensators. International Journal of Control 52(3), 705–724 (1990)
6. Walgama, K.S., Sternby, J.: On the Convergence Properties of Adaptive Pole-placement Controllers with Anti-windup Compensators. IEEE Transactions on Automatic Control 38(1), 128–132 (1993)
7. Visioli, A.: Modified Anti-windup Scheme for PID Controllers. IEE Proceedings-D 150(1), 49–54 (2003)

Interdisciplinary Modeling of Autonomous Systems Deployed in Uncertain Dynamic Environments

Manuela L. Bujorianu and Marius C. Bujorianu

School of Mathematics, University of Manchester, U.K.
{Manuela,Marius}.Bujorianu@manchester.ac.uk

Abstract. The autonomous systems that operate in physical, random and highly dynamical environments involve a large amount of computation, which depends on the number of parameters. We tackle the problem of complexity reduction for these systems based on an interdisciplinary and qualitative approach. We define concepts like qualitative model reduction and adaptive bismulation, and use them to investigate the stochastic model checking. These concepts prove to be very useful in capturing the co-evolution between system and its environment. Potential applications in renewable energies are discussed.

Keywords: Autonomous systems, Uncertain cyber-physical systems, Adaptive bisimulation, System environment co-evolution, Stochastic model checking, Renewable energy.

1 Introduction

System autonomy is becoming a crucial feature of the modern systems. For systems deployed in remote locations like wind turbines, or space systems, the processes of monitoring and control are increasingly difficult to execute. For such systems, nowadays called cyber physical systems, the autonomic capacity of self monitoring, self optimizing and self healing is essential. A major problem in achieving autonomy is raised by the massive amount of computation necessary to evaluate various (optimization, performance or safety) criteria. In most cases, the computation exponentially depends on the number of parameters. Therefore, reducing the state space size for reasoning about the system behaviour represents a major research topic for autonomous cyber physical systems.

Currently, in the research community, there are several concurrent definitions for cyber physical systems. For our purposes, a cyber physical system involves a tight interaction between physical and computational processes [14]. This interaction can be due to the presence of heterogeneous discrete/continuous components (like the case of hybrid systems) or because of the system deployment within an uncertain dynamic physical environment (like the case of embedded systems).

In this paper, we introduce a general model for uncertain cyber physical systems operating in a random dynamical environment. In this model, the environment influence on the system is modelled implicitly using stochastic differential equations. The computational processes are modelled in a computer science fashion, as sequences of discrete transitions. The interaction between physical and computational processes is

J.A. Cetto et al. (Eds.): Informatics in Control Automation and Robotics, LNEE 85, pp. 337–349.
springerlink.com

captured by interleaving using stochastic kernels. For this general model, we define an abstract measure for performance/safety in the form of a probability of reaching a given target state set. The evaluation of this measure is called probabilistic model checking in computer science and stochastic reachability analysis in control engineering. For the general model we consider, this evaluation will be called stochastic model checking.

For stochastic model checking, efficient computational methods exist only for certain classes of systems. For uncertain cyber physical systems, obtaining efficient evaluation methods is still an open problem. This situation is also due to underdeveloped mathematical investigation of these systems. In this work, we address this problem combining tools from mathematics and computer science in the form of Hilbertian formal methods, which is a recent modelling paradigm based on functional analysis. We present a method for reducing the size of the state space up to a single dimension, if necessary. This method resembles of model order reduction from numerical analysis, but it is casted in a qualitative manner. This is why, the method is called qualitative model reduction. As an application, we define adaptive bisimulation which is a behavioural equivalence concept. Another application is to model co-evolution between the system and its environment and we show the co-evolution is preserved by the qualitative model reduction.

In the end, we propose an application area drawn from renewable energy industries. Specifically, wind turbines are uncertain cyber physical systems that closely interact with a very dynamic physical environment. For wind turbines, it is desirable to have autonomous controllers that self optimize the turbine performance according with the changes within environment.

2 The Mathematical Setting

2.1 Uncertain Cyber-physical Systems

The theory of hybrid systems is a well-established modeling paradigm for embedded systems. Similarly, the theory of concurrent embedded hybrid systems [3] constitutes a suitable modeling framework for CPS. In the following an uncertain cyber-physical system is modeled as a randomized embedded hybrid system.

There are two major ways to randomize a continuous or hybrid dynamical system: In one approach, the concept of noise is used to model small random perturbations. The randomized system has trajectories that closely resemble those of the deterministic initial system. The noise based randomization is carried out using stochastic differential equations. When the influence of the random perturbation changes dramatically the system evolution, the randomization is carried out using stochastic kernels that replace the concept of reset maps from deterministic hybrid system models.

A UCPS $U = (Q, \mathcal{X}, F, R, \lambda)$ consists of

- a finite set of discrete variables Q;
- a map $\mathcal{X} : Q \to \mathbb{R}^{d(\cdot)}$ that sends each $q \in Q$ into a mode (an open subset) X^q of $\mathbb{R}^{d(q)}$, where $d(q)$ is the Euclidean dimension of the corresponding mode;
- a map $F : Q \to 2^{\mathcal{F}_{SDE}}$ which specifies the continuous evolution of the automaton in terms of stochastic differential equations (SDE) over the continuous state x^q for each mode;

• a family of stochastic kernels $R = (R^q)_{q \in Q}$,

$$R^q : \overline{X}^q \times (\cup \mathcal{B}(X^j) | j \in Q \backslash \{q\}) \rightarrow [0, 1]; \tag{1}$$

• a transition rate function

$$\lambda : (\cup \overline{X}^j | j \in Q) \rightarrow \mathbb{R}^+, \tag{2}$$

which gives the distributions of the jump times.

