

Joaquim Filipe  
Jean-Louis Ferrier

Juan Andrade-Cetto  
*Editors*

# Informatics in Control, Automation and Robotics

Selected Papers from the International  
Conference on Informatics in Control,  
Automation and Robotics 2007

Lecture Notes in Electrical Engineering

---

Volume 24

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

# Informatics in Control, Automation and Robotics

Selected Papers from the International  
Conference on Informatics in Control,  
Automation and Robotics 2007

 Springer

Joaquim Filipe  
INSTICC  
Av. D. Manuel I, 27A 2º Esq.  
2910-595 Setubal  
Portugal  
jfilipe@insticc.org

Juan Andrade Cetto  
Univ. Politecnica Catalunya  
Institut Robotica i  
Informatica Industrial  
Llorens i Artigas, 4-6  
Edifici U  
08028 Barcelona  
Spain  
cetto@cvc.uab.es

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

ISBN: 978-3-540-85639-9

e-ISBN: 978-3-540-85640-5

Library of Congress Control Number: 2008934301

© Springer-Verlag Berlin Heidelberg 2009

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

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

*Cover design:* eStudio Calamar S.L.

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 fourth “International Conference on Informatics in Control Automation and Robotics” (ICINCO 2007), held at the University of Angers, France, from 9 to 12 May 2007. 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 2007 received 435 paper submissions, from more than 50 different countries in all continents. From these, after a blind review process, only 52 were accepted as full papers, of which 22 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 2008, to be held in Funchal, Madeira - Portugal, and in future editions of this conference.

Furthermore, ICINCO 2007 included 3 plenary keynote lectures given by Dimitar Filev (Ford Motor Company), Patrick Millot (Université de Valenciennes) and Mark W. Spong (University of Illinois at Urbana-Champaign).

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 time. 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.

June 2008

Juan A. Cetto  
Jean-Louis Ferrier  
Joaquim Filipe

# Conference Committee

## Conference Co-chairs

Jean-Louis Ferrier, University of Angers, France

Joaquim Filipe, Polytechnic Institute of Setúbal / INSTICC, Portugal

## Program Co-chairs

Juan Andrade Cetto, Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Spain

Janan Zaytoon, CReSTIC, URCA, France

## Organising Committee

Paulo Brito, INSTICC, Portugal

Marina Carvalho, INSTICC, Portugal

Helder Coelhas, INSTICC, Portugal

Andreia Costa, INSTICC, Portugal

Bruno Encarnação, INSTICC, Portugal

Vítor Pedrosa, INSTICC, Portugal

## Programme Committee

Eugenio Aguirre, Spain

Arturo Hernandez Aguirre, Mexico

Frank Allgower, Germany

Fouad AL-Sunni, Saudi Arabia

Bala Amavasai, UK

Francesco Amigoni, Italy

Yacine Amirat, France

Nicolas Andreff, France

Stefan Andrei, Singapore

Plamen Angelov, UK

Luis Antunes, Portugal

Peter Arato, Hungary

Helder Araújo, Portugal

Gustavo Arroyo-Figueroa, Mexico

Marco Antonio Arteaga, Mexico

Vijanth Sagayan Asirvadam, Malaysia

Nikos Aspragathos, Greece

Robert Babuska, The Netherlands

Ruth Bars, Hungary

Karsten Berns, Germany

Robert Bicker, UK

Stjepan Bogdan, Croatia

Patrick Boucher, France

Alan Bowling, USA

Edmund Burke, UK

Kevin Burn, UK

Clifford Burrows, UK

Luis M. Camarinha-Matos, Portugal

Marco Campi, Italy

Marc Carreras, Spain

Jorge Martins de Carvalho, Portugal

Alicia Casals, Spain

Alessandro Casavola, Italy

Christos Cassandras, USA

Riccardo Cassinis, Italy

Raja Chatila, France

Tongwen Chen, Canada  
YangQuan Chen, USA  
Albert M. K. Cheng, USA  
Graziano Chesi, China  
Sung-Bae Cho, Korea  
Ryszard S. Choras, Poland  
Carlos Coello Coello, Mexico  
Patrizio Colaneri, Italy  
António Dourado Correia, Portugal  
Yechiel Crispin, USA  
Keshav Dahal, UK  
Mariolino De Cecco, Italy  
Bart De Schutter, The Netherlands  
Angel P. del Pobil, Spain  
Guilherme DeSouza, USA  
Rüdiger Dillmann, Germany  
Feng Ding, China  
Denis Dochain, Belgium  
Tony Dodd, UK  
Alexandre Dolgui, France  
Marco Dorigo, Belgium  
Petr Ekel, Brazil  
Heinz-Hermann Erbe, Germany  
Gerardo Espinosa-Perez, Mexico  
Simon Fabri, Malta  
Sergej Fatikow, Germany  
Jean-Marc Faure, France  
Jean-Louis Ferrier, France  
Florin Gheorghe Filip, Romania  
Georg Frey, Germany  
Manel Frigola, Spain  
Colin Fyfe, UK  
Dragan Gamberger, Croatia  
Leonardo Garrido, Mexico  
Ryszard Gessing, Poland  
Lazea Gheorghe, Romania  
Maria Gini, USA  
Alessandro Giua, Italy  
Luis Gomes, Portugal  
John Gray, UK  
Dongbing Gu, UK