The executions of a UCPS can be described as follows: start with an initial point $x_0 \in X^q$, follow a solution of the SDE associated to X^q, jump when this trajectory hits the boundary or according with the transition rate λ (the jump time is the minimum of the boundary hitting time and the time, which is exponentially distributed with the transition rate λ). Under standard assumptions, for each initial condition $x \in j \in Q \cup X^j$, the possible trajectories starting from x, form a stochastic process. Moreover, for all initial conditions x, the executions of a UCPS form the semantics, which can be thought of as a Markov process in a general setting. Let us consider $M = (\Omega, \mathcal{F}, \mathcal{F}_t, x_t, P_x)$ be the semantics of U. Under mild assumptions on the parameters of U, M can be viewed as a family of Markov processes with the state space $(X, \mathcal{B})$, where X is the union of modes and $\mathcal{B}$ is its Borel σ-algebra. Let $\mathcal{B}^b(X)$ be the lattice of bounded positive measurable functions on X. The meaning of the elements of M can be found in any source treating continuous-parameter Markov processes (see, for example, [9]). Suppose we have given a σ-finite measure μ on $(X, \mathcal{B})$.

In the following we give some operator characterizations of stochastic processes, which are employed in this paper to define a qualitative model reduction for UCPS.

2.2 Hilbertean Formal Methods

The HFM [4] abstract away the analytical properties of deterministic and stochastic differential operators using the so called kernel operator (defined in the following). Using methods of functional analysis HFM elegantly generalize both deterministic and stochastic systems. In this work we focus on the stochastic case. Let us describe briefly the mathematical apparatus that is usually employed to study continuous time continuous space Markov processes.

The transition probability function is $p_t(x, A) = P_x(x_t \in A)$, $A \in \mathcal{B}$. This is the probability that, if $x_0 = x$, x_t will lie in the set A. The operator semigroup $\mathcal{P}$ is defined by $P_t f(x) = \int f(y) p_t(x, dy) = E_x f(x_t), \forall x \in X$, where E_x is the expectation w.r.t. P_x. The operator resolvent $\mathcal{V} = (V_\alpha)_{\alpha \geq 0}$ associated with $\mathcal{P}$ is $V_\alpha f(x) = \int_0^\infty e^{-\alpha t} P_t f(x) dt$, $x \in X$. Let denote by V the initial operator V_0 of $\mathcal{V}$, which is known as the kernel operator of the Markov process M. The operator resolvent $(V_\alpha)_{\alpha \geq 0}$ is the Laplace transform of the semigroup.

The strong generator $\mathcal{L}$ is the derivative of P_t at $t = 0$. Let $D(\mathcal{L}) \subset \mathcal{B}_b(X)$ be the set of functions f for which the following limit exists (denoted by $\mathcal{L}f$): $\lim_{t \searrow 0} \frac{1}{t}(P_t f - f)$.

In the HFM, there is developed a semantic framework for concurrent embedded systems constructed using energy forms. We specialize this theory for function spaces, reaching in this way the theory of Dirichlet forms [12].

A quadratic form $\mathcal{E}$ can be associated to the generator of a Markov process in a natural way.

Let $L^2(X,\mu)$ be the space of square integrable μ-measurable extended real valued functions on X, w.r.t. the natural inner product $< f, g >_\mu = \int f(x)g(x)d\mu(x)$.
The quadratic form $\mathcal{E}$:

$$\mathcal{E}(f,g) = - < \mathcal{L}f, g >_\mu, f \in D(\mathcal{L}), g \in L^2(X,\mu) \tag{3}$$

defines a closed form. This leads to another way of parameterizing Markov processes. Instead of writing down a generator one starts with a quadratic form. As in the case of a generator it is typically not easy to fully characterize the domain of the quadratic form. For this reason one starts by defining a quadratic form on a smaller space and showing that it can be extended to a closed form in subset of $L^2(\mu)$. When the Markov process can be initialized to be stationary, the measure μ is typically this stationary distribution (see [9] p.111). More generally, μ does not have to be a finite measure.

A coercive closed form is a quadratic form $(\mathcal{E}, D(\mathcal{E}))$ with $D(\mathcal{E})$ dense in $L^2(X,\mu)$, which satisfies the: (i) closeness axiom, i.e. its symmetric part is positive definite and closed in $L^2(X,\mu)$, (ii) continuity axiom. $\mathcal{E}$ is called bilinear functional energy (BLFE) if, in addition, it satisfies the third axiom: (iii) contraction condition, i.e. $\forall u \in D(\mathcal{E})$, $u^* = u^+ \wedge 1 \in D(\mathcal{E})$ and $\mathcal{E}(u \pm u^*, u \mp u^*) \geq 0$. For the general theory of closed forms associated with Markov processes see [12].

Let $(\mathcal{L}, D(\mathcal{L}))$ be the generator of a coercive form $(\mathcal{E}, D(\mathcal{E}))$ on $L^2(X,\mu)$, i.e. the unique closed linear operator on $L^2(X,\mu)$ such that $1 - \mathcal{L}$ is onto, $D(\mathcal{L}) \subset D(\mathcal{E})$ and $\mathcal{E}(u,v) =< -\mathcal{L}u, v >$ for all $u \in D(\mathcal{L})$ and $v \in D(\mathcal{E})$. Let $(T_t)_{t>0}$ be the strongly continuous contraction semigroup on $L^2(X,\mu)$ generated by $\mathcal{L}$ and $(G_\alpha)_{\alpha>0}$ the corresponding strongly continuous contraction semigroup (which exist according to the Hille-Yosida theorem).

A right process M with the state space X is associated with a BLFE $(\mathcal{E}, D(\mathcal{E}))$ on $L^2(X,\mu)$ if the semigroup (P_t) of the process M is a μ-version[1] of the form semigroup (T_t). It has been proved [1] and [12] that only those BLFEs, which satisfy some regularity conditions can be associated with some right Markov processes and viceversa (Th.1.9 of [1]).

Prop. 4.2 from [1] states that two right Markov processes M and M' with state space X associated with a common quasi-regular BLFE $(\mathcal{E}, D(\mathcal{E}))$ are stochastically equivalent [12]. That means a quasi-regular BLFE characterizes a class of stochastically equivalent right Markov processes.

Let $M = (\Omega, \mathcal{F}, \mathcal{F}_t, x_t, P_x)$ be a right Markov process with the state space X. Now assume that X is a Lusin space (i.e. it is homeomorphic to a Borel subset of a compact metric space) and $\mathcal{B}(X)$ or $\mathcal{B}$ is its Borel σ-algebra. Assume also that μ is a σ-finite measure on $(X, \mathcal{B})$ and μ is a stationary measure of the process M. Let $X^\#$ another Lusin space (with $\mathcal{B}^\#$ its Borel σ-algebra) and $F : X \to X^\#$ be a measurable function. Let $\sigma(F)$ be the sub-σ-algebra of $\mathcal{B}$ generated by F. If μ is a probability measure then the projection operator between $L^2(X, \mathcal{B}, \mu)$ and $L^2(X, \sigma(F), \mu)$ is the conditional expectation $E_\mu[\cdot|F]$. Recall that E_μ is the expectation defined w.r.t. P_μ and

[1] I.e., for all $f \in L^2(X,\mu)$ the function $P_t f$ is a μ-version (differs on a set of μ-measure zero) of $T_t f$ for all $t > 0$.

that $P_\mu(A) = \int P_x(A)d\mu$, $A \in \mathcal{F}$. We denote by $\mu^\#$ the image of μ under F, i.e. $\mu^\#(A^\#) = \mu(F^{-1}(A^\#))$, for all $A^\# \in \mathcal{B}^\#$. In general, anything associated with $X^\#$ will carry the #-superscript symbol in this section.

Let $\mathcal{E}$ be the BLFE on $L^2(X, \mu)$ associated to M. F induces a form $\mathcal{E}^\#$ on $L^2(X^\#, \mu^\#)$ by

$$\mathcal{E}^\#(u^\#, v^\#) = \mathcal{E}(u^\# \circ F, v^\# \circ F); \tag{4}$$

for $u^\#, v^\# \in D[\mathcal{E}^\#]$, where

$$D[\mathcal{E}^\#] = \{u^\# \in L^2(X^\#, \mu^\#) | u^\# \circ F \in D[\mathcal{E}]\}. \tag{5}$$

It can be shown (see Prop.1.4 in [11]), under a mild condition on the conditional expectation operator $E_\mu[\cdot|F]$ that $\mathcal{E}^\#$ is a BLFE. If, in addition, $\mathcal{E}^\#$ is quasi-regular then we can associate it a right Markov process $M^\# = (\Omega, \mathcal{F}, \mathcal{F}_t, x_t^\#, P_x^\#)$ with the state space $X^\#$. The process $M^\#$ is called the induced Markov process w.r.t. to the proper map F. If the image of M under F is a right Markov process then $x_t^\# = F(x_t)$. The process $M^\#$ might have some different interpretations like a refinement of discrete transitions structure, or an approximation of continuous dynamics or an abstraction of the entire process. It is difficult to find a practical condition to impose on F, which would guarantee that $\mathcal{E}^\#$, as defined by (4) and (5), is also quasi-regular. To circumvent this problem, it is possible to restrict the original domain $D[\mathcal{E}^\#]$ and impose some regularity conditions on F (for more details, see [11]).

Assumption 1. Suppose that $\mathcal{E}^\#$ is a quasi-regular BLFE.

3 Stochastic Model Checking

Let us consider $M = (\Omega, \mathcal{F}, \mathcal{F}_t, x_t, P_x)$ a strong Markov process, which is the semantics of a UCPS. For this strong Markov process we address a verification problem consisting of the stochastic reachability problem defined as follows. Given a set $A \in \mathcal{B}(X)$ and a time horizon $T > 0$, let us to define [6]:

$$\begin{aligned} Reach_T(A) &= \{\omega \in \Omega \mid \exists t \in [0, T] : x_t(\omega) \in A\} \\ Reach_\infty(A) &= \{\omega \in \Omega \mid \exists t \geq 0 : x_t(\omega) \in A\}. \end{aligned} \tag{6}$$

These two sets are the sets of trajectories of M, which reach the set A (the flow that enters A) in the interval of time $[0, T]$ or $[0, \infty)$.

The reachability problem consists of determining the probabilities of such sets. The reachability problem is well-defined, i.e. $Reach_T(A)$, $Reach_\infty(A)$ are indeed measurable sets. Then the probabilities of reach events are

$$P(T_A < T) \text{ or } P(T_A < \infty) \tag{7}$$

where $T_A = \inf\{t > 0 | x_t \in A\}$ and P is a probability on the measurable space $(\Omega, \mathcal{F})$ of the elementary events associated to M. P can be chosen to be P_x (if we want to consider the trajectories, which start in x) or P_μ (if we want to consider the trajectories, which start in x_0 given by the distribution μ).

Usually a target set A in the state space is a level set for a given function $F : X \to \mathbb{R}$, i.e. $A = \{x \in X | F(x) > l\}$ (F can be chosen as the Euclidean norm or as the distance to the boundary of E). The probability of the set of trajectories, which hit A until time horizon $T > 0$ can be expressed as

$$P(\sup F(x_t) | t \in [0, T]) > l. \tag{8}$$

Our goal is to define a new stochastic process $M^{\#}$ (qualitative model reduction) such that the probabilities (7) are preserved. Ideally, since (7) can be written as (8), $F(x_t)$ would represent the best candidate for defining a possible qualitative model reduction for M, which preserves the reach set probabilities. The main difficulty is that $F(x_t)$ is a Markov process only for special choices of F [13]. The problem is how to choose an appropriate F.