Jason Gu, Canada  
José J. Guerrero, Spain  
Jatinder (Jeet) Gupta, USA  
Thomas Gustafsson, Sweden  
Maki K. Habib, Japan  
Hani Hagras, UK  
Wolfgang Halang, Germany  
J. Hallam, Denmark  
Riad Hammoud, USA  
Uwe D. Hanebeck, Germany  
John Harris, USA  
Robert Harrison, UK  
Vincent Hayward, Canada  
Dominik Henrich, Germany  
Francisco Herrera, Spain  
Victor Hinostroza, Mexico  
Weng Ho, Singapore  
Wladyslaw Homenda, Poland  
Alamgir Hossain, UK  
Dean Hougen, USA  
Amir Hussain, UK  
Seth Hutchinson, USA  
Atsushi Imiya, Japan  
Sirkka-Liisa Jämsä-Jounela, Finland  
Ray Jarvis, Australia  
Odest Jenkins, USA  
Ping Jiang, UK  
Ivan Kalaykov, Sweden  
Dimitrios Karras, Greece  
Dusko Katic, Serbia  
Graham Kendall, UK  
Uwe Kiencke, Germany  
Jozef Korbicz, Poland  
Israel Koren, USA  
Bart Kosko, USA  
George L. Kovács, Hungary  
Krzysztof Kozłowski, Poland  
Gerhard Kraetzschmar, Germany  
Cecilia Laschi, Italy  
Loo Hay Lee, Singapore  
Soo-Young Lee, Korea

Graham Leedham, Singapore  
 Cees van Leeuwen, Japan  
 Kauko Leiviskä, Finland  
 Kang Li, UK  
 Yangmin Li, China  
 Zongli Lin, USA  
 Cheng-Yuan Liou, Taiwan  
 Vincenzo Lippiello, Italy  
 Honghai Liu, UK  
 Luís Seabra Lopes, Portugal  
 Brian Lovell, Australia  
 Peter Luh, USA  
 Anthony Maciejewski, USA  
 N. P. Mahalik, Korea  
 Bruno Maione, Italy  
 Frederic Maire, Australia  
 Om Malik, Canada  
 Danilo Mandic, UK  
 Jacek Mandziuk, Poland  
 Hervé Marchand, France  
 Philippe Martinet, France  
 Aleix Martinez, USA  
 Aníbal Matos, Portugal  
 Rene V. Mayorga, Canada  
 Barry McCollum, UK  
 Ken McGarry, UK  
 Gerard McKee, UK  
 Seán McLoone, Ireland  
 Basil Mertzios, Greece  
 José Mireles Jr., Mexico  
 Sushmita Mitra, India  
 Vladimir Mostyn, Czech Republic  
 Rafael Muñoz-Salinas, Spain  
 Kenneth Muske, USA  
 Ould Khessal Nadir, Canada  
 Fazel Naghdy, Australia  
 Tomoharu Nakashima, Japan  
 Andreas Nearchou, Greece  
 Luciana Porcher Nedel, Brazil  
 Sergiu Nedeveschi, Romania  
 Maria Neves, Portugal  
 Hendrik Nijmeijer, The Netherlands  
 Juan A. Nolzco-Flores, Mexico  
 Urbano Nunes, Portugal  
 Gustavo Olague, Mexico  
 José Valente de Oliveira, Portugal  
 Andrzej Ordys, UK  
 Djamila Ouelhadj, UK  
 Manuel Ortigueira, Portugal  
 Christos Panayiotou, Cyprus  
 Evangelos Papadopoulos, Greece  
 Panos Pardalos, USA  
 Michel Parent, France  
 Thomas Parisini, Italy  
 Igor Paromtchik, Japan  
 Gabriella Pasi, Italy  
 Witold Pedrycz, Canada  
 Carlos Eduardo Pereira, Brazil  
 Maria Petrou, UK  
 J. Norberto Pires, Portugal  
 Marios Polycarpou, Cyprus  
 Marie-Noëlle Pons, France  
 Libor Preucil, Czech Republic  
 Joseba Quevedo, Spain  
 Robert Reynolds, USA  
 A. Fernando Ribeiro, Portugal  
 Bernardete Ribeiro, Portugal  
 Robert Richardson, UK  
 John Ringwood, Ireland  
 Rodney Roberts, USA  
 Kurt Rohloff, USA  
 Juha Röning, Finland  
 Agostinho Rosa, Portugal  
 Hubert Roth, Germany  
 António Ruano, Portugal  
 Carlos Sagüés, Spain  
 Mehmet Sahinkaya, UK  
 Antonio Sala, Spain  
 Abdel-Badeeh Salem, Egypt  
 Ricardo Sanz, Spain  
 Medha Sarkar, USA  
 Nilanjan Sarkar, USA