Note, if $A^{\#}$ is open in $X^{\#}$ and $A = F^{-1}(A^{\#})$, then we consider the two first hitting times T_A (w.r.t. M) and $T^{\#}_{A^{\#}}$ (w.r.t. $M^{\#}$) of A and $A^{\#}$, respectively. Recall that $T_A = \inf\{t > 0 | x_t \in A\}$.

The following results show that the stochastic model checking problem is solvable for uncertain cps.

Proposition 1. *Under the assumption 1, if μ is a probability measure and $\xi = +\infty$ (M has no killing), then*

$$E_{\mu} \exp(-T_A) \leq E_{\mu^{\#}} \exp(-T^{\#}_{A^{\#}}) \tag{9}$$

where E_{μ} (resp. $E_{\mu^{\#}}$) is the expectation defined w.r.t. P_{μ} (resp. $P_{\mu^{\#}}$).

Proof. Using Th.3.1 from [1], we have

$$E_{\mu^{\#}} \exp(-T^{\#}_{A^{\#}}) = \mathcal{E}^{\#}_1(e_{A^{\#}}, \widehat{e}_{A^{\#}}); E_{\mu} \exp(-T_A) = \mathcal{E}_1(e_A, \widehat{e}_A)$$

Here, $e_{A^{\#}}$ and $\widehat{e}_{A^{\#}}$ are the potential and respectively the copotential of $A^{\#}$ w.r.t. $\mathcal{E}^{\#}$. $e_{A^{\#}}$ is the 1-*reduced of* 1 on $A^{\#}$, i.e. is the unique 1-excessive function in $D[\mathcal{E}^{\#}]$ such that $e_{A^{\#}} = 1$ $\mu^{\#}$-a.e. on $A^{\#}$ and $\mathcal{E}^{\#}_1(e_{A^{\#}}, v^{\#}) = 0$, $\forall v^{\#} \in D[\mathcal{E}^{\#}]_{X^{\#} \setminus A^{\#}}$ (the elements of $D[\mathcal{E}^{\#}]$, which take the value 0 on $A^{\#}$). $\widehat{e}_{A^{\#}}$ is the 1-*coreduced of* 1 on $A^{\#}$ defined correspondingly with the two entries of $\mathcal{E}^{\#}$ interchanged (see [12], for detailed definitions and characterizations). In the same way, e_A and $\widehat{e}_A$ are defined w.r.t. $\mathcal{E}$. Then $e_{A^{\#}} \circ F = 1$ μ-a.e. on $F^{-1}(A^{\#})$ and $\mathcal{E}_1(e_{A^{\#}} \circ F, v^{\#} \circ F) = 0$, $\forall v^{\#} \in D[\mathcal{E}^{\#}]_{X^{\#} \setminus A^{\#}}$ or $\mathcal{E}_1(e_{A^{\#}} \circ F, v) = 0$, $\forall v \in D[\mathcal{E}]_{X|A}$ such that there exists $v^{\#} \in D[\mathcal{E}^{\#}]_{X^{\#} \setminus A^{\#}}$ and $v = v^{\#} \circ F$. Moreover, it can be easily checked, using (14), that $e_{A^{\#}} \circ F$ is 1-excessive function in $D[\mathcal{E}]$. Then $e_{A^{\#}} \circ F$ might be a candidate for the potential of A w.r.t. $\mathcal{E}$ if F is 'compatible' with the cone of excessive functions associated to the given process (if $Fx = Fx' = x^{\#} \in X^{\#} \setminus A^{\#}$ then $v(x) = v(x'), \forall v \in D[\mathcal{E}]_{X \setminus A}$). We have similar relations for $\widehat{e}_{A^{\#}}$. Therefore

$$\begin{aligned}
&E_{\mu^{\#}} \exp(-T^{\#}_{A^{\#}}) = \mathcal{E}^{\#}_1(e_{A^{\#}}, \widehat{e}_{A^{\#}}) = \mathcal{E}^{\#}(e_{A^{\#}}, \widehat{e}_{A^{\#}}) + < e_{A^{\#}}, \widehat{e}_{A^{\#}} >_{\mu^{\#}} \\
&= \mathcal{E}(e_{A^{\#}} \circ F, \widehat{e}_{A^{\#}} \circ F) + < e_{A^{\#}} \circ F, \widehat{e}_{A^{\#}} \circ F >_{\mu} = \mathcal{E}_1(e_{A^{\#}} \circ F, \widehat{e}_{A^{\#}} \circ F) \\
&= \mathcal{E}_1(e_A, \widehat{e}_A) + \mathcal{E}_1(e_{A^{\#}} \circ F - e_A, \widehat{e}_{A^{\#}} \circ F - \widehat{e}_A) \\
&\geq \mathcal{E}_1(e_A, \widehat{e}_A) = E_{\mu} \exp(-T_A)
\end{aligned}$$

Last inequality is the result of the fact that $\mathcal{E}_1(e_{A^\#} \circ F - e_A, \widehat{e}_{A^\#} \circ F - \widehat{e}_A) \geq 0$. This comes from the characterization of e_A as the smallest function, which is equal to 1 on A and is bounded by 1 in rest of the space (Prop.1.5[12]). Then $e_A \leq e_{A^\#} \circ F$. With a similar argument we get that $\widehat{e}_A \leq \widehat{e}_{A^\#} \circ F$. In the symmetric case, the result is a consequence of the Th.1.8 from [11].

If M is the semantics of a UCPS U, given a target state set $A \in \mathcal{B}(X)$, then the goal in the stochastic reachability analysis is to compute the probability $P_\mu(T_A \leq T)$ for a finite horizon time $T > 0$. We now translate the relation (9) in terms of probability of the reachable sets.

Proposition 2. *Under the assumption.1, if μ is a probability measure, then*
$P_\mu(T_A \leq T) \leq eK \min\{T\mathcal{E}^\#(u^\#, u^\#) + < u^\#, u^\# >_{\mu^\#} | u^\# \in D(\mathcal{E}^\#), u^\# \geq 1,\ \mu^\# - a.e.\ on\ A^\#\}$
where $K > 0$ is the sector constant of $\mathcal{E}$.

4 Adapting Verification to Co-evolution

The idea is to apply a "state space reduction" technique based on the general 'induced BLFEs' method to achieve qualitative model reductions for UCPS. With this technique, the semantics of UCPS are 'approximated' by a one-dimensional stochastic process with a much smaller state space.

4.1 Qualitative Model Reduction

The stochastic reachability definition gives the idea to introduce the following concept of qualitative model reduction (QMR) for UCPS.

Definition 1. *Given a right Markov process M defined on the Lusin state space $(X, \mathcal{B})$, and $F : X \to \mathbb{R}$ a measurable weight function, suppose that assumption.1 is fulfilled. The process $M^\#$ associated to the induced BLFE $\mathcal{E}^\#$ under function F is called a QMR of M.*

Let U be a UCPS and M its semantics. Suppose that M is a right Markov process defined on the Lusin state space $(X, \mathcal{B})$.

Definition 2. *Any UCPS $U^\#$ whose semantics is a QMR of M is called a QMR of U.*

Let U be a UCPS and M its semantics (that is a right Markov process, with the state space X).

Proposition 3. *If M is a diffusion then any QMR $M^\#$ of M is a diffusion.*

Proposition 4. *If M is a jump process then any QMR $M^\#$ of M is again a jump process.*

Proof. This statement can be obtained as a consequence of the abstract version of the Kolmogorov backward equations [9]

$$\frac{\partial}{\partial t}P_t f(x) = LP_t f(x),\ P_0 f = f,\ f \in D(\mathcal{L}) \tag{10}$$

and the equality (12). If the equations (10) are associated to an initial diffusion process (resp. jump process) then the relation (12) allow to obtain the fact that the transition probabilities of the induced process satisfy a similar equation, such that the induced process is still a diffusion process (resp. jump process). The same conclusion can be obtain using the stochastic calculus of BLFEs [11]. ■

Since the semantics of a UCPS is, in most cases, a stochastic process, which can be viewed an interleaving between some diffusion processes and a jump process (see [5] for a very general model for UCPS and its semantics as a Markov string), we can write the following result as a corollary of Prop.3.

Proposition 5. *Any QMR of a UCPS is again a UCPS.*

Let $(\mathcal{L}, D(\mathcal{L}))$ and $(\mathcal{L}^{\#}, D(\mathcal{L}^{\#}))$ be the generators of $\mathcal{E}$ and $\mathcal{E}^{\#}$, respectively. For the following results we suppose that the Ass.1 is fulfilled.

Proposition 6. *The generators $\mathcal{L}$ and $\mathcal{L}^{\#}$ are related as follows*

$$\mathcal{L}(u^{\#} \circ F) = \mathcal{L}^{\#}u^{\#} \circ F, \forall u^{\#} \in D(\mathcal{L}^{\#}) \tag{11}$$

Theorem 1. *For all $A^{\#} \in \mathcal{B}^{\#}(X^{\#})$ and for all $t > 0$ we have*

$$p_t^{\#}(Fx, A^{\#}) = p_t(x, F^{-1}(A^{\#})) \tag{12}$$

where $(p_t^{\#})$ and (p_t) are the transition functions of $M^{\#}$ and M, respectively.

Proof. Let $F^{\#}$ be defined as $F^{\#} : \mathcal{B}^b(X^{\#}) \to \mathcal{B}^b(X)$; $F^{\#}u^{\#} = u^{\#} \circ F$. Then (11) becomes $(\mathcal{L} \circ F^{\#})u^{\#} = (F^{\#} \circ \mathcal{L}^{\#})u^{\#}, \forall u^{\#} \in D(\mathcal{L}^{\#})$ $(**)$. For a strong Markov process, the opus of the kernel operator is the inverse operator of the infinitesimal generator of the process. Now, from $(**)$ we get a similar relation between the kernel operators V and $V^{\#}$ of the processes M and $M^{\#}$, i.e. $F^{\#} \circ V^{\#} = V \circ F^{\#}$ on $\mathcal{B}^b(X^{\#})$, or

$$V^{\#}u^{\#} \circ F = V(u^{\#} \circ F), \forall u^{\#} \in \mathcal{B}^b(X^{\#}) \tag{13}$$

since if $u^{\#} \in \mathcal{B}^b(X^{\#})$ then $V^{\#}u^{\#} \in D(\mathcal{L}^{\#})$. For $u^{\#} = 1_{A^{\#}}$ (the indicator function of $A^{\#}$), by the kernel operator integral definition, we obtain (12). ■

Relation (13) implies the following corollary:

Corollary 1. *The semigroups $(P_t^{\#})$ and (P_t) of $M^{\#}$ and M are related by*

$$P_t^{\#}u^{\#} \circ F = P_t(u^{\#} \circ F), \forall u^{\#} \in \mathcal{B}^b(X^{\#}). \tag{14}$$

4.2 Adaptive Bisimulation

In this subsection we define a new concept of adaptive bisimulation for cps. This concept is defined as measurable relation, which induces equivalent BLFEs on the quotient

spaces. In defining adaptive bisimulation, we do not impose the equivalence of the quotient processes, which might not have Markovian properties [13], but we impose the equivalence of the QMRs (that can differ from the quotient processes) associated with the induced BLFEs, with respect to the projection maps.