Jurek Sasiadek, Canada  
Daniel Sbarbaro, Chile  
Carsten Scherer, The Netherlands  
Carla Seatzu, Italy  
Klaus Schilling, Germany  
Yang Shi, Canada  
Michael Short, UK  
Chi-Ren Shyu, USA  
Bruno Siciliano, Italy  
João Silva Sequeira, Portugal  
Silvio Simani, Italy  
Amanda Sharkey, UK  
Michael Small, China  
Burkhard Stadlmann, Austria  
Tarasiewicz Stanislaw, Canada  
Aleksandar Stankovic, USA  
Raúl Suárez, Spain  
Ryszard Tadeusiewicz, Poland  
Tianhao Tang, China  
Adriana Tapus, USA  
József K. Tar, Hungary  
Daniel Thalmann, Switzerland  
Gui Yun Tian, UK  
Antonios Tsourdos, UK  
Nikos Tsourveloudis, Greece

### **Auxiliary Reviewers**

Rudwan Abdullah, UK  
Luca Baglivo, Italy  
Prasanna Balaprakash, Belgium  
João Balsa, Portugal  
Alejandra Barrera, Mexico  
Frederik Beutler, Germany  
Alecio Binotto, Brazil  
Nizar Bouguila, Canada  
Dietrich Brunn, Germany  
Maria Paola Cabasino, Italy  
Joao Paulo Caldeira, Portugal  
Aneesh Chauhan, Portugal  
Paulo Gomes da Costa, Portugal

Ivan Tyukin, Japan  
Masaru Uchiyama, Japan  
Nicolas Kemper Valverde, Mexico  
Marc Van Hulle, Belgium  
Annamaria R. Varkonyi-Koczy,  
Hungary  
Luigi Villani, Italy  
Markus Vincze, Austria  
Bernardo Wagner, Germany  
Axel Walthelm, Germany  
Lipo Wang, Singapore  
Alfredo Weitzenfeld, Mexico  
Dirk Wollherr, Germany  
Sangchul Won, Korea  
Kainam Thomas Wong, China  
Jeremy Wyatt, UK  
Alex Yakovlev, UK  
Hujun Yin, UK  
Xinghuo Yu, Australia  
Du Zhang, USA  
Janusz Zalewski, USA  
Marek Zaremba, Canada  
Dayong Zhou, USA  
Argyrios Zolotas, UK  
Albert Zomaya, Austrália

Xevi Cufí, Spain  
Sérgio Reis Cunha, Portugal  
Paul Dawson, USA  
Mahmood Elfandi, Libya  
Michele Folgheraiter, Italy  
Diamantino Freitas, Portugal  
Reinhard Gahleitner, Austria  
Nils Hagge, Germany  
Onur Hamsici, USA  
Renato Ventura Bayan Henriques,  
Brazil  
Matthias Hentschel, Germany  
Marco Huber, Germany

Markus Kemper, Germany  
Vesa Klumpp, Germany  
Daniel Lecking, Germany  
Gonzalo Lopez-Nicolas, Spain  
Cristian Mahulea, Spain  
Nikolay Manyakov, Belgium  
Antonio Muñoz, Spain  
Ana C. Murillo, Spain  
Andreas Neacrou, Greece  
Marco Montes de Oca, Belgium  
Sorin Olaru, France  
Karl Pauwels, Belgium  
Luis Puig, Spain  
Ana Respício, Portugal  
Pere Ridaó, Spain  
Kathrin Roberts, Germany  
Paulo Lopes dos Santos, Portugal  
Felix Sawo, Germany

Frederico Schaf, Brazil  
Oliver Schrempf, Germany  
Torsten Sievers, Germany  
Razvan Solea, Portugal  
Wolfgang Steiner, Austria  
Christian Stolle, Germany  
Alina Tarau, The Netherlands  
Rui Tavares, Portugal  
Paulo Trigo, Portugal  
Haralambos Valsamos, Greece  
José Luis Villarroel, Spain  
Yunhua Wang, USA  
Florian Weißel, Germany  
Jiann-Ming Wu, Taiwan  
Oliver Wulf, Germany  
Ali Zayed, Libya  
Yan Zhai, USA