Let $(X, \mathcal{B}(X))$ and $(Y, \mathcal{B}(Y))$ be Lusin spaces and let $\mathcal{R} \subset X \times Y$ be a relation such that $\Pi^1(\mathcal{R}) = X$ and $\Pi^2(\mathcal{R}) = Y$. We define the equivalence relation on X that is induced by the relation $\mathcal{R} \subset X \times Y$, as the transitive closure of $\{(x, x') | \exists y$ s.t. $(x, y) \in \mathcal{R}$ and $(x', y) \in \mathcal{R}\}$. Analogously, the induced (by $\mathcal{R}$) equivalence relation on Y can be defined. We write $X/_{\mathcal{R}}$ and $Y/_{\mathcal{R}}$ for the sets of equivalence classes of X and Y induced by $\mathcal{R}$. We denote the equivalence class of $x \in X$ by $[x]$. Let

$$\mathcal{B}^{\#}(X) = \mathcal{B}(X) \cap \{A \subset X | \text{ if } x \in A \text{ and } [x] = [x'] \text{ then } x' \in A\}$$

be the collection of all Borel sets, in which any equivalence class of X is either totally contained or totally not contained. It can be checked that $\mathcal{B}^{\#}(X)$ is a σ-algebra. Let $\pi_X : X \to X/_{\mathcal{R}}$ be the mapping that maps each $x \in X$ to its equivalence class and let

$$\mathcal{B}(X/_{\mathcal{R}}) = \{A \subset X/_{\mathcal{R}} | \pi_X^{-1}(A) \in \mathcal{B}^{\#}(X)\}.$$

Then $(X/_{\mathcal{R}}, \mathcal{B}(X/_{\mathcal{R}}))$, which is a measurable space, is called the quotient space of X w.r.t. $\mathcal{R}$. The quotient space of Y w.r.t. $\mathcal{R}$ is defined in a similar way. We define a bijective mapping $\psi : X/_{\mathcal{R}} \to Y/_{\mathcal{R}}$ as

$$\psi([x]) = [y] \text{ if } (x, y) \in \mathcal{R} \text{ for some } x \in [x] \text{ and some } y \in [y].$$

We say that the relation $\mathcal{R}$ is measurable if X and Y if for all $A \in \mathcal{B}(X/_{\mathcal{R}})$ we have $\psi(A) \in \mathcal{B}(Y/_{\mathcal{R}})$ and vice versa, i.e. ψ is a homeomorphism. Then the real measurable functions defined on $X/_{\mathcal{R}}$ can be identified with those defined on $Y/_{\mathcal{R}}$ through the homeomorphism ψ. We can write $\mathcal{B}^b(X/_{\mathcal{R}}) \overset{\psi}{\cong} \mathcal{B}^b(Y/_{\mathcal{R}})$. Moreover, these functions can be thought of as real functions defined on X or Y measurable w.r.t. $\mathcal{B}^{\#}(X)$ or $\mathcal{B}^{\#}(Y)$.

Assumption 2. Suppose that $X/_{\mathcal{R}}$ and $Y/_{\mathcal{R}}$ with the topologies induced by projection mappings are Lusin spaces.

Suppose we have given two right Markov processes M and W with the state spaces X and Y. Assume that μ (resp. ν) is a stationary measure of the process M (resp. W). Let $\mu/_{\mathcal{R}}$ (resp. $\nu/_{\mathcal{R}}$) the image of μ (resp. ν) under π_X (resp. π_Y). Let $\mathcal{E}$ (resp. $\mathcal{F}$) the quasi-regular BLFE corresponding to M (resp. W). The equivalence between the induced processes can be used to define a new bisimulation between Markov processes, as follows.

Definition 3. *Under assumptions 1 and 2, a measurable relation $\mathcal{R} \subset X \times Y$ is a bisimulation between M and W if the mappings π_X and π_Y define the same induced BLFE on $L^2(X/_{\mathcal{R}}, \mu/_{\mathcal{R}})$ and $L^2(Y/_{\mathcal{R}}, \nu/_{\mathcal{R}})$, respectively.*

This definition states that M and W are bisimilar if $\mathcal{E}/_{\mathcal{R}} = \mathcal{F}/_{\mathcal{R}}$. Here, $\mathcal{E}/_{\mathcal{R}}$ (resp. $\mathcal{F}/_{\mathcal{R}}$) is the induced BLFE of $\mathcal{E}$ (resp. $\mathcal{F}$) under the mapping π_X (resp. π_Y). Clearly, this can be possible iff $\mu/_{\mathcal{R}} = \nu/_{\mathcal{R}}$.

Assumption 3. Suppose that $\mathcal{E}/_{\mathcal{R}}$ and $\mathcal{F}/_{\mathcal{R}}$ are quasi-regular BLFE.

Let $M/_{\mathcal{R}}$ (resp. $W/_{\mathcal{R}}$) be the Markov process associated to $\mathcal{E}/_{\mathcal{R}}$ (resp. $\mathcal{F}/_{\mathcal{R}}$).

Proposition 7. *Under assumptions 1, 2 and 3, M and W are stochastic bisimilar under $\mathcal{R}$ iff their QMRs $M/_{\mathcal{R}}$ and $W/_{\mathcal{R}}$ with respect to π_X and, respectively π_Y are $\mu/_{\mathcal{R}}$-equivalent.*

Let U and U' be two UCPSs, with the semantics M and W, strong Markov processes defined on the state spaces $(X, \mathcal{B}(X))$ and $(Y, \mathcal{B}(Y))$, respectively.

Definition 4. *U and U' are called bisimilar if there exist a bisimulation relation under which their semantics M and W are bisimilar.*

5 Applications in Renewable Energy

Typically, the behaviour of a power system is described by a complex combination of continuous and discrete dynamics, where the discrete transitions are usually given by a switching controller. Often, these power systems evolve in an uncertain environment and their behaviour might be the result of the interaction with this environment. Therefore, modeling their dynamics with UCPS seems to be a natural idea. We consider the wind turbine description with the associated dynamical equations developed in [2], but we model this in the framework of UCPS. In [2], the scopewas to compute probability distribution function via forward Kolmogorov equation, method that works pretty well when the dimension of the state space is small (less than five). When the state space dimension is bigger, our methodology to reduce this dimension to one via QMR represents a good alternative. In general, a wind turbine model is composed of three subsystems: the rotor, the generator and the gearbox (that connects the rotor and the generator shafts). The system dynamics is modeled as a stochastic hybrid system whose input and output are respectively the wind speed $v(t)$ and the generator power $P_G(t)$. Looking at the pictorial representation of a wind turbine from Fig. 1 (also discussed in [10]), a simple observation shows that the wind speed plays the role a stochastic input for the wind turbine. We may model the wind speed as stochastic hybrid physical process and the wind turbine as a stochastic hybrid control system. The picture shows that the out of the wind speed is the input of wind turbine, so we have to define interconnected stochastic hybrid systems. This implies that the study of such a system should consider control, simulation, verification problems for the wind turbine model for different modes of the wind speed. Of course, this problem has been studied in the deterministic modelling framework, but now the main challenge will be to use the right uncertain measures for modelling wind influence.

The relation between driving and braking torques and the turbine acceleration is expressed as follows:

$$J \cdot \dot{\omega} = \frac{P_{drive} - P_{brake}}{\omega} \tag{15}$$

where ω is the rotor speed, J is the moment of inertia, P_{drive} is the aerodynamic power captured by the wind turbine, and P_{brake} is the braking power from the generator.

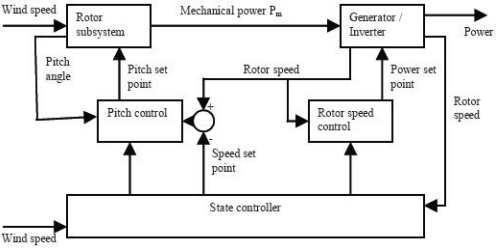


Fig. 1. The structure of a wind turbine

The generator power output is related to the braking power by $P_G = \eta P_{brake}$, where η is a constant. The aerodynamic power is given by the algebraic relation $P_{drive} = \frac{\pi}{2}\rho R^2 c_\rho(\lambda, \theta)v^3$, where ρ is the air density, R is the rotor radius, θ is the pitch angle, $\lambda = R\omega/v$, c_ρ is the power coefficient.

The turbine is operated by a switched controller with two discrete modes:

• In the first mode denoted by q_1 (lower to medium wind speed region), the rotor speed is controlled by driving P_G (the generator power output) to follow a given optimal power schedule $S(\omega)$ (see [2] for the expression of $S(\omega)$)
• In the second mode denoted by q_2 (higher wind speed region), P_G is kept constant to its nominal value $P_{G,nom}$, and aerodynamic power P_{drive} is controlled by the pitch control in order to keep ω around its nominal value ω_{nom}.

The controller switches from the mode q_1 to mode q_2 when $\omega = \omega_{nom}$, and back to the mode q_1 when $\omega = \omega_{12} < \omega_{nom}$. The strict inequality defines a hysteric behaviour that is necessary in order to prevent Zeno phenomena between the two modes.

In the second mode, for the pitch angle is used a proportional controller:

$$\frac{d\theta}{dt} = \begin{cases} 0 & \text{if } \theta = 0 \text{ and } \omega \leq \omega_{nom} \\ 0 & \text{if } \theta = \theta_{\max} \text{ and } \omega \geq \omega_{nom} \\ h(K(\omega - \omega_{nom}) & \text{otherwise} \end{cases} \tag{16}$$

where $h(\widehat{\theta}) = \min(\widehat{\theta}_{\max}, \max(\widehat{\theta}_{\min}, \widehat{\theta}\,)$.

Following [2],we model the horizontal wind speed as a one-dimensional stochastic differential equation (SDE):

$$dv(t) = -\frac{v(t) - \overline{v}}{T}dt + k\overline{v}\sqrt{2/T}dW(t) \tag{17}$$

where $T = L/\overline{v}$, with L the wind turbulence length scale. The SDE (17) defines a one-dimensional diffusion process. A reflecting barrier $v = 0$ is introduced to preserve the positivity of this process.

The evolution of this wind turbine can be described by a UCPS U whose continuous behaviour is described by $x_t = (\omega(t), \theta(t), v(t))$, which obeys to the equations (15), (16) and (17), and the discrete behaviour is described by (q_t) that evolves in $Q = \{q_1, q_2)$. In this case, the reset map R moves backward and forward between the modes q_1 and q_2, without affecting the continuous state. Therefore, the realization

of the UCPS U associated to this wind turbine is nothing else but switching process (in this case, a switching dynamical system combined with a switching diffusion process).