### **Invited Speakers**

Dimitar Filev, The Ford Motor Company, USA  
Mark W. Spong, University of Illinois at Urbana-Champaign, USA  
Patrick Millot, Université de Valenciennes, France

# Contents

## Invited Papers

Toward Human-Machine Cooperation <i>Patrick Millot</i> .....	3
---	---

## Part I: Intelligent Control Systems and Optimization

Planning of Maintenance Operations for a Motorway Operator based upon Multicriteria Evaluations over a Finite Scale and Sensitivity Analyses <i>Céline Sanchez, Jacky Montmain, Marc Vinches and Brigitte Mahieu</i> .....	23
A Multiple Sensor Fault Detection Method based on Fuzzy Parametric Approach <i>Frederic Lafont, Nathalie Pessel and Jean-Francois Balmat</i> .....	37
The Shape of a Local Minimum and the Probability of its Detection in Random Search <i>Boris Kryzhanovsky, Vladimir Kryzhanovsky and Andrey Mikaelian</i> .....	51
Rapidly-exploring Sorted Random Tree: A Self Adaptive Random Motion Planning Algorithm <i>Nicolas Jouandeau</i> .....	63
Applying an Intensification Strategy on Vehicle Routing Problem <i>Etiene P. L. Simas and Arthur Tórgo Gómez</i> .....	75
Detection of Correct Moment to Model Update <i>Heli Koskimäki, Ilmari Juutilainen, Perttu Laurinen and Juha Röning</i> .....	87
Robust Optimizers for Nonlinear Programming in Approximate Dynamic Programming <i>Olivier Teytaud and Sylvain Gelly</i> .....	95

## Part II: Robotics and Automation

Improved Positional Accuracy of Robots with High Nonlinear Friction using a Modified Impulse Controller <i>Stephen van Duin, Christopher D. Cook, Zheng Li and Gursel Alici</i> .....	109
---	-----

An Estimation Process for Tire-Road Forces and Sideslip Angle for Automotive Safety Systems <i>Guillaume Baffet, Ali Charara, Daniel Lechner and Damien Thomas</i> .....	125
SMARTMOBILE and its Applications to Guaranteed Modeling and Simulation of Mechanical Systems <i>Ekaterina Auer and Wolfram Luther</i> .....	139
Path Planning for Cooperating Unmanned Vehicles over 3-D Terrain <i>Ioannis K. Nikolos and Nikos C. Tsourveloudis</i> .....	153
Tracking of Manoeuvring Visual Targets <i>C. Pérez, N. García, J. M. Sabater, J. M. Azorín and L. Gracia</i> .....	169
Motion Control of an Omnidirectional Mobile Robot <i>Xiang Li and Andreas Zell</i> .....	181
A Strategy for Exploration with a Multi-robot System <i>Jonathan A. Rogge and Dirk Aeyels</i> .....	195
Tracking of Constrained Submarine Robot Arms <i>Ernesto Olguín-Díaz and Vicente Parra-Vega</i> .....	207

**Part III: Signal Processing, Systems Modeling and Control**

Modelling Robot Dynamics with Masses and Pulleys <i>Leo J. Stocco and Matt J. Yedlin</i> .....	225
Stochastic Nonlinear Model Predictive Control based on Gaussian Mixture Approximations <i>Florian Weissel, Marco F. Huber and Uwe D. Hanebeck</i> .....	239
The Conjugate Gradient Partitioned Block Frequency-Domain for Multichannel Adaptive Filtering <i>Lino García Morales</i> .....	253
Guaranteed Characterization of Capture Basins of Nonlinear State-Space Systems <i>Nicolas Delanoue, Luc Jaulin, Laurent Hardouin and Mehdi Lhommeau</i> .....	265
In Situ Two-Thermocouple Sensor Characterisation using Cross-Relation Blind Deconvolution with Signal Conditioning for Improved Robustness <i>Peter Hung, Seán McLoone, George Irwin, Robert Kee and Colin Brown</i> .....	273



Dirac Mixture Approximation for Nonlinear Stochastic Filtering  
*Oliver C. Schrempf and Uwe D. Hanebeck* ..... 287

On the Geometry of Predictive Control with Nonlinear Constraints  
*Sorin Oлару, Didier Dumur and Simona Dobre* ..... 301

Author Index..... 315

# **Invited Papers**

# Toward Human-Machine Cooperation

Patrick Millot

Univ Lille Nord de France, UVHC, F- 59313 Valenciennes  
Laboratoire d'Automatique de Mécanique et d'Informatique Industrielles et Humaines LAMIH  
UMR CNRS 8530, France  
patrick.millot@univ-valenciennes.fr

**Abstract.** In human-machine systems, human activities are mainly oriented toward decision-making, encompassing monitoring and fault detection, fault anticipation, diagnosis and prognosis, and fault prevention and recovery. The objectives of this decision-making are related to human-machine system performance (production quantity and quality) as well as to overall system safety. In this context, human operators often play a double role: one negative in that they may perform unsafe or erroneous actions affecting the process, and one positive in that they are able to detect, prevent or recover an unsafe process behavior caused by another operator or by automated decision-makers. This pluridisciplinary study combines two approaches to human-machine systems: a human engineering approach, aimed at designing dedicated assistance tools for human operators and integrating them into human activities through human-machine cooperation, and an approach centered on cognitive psychology and ergonomics, which attempts to analyze the human activities in terms of the need for and use of such tools. This paper first focuses on parameters related to human-machine interaction, which have an influence on safety (e.g., degree of automation, system complexity, human complexity when dealing with normative and erroneous behaviors). The concept of cooperation is then introduced in response to safety concerns, and a framework for implementing cooperation is proposed. Examples in Air Traffic Control and in Telecommunication networks are used to illustrate our proposal.

**Keywords.** Human-machine systems, human modeling, task sharing, supervision, decision support systems, human-machine cooperation.

## 1 Introduction

In this field of research, the term, “machine”, refers not only to computers, but also to diverse control devices in complex dynamic situations, such as industrial processes or transportation networks. Human activities are mainly oriented toward decision-making, including monitoring and fault detection, fault anticipation, diagnosis and prognosis, and fault prevention and recovery. The objectives of this decision-making are related to human-machine system performance (production quantity and quality) as well as to overall system safety.

In this context human operators may have a double role: a negative role in that operators may perform unsafe or erroneous actions affecting the process, and a positive role in that they are able to detect, prevent or recover an unsafe process

behavior caused by another operator or by automated decision-makers. In both cases, the operators can be the victims of an erroneous action affecting the process.

This pluridisciplinary study combines two approaches to human-machine systems: a human engineering approach, aimed at designing dedicated assistance tools for human operators and integrating them into human activities through human-machine cooperation, and an approach centered on cognitive psychology and ergonomics, which attempts to analyze the human activities in terms of the need for and use of such tools.

The influence of these two approaches, one proposing technical solutions and the other, evaluating of the ergonomic acceptability of these solutions for the human operators, is apparent throughout this paper. First, the main parameters influencing human-machine system performance and safety are described. Then, human-machine cooperation is defined and a framework for implementing this cooperation is proposed.

## **2 Parameters Influencing Automated System Performance and Safety**

Human-machine system safety depends on 3 kinds of parameters: human parameters, technical parameters and parameters related to the interaction of the first two. This section focuses on some of the parameters in the last category.

### **2.1 Degrees of Automation**

The influence of the human role and the degree of human involvement on overall human-machine system performance (production, safety) has been studied since the early 1980s. Sheridan [1] defined the well-known degrees of automation and their consequences: at one extreme, in fully manual controlled systems, safety depends entirely on the human controller's reliability; at the other extreme, fully automated systems eliminate the human operator from the supervision and control loop, which can lead to a lack of vigilance and a loss of skill, preventing operators from assuming responsibility for the system and, consequently, making system safety almost totally dependent on technical reliability. Between the two extremes, there is an intermediate solution consisting of establishing supervisory control procedures that will allow task-sharing between the human operators and the automated control systems. In addition, dedicated assistance tools (e.g., DSS, or Decision Support Systems) can be introduced into the supervision and control loop in order to enhance the human ability to apply the right decision and/or to manage the wrong decisions.

### **2.2 Managing System Complexity**

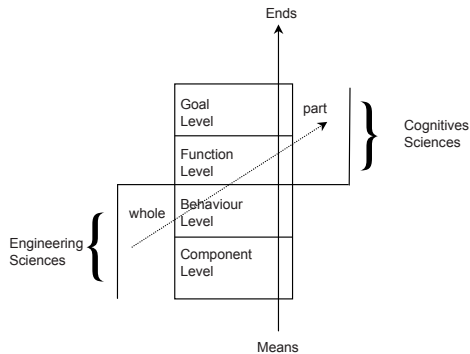
Technical failures and human errors generally increase with the size and/or the complexity of the system (i.e., the number of interconnections between the controlled variables and their degree of interconnection). For instance, large systems, such as

power plants or transport networks, can have several thousand interconnected variables that need to be supervised. In order to manage the resulting complexity, the human supervisor must be able to understand the system's behavior. This understanding can be facilitated by defining dedicated analysis methods, based on systemic approaches [2].

For instance, Multilevel Flow Modelling (MFM), developed by Lind [3], [4] is a very fruitful method for reducing the complexity, by breaking the system down hierarchically. The global system is decomposed according to two axes, the means-ends axis, and the whole-part axis (Fig. 1).