For a standard diffusion process defined on $X = \mathbb{R}^n$ and $F = |\cdot|$, the Euclidean norm (which is clearly a proper map), we consider the corresponding BLFE form:

$$\mathcal{E}(u,v) = \sum_{i,j=1}^{n} \int_{\mathbb{R}^n} a_{ij}(y) \frac{\partial}{\partial y_i} u(y) \frac{\partial}{\partial y_j} v(y) m(y) dy$$

$u, v \in C_0^\infty(\mathbb{R}^n), m \in C^1(\mathbb{R}^n) \cap L^1(\mathbb{R}^n; dy)$ and $m(y) > 0, \forall y \in \mathbb{R}^n, \int_{\mathbb{R}^n} m(y)dy = 1$ $\forall i,j : a_{ij} = a_{ji} \in L^1_{loc}(\mathbb{R}^n; \mu(dy))$ where $\mu(dy) \equiv m(y)dy$, $\exists \delta > 0$ such that $\forall y \in \mathbb{R}^n$

$$\sum_{i,j=1}^{n} a_{ij}(y)\xi_i\xi_j \geq \delta|\xi|^2, \forall \xi \in \mathbb{R}^n.$$

Then the QMR of this standard diffusion process is given by the following induced form

$$\mathcal{E}^{\#}(u^{\#}, v^{\#}) = \int_0^\infty f(r)(u^{\#})'(r)(v^{\#})'(r)dr, u^{\#}, v^{\#} \in D(\mathcal{E}^{\#}), \tag{18}$$

where $a.e.\ r \geq 0$, $f(r) = \int_{|y|=1} \sum_{i,j=1}^{n} a_{ij}(r\theta)\theta_i\theta_j m(r\theta) d\sigma(\theta) r^{n-1}$, σ is the hypersurface measure of $\{y \in \mathbb{R}^n : |y| = 1\}$.

Considering the fact that the evolution of the wind turbine is nothing else but a switching diffusion process, its QMR will be given by a "switching" induced form which is given by the concatenation of some induced forms whose expressions are obtained instantiating the form (18).

6 Final Remarks

In this paper, we have used the concept of energy, which is a key ingredient of Hilbertean formal methods, to define QMR and behavioral equivalence for cyber-physical systems operating in random environments. Energy is a versatile analytical concept that characterizes in a subtle way the interaction between computation and physics, as well as their co-evolution.

Adaptive bisimulation means the energy preservation of the stochastic processes generated by the cyber-physical system evolutions. The energy concept can be also used to define QMRs for cyber-physical systems. Given a QMR function that reduces the state space, we have defined a standard construction that associates a QMR (called standard) on the reduced state space. The mathematical results from Section 4.1 show that the QMR method preserves important analytic properties (related to HFM). Two uncertain CPS are adaptive bisimilar if they have the same energy. The theorem from Section 4.2 shows that two uncertain CPS are adaptive bisimilar iff their standard QMRs are equivalent as Markov processes.

We have formulated the stochastic model checking problem (a subproblem of stochastic reachability analysis, corresponding to the probabilistic model checking of Markov chains). We proved two results that show that the problem is solvable for uncertain cyber-physical systems. The mathematical results from Section 3 provide a upper

bound for the reach set probabilities. In this way, one can prove that the probability of reaching a state in a certain set can be small enough.

In renewable energy industry, it is inherent to use multiple energy sources and then defining a highly heterogeneous global system. Therefore, networked stochastic hybrid system models would be required to model such systems. Then optimization of the produced energy should be studied for these networks. That means the switching from one system to another one has to be made smoothly with no big "jumps" or big fluctuation intervals. Formally, these networks will be modelled by hierarchical UCPS, and then for optimization purpose we need to define and study hierarchical control.

A preliminary version of this paper was reported in [7].

Acknowledgements. This work was funded by the EPSRC project EP/E050441/1 CICADA.

References

1. Albeverio, S., Ma, Z.M., Rockner, M.: Quasi-regular Dirichlet Forms and Markov Processes. J. of Functional Analysis 111, 118–154 (1993)
2. Bect, J., Phulpin, Y., Baili, H., Fleury, G.: On the Fokker-Planck equation for stochastic hybrid systems: application to a wind turbine model (2006)
3. Bujorianu, M.C., Bujorianu, M.L.: A Randomized Model for Communicating Embedded Systems. In: Proceedings of the 16th Mediterranean Conference on Control and Automation, IEEE, Los Alamitos (2008)
4. Bujorianu, M.L., Bujorianu, M.C.: Uncertainty and Reconfigurability in Hilbertean Formal Methods, Technical Report TR-CTIT-08-20 Centre for Telematics and Information Technology, University of Twente (2008)
5. Bujorianu, M.L., Lygeros, J.: General Stochastic Hybrid Systems: Modelling and Optimal Control. In: Proc. 43th Conference in Decision and Control, pp. 182–187. IEEE Press, Los Alamitos (2004)
6. Bujorianu, M.L.: Extended Stochastic Hybrid Systems and their Reachability Problem. In: Alur, R., Pappas, G.J. (eds.) HSCC 2004. LNCS, vol. 2993, pp. 234–249. Springer, Heidelberg (2004)
7. Bujorianu, M.L., Bujorianu, M.C.: Co-evolution Preserving Model Reduction for Uncertain Cyber-physical Systems. In: Filipe, J., Andrade-Cetto, J., Ferrier, J.-L. (eds.) Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics - Signal Processing, Systems Modeling and Control, Milan, Italy, July 2-5, pp. 39–46. INSTICC Press (2009)
8. Ethier, S.N., Kurtz, T.G.: Markov Processes: Characterization and Convergence. John Wiley and Sons, Chichester (1986)
9. Davis, M.H.A.: Markov Models and Optimization. Chapman and Hall, Boca Raton (1993)
10. Bayem, H., Phulpin, Y., Dessante, P., Bect, J.: Probabilistic Computation of Wind Farm Power Generation based on Wind Turbine Dynamic Modelling. PMAPS (2008)
11. Iscoe, I., McDonald, D.: Induced Dirichlet Forms and Capacitary Inequalities. Ann. Prob. 18(3), 1195–1221 (1990)
12. Ma, M., Rockner, M.: The Theory of (Non-Symmetric) Dirichlet Forms and Markov Processes. Springer, Heidelberg (1990)
13. Rogers, L.C.G., Pitman, J.W.: Markov Functions. Ann. Prob. 9(4), 573–582 (1981)
14. Tabuada, P.: Cyber-Physical Systems Position Paper presented at NSF Workshop on Cyber-Physical Systems (2006)

Author Index