- The means-ends axis is composed of 4 model levels: the higher the level, the more global and abstract the model, and conversely, the lower the level, the more concrete and detailed the model. Each level corresponds to a different model: goal models (e.g., a financial view-point of the system), functional models (e.g., symbolic or graphic relationships) behavioral models (e.g., differential equations), and technical models (e.g., mechanical or electrical components. Each level's model is the means of attaining the higher-level model and is the ends (goal) of the lower-level model.

- The whole-part axis is linked to the decomposition imposed by the means-ends axis. At the highest level, the system is analyzed at a very global level (i.e., without details), making it possible to take the whole system into account. At the lowest level, the analysis is very detailed, providing a view of each component and looking at only one part of the system at a time.



**Fig.1.** Multilevel decomposition of a system by Lind [3], [4].

This modelling method seems promising, as it tends to associate cognitive sciences and engineering sciences in a complementary manner that allows complex systems to be modelled.

## 2.3 Understanding the Human Complexity

Modelling human problem solving in supervision tasks is another difficult objective for controlling Human-Machine systems. A lot of models have been proposed, among them the well-known Rasmussen's ladder [5], more recently revisited by Hoc [6] (Fig.2). This functional model groups many major functions: 1) abnormal event

detection, 2) situation assessment by perceiving information for identifying (diagnosis) and/or predicting (prognosis) the process state, and 3) decision-making by predicting the consequences of this state on the process goals, defining targets to be achieved, breaking targets down into tasks and procedures, and finally performing the resulting tasks and procedures to affect the process.

Since the early 1980s, a strong parallel has been drawn with artificial intelligence used to model and implement some of these functions in machines. Hoc has introduced revisions that provides more details about the cognitive mechanisms for assessing situations (e.g., hypothesis elaboration and testing) and about some temporal aspects of the process (i.e., diagnosis of the present state, prognosis of the future state, expected evolution of the process or projections regarding the appropriate instant for performing an action or for perceiving information).

This model has 3 behavioral levels, which enhance its effectiveness:

1. A skill-based behavior is adopted by a trained operator performs an action in an automatic manner when perceiving a specific signal.
2. A rule-based behavior is adopted by an expert operator, faced with a known problem, reuses a solution learned in the past.
3. A knowledge-based behavior is adopted when the operator is faced with an unknown problem and must find a new solution; in this special case, the operator needs to be supported either by other operators or by a decision- support system in order to understand the process situation and make the right decision.

The second and third levels involve cognitive mechanisms.

This model also served as J. Reason’s starting point [7] for understanding human error mechanisms and for providing barriers for preventing and/or managing these errors. For instance, an erroneous action can be the result of either the wrong application of a right decision or the right application of a wrong decision. The erroneous decision itself can either produce a wrong solution after a correct assessment of the situation or a sound solution based on an incorrect situation assessment, and so on.

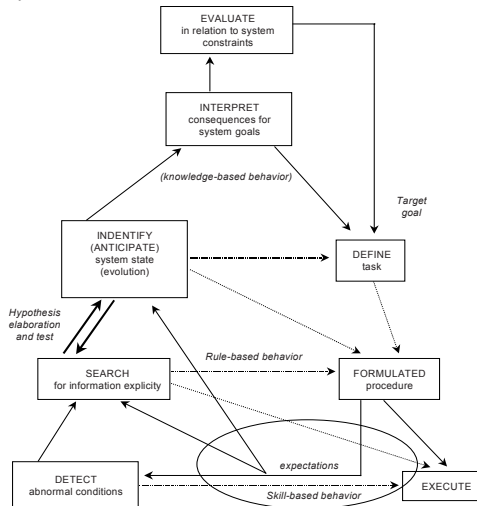


Fig. 2. Rasmussen’s step-ladder (revisited after Hoc [6]).

Reason divides human errors into two categories: non-intentional and intentional. These categories are further sub-divided into slips and lapses for non-intentional actions, and mistakes and violations for intentional decisions/actions, thus a total of 4 kinds of human errors. Violations differ from mistakes in that the decision-maker is conscious of violating the procedure, with either negative intent (e.g., sabotage) or positive intent (e.g., preventing an accident). Amalberti [8] tries to explain the production of certain violations through the need for human operators to reach a compromise solution for three joint, sometimes contradictory, objectives: performance standards, imposed either by the organization or by the individual operator; system and/or operator safety; and the cognitive and physiological costs of attaining the first two objectives (e.g., workload, stress). For Rasmussen [9], these 3 dimensions are limited and they limit the field of human action. An action that crosses this limit can lead to a loss of control, and subsequently, an incident or an accident.

Technical, organizational or procedural defenses can sometimes remedy faulty actions or decisions. Thus, several risk analysis methods have been proposed for detecting risky situations and providing such remedies [10], [11], [12], [13], [14]. Usually, risk management involves three complementary steps, which must be foreseen when designing the system:

-Prevention: the first step is to prevent risky behaviors. Unexpected behaviors should be foreseen when designing the system, and technical, human and organizational defenses should be implemented to avoid these behaviors (e.g., norms, procedures, maintenance policies, supervisory control).

-Correction: if prevention fails, the second step allows these unexpected behaviors to be detected (e.g., alarm detection system in a power plant) and corrected (e.g., fast train brakes).

-Recovery: if the corrective action fails, an accident may occur. The third step attempts to deal with the consequences of a failed corrective action, by intervening to minimize the negative consequences of this accident (e.g., emergency care on the road).

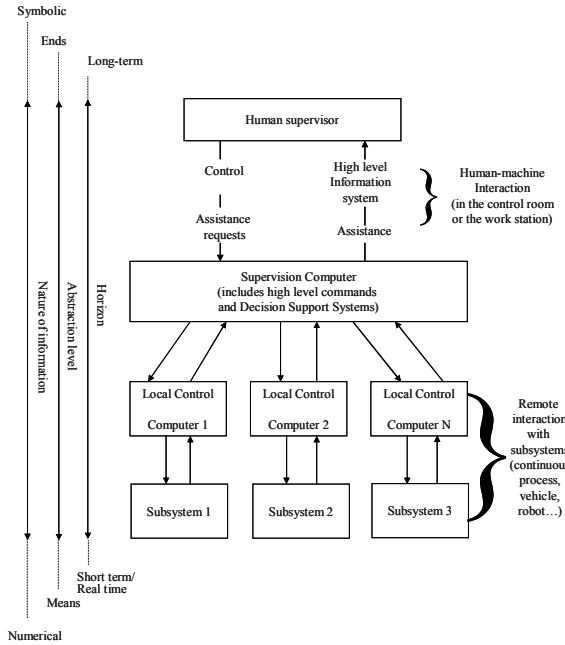
These three steps provide prevention, correction or recovery tasks that can be performed, some by the human operators and some by the machine. The question is then: "How should the tasks be shared between the human and the machine?"

### 3 Criterion for Operator-machine Task-sharing

#### 3.1 Identifying the Tasks to be Shared

As mentioned above, the task allocation determines the system's degree of automation. Sheridan [15] has proposed a hierarchical organization in 4 levels (Fig. 3):

- the process to be controlled or supervised, decomposed into subsystems,
- the local control units of these subsystems,
- the coordination of the local control units, including DSS, and
- the supervision of the human team and the automated device.



**Fig. 3.** Supervisory Control (adapted from Sheridan [15] and completed by Millot [16]).

Millot [16] added three scales to this structure:

- a scale related to the nature of the information, with the precise numerical information towards the bottom of the scale and the symbolic and global information towards the top,
- a scale related to the level of abstraction, with the means towards the bottom and the objectives towards the top (similar to Lind’s hierarchy),
- a scale related to the temporal horizons, with the activities to be performed in real time (e.g., the subsystem control tasks) towards the bottom and the long-term activities (e.g., planning or strategic decision-making) towards the top.

Sorting the tasks according to these 3 scales allows the nature of the task and the expected task performance to be defined at each level. Human abilities are best suited to processing symbolic information and planning and anticipating decisions about global objectives rather than specific means, and this on a middle or long-term horizon. For this reason, activities towards the bottom of the scale risk are not well suited to human capabilities and thus can result in human errors.

From these observations, a method for defining the roles of human operators and allocating their tasks can be deduced (for further details see Millot [16]):

- First, the technical constraints (e.g., dynamics, safety) with respect to the different predictable system operation modes need to be extracted. Functional or dysfunctional system analysis methods proposed by Fadier [17] or Villemeur [18] can be used to deduce (or induce) troubleshooting tasks needed to deal with the system.
- Then, these troubleshooting tasks can then be allocated to the human and the machine, according to their respective capabilities. To do so, the tasks must be



specified in terms of their objectives, acceptable means (e.g., sensors, actuators) and functions. (Section 3.2 examines this task-sharing process in more detail.)

- At this point, it is necessary to implement the automated processors for managing future automated tasks and the human-machine interfaces that will facilitate future human tasks.
- Finally, the entire system must be evaluated in terms of technical and ergonomic criteria.

### 3.2 Sharing the Tasks between Human and Machine

Human-machine task-sharing decisions are made according to two criteria: **technical feasibility** and **ergonomic feasibility**.

- The **Technical Feasibility** criterion is used to divide the initial task set into two classes (Fig.4):

- . TA tasks are technically able to be performed automatically,

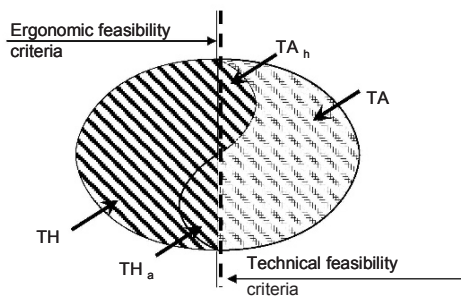
- . TH tasks cannot be performed automatically due to lack of information or to technical or even theoretical reasons, and thus must be allocated to human operators.

- The **Ergonomic Feasibility** criterion is applied to both subsets, TA and TH, to evaluate the human tasks in terms of global system safety and security:

- . in the subset TA, some automatable tasks  $TA_h$  can also be performed by humans, and allocating them to the human operators can allow these operators to better supervise and understand the global system and the automated devices.  $TA_h$  is thus the set of the **shareable tasks** used in a form of human-machine cooperation (i.e., the dynamic task allocation presented below).

- . in the subset TH, some subtasks  $TH_a$  are very complex, or their complexity is increased by a very short response time. Humans performing such tasks could be aided by a Decision Support System or a Control Support System. The subset  $TH_a$  thus can be the basis of another form of human-machine cooperation.

The ergonomic feasibility criterion is based on human operator models that define the possible human resources, as well as the intrinsic limits of the operators (perceptual and/or physical) when performing the related actions. The humans' cognitive resources depend on the context, and their physical resources can be determined through ergonomic guidelines [19], [7], [6], [8].



Legend: TA: automatable tasks; TH: tasks that cannot be automated and must be performed by humans.  $TA_h$ : tasks that can be performed by both humans and machines.  $TH_a$ : tasks that cannot be performed by machines or humans working alone.

**Fig. 4.** Sharing tasks between human and machine.

## 4 Human-Machine Cooperation as a Method for Preserving Human-Machine Safety

### 4.1 Defining an Agent

Decision Support Systems (DSS) provide assistance that makes Human Operator tasks easier and help prevent faulty actions. Both the DSS and the Human Operator are called agents. Agents (either human or machine) can be modelled according to 3 classes of capabilities — Know-How, Know-How-to Cooperate, and Need-to-Cooperate.

- 1) **Know-How** (KH) is applied to solve problems and perform tasks autonomously, while acquiring problem solving capabilities (e.g., sources of knowledge, processing abilities) and communicating with the environment and other agents through sensors and control devices.
- 2) **Know-How-to Cooperate** (KHC) is a class of specific capabilities that is needed for Managing Interferences between goals (MI) and for facilitating other agents' goals (FG) with respect to the definition of cooperation given in the next section [20].
- 3) **Need-to-Cooperate** (NC) is a new class combining [21]:
  - the **Adequacy** of the agents' personal KH (i.e., knowledge and processing abilities) in terms of the task constraints.
  - the **Ability** to perform the task (the human agents' workload (WL) produced by the task, perceptual abilities, and control abilities)
  - the **Motivation-to-Cooperate** of the agents (motivation to achieve the task, self-confidence, trust [22], confidence in the cooperation [23]).

In a multi-disciplinary approach, drawing on research in cognitive psychology and human engineering, we try to exploit these basic concepts and highlight the links between them, in order to propose a method for designing cooperative human-machine systems.

### 4.2 Defining Know-How-to-Cooperate

In the field of cognitive psychology, Hoc [6] and Millot & Hoc [20] have proposed the following definition: “two agents are cooperating if 1) each one strives towards goals and can interfere with the other, and 2) each agent tries to detect and process such interference to make the other's activities easier”.

From this definition, two classes of cooperative activities can be derived and combined, they constitute know-how-to-cooperate (KHC) as defined by Millot [24], [25]:

- The first activity, Managing Interference (**MI**), requires the ability to detect and manage interferences between goals. Such interferences can be seen as positive (e.g., common goal or sub-goal) or negative (e.g., conflicts between goals or sub-goals or about common shared resources).
- The second activity, Facilitating Goals (**FG**), requires the ability to make it easier for other agents' to achieve their goals.

MI requires more **coordination** abilities, while FG involves a more benevolent kind of agent behavior. Before specifying the exact abilities required for MI and FG, the organizational aspects of the cooperation must first be considered.

### 4.3 Structures for Cooperation

In an organization, agents play roles and thus perform tasks combining the different activities needed to acquire and process information and make decisions. The decisions may, or may not, result in actions. Defining the organization has often been seen as a way to prevent or resolve decisional conflicts between agents, especially in human engineering in which agents may be human or artificial DSS. This aspect is also studied under the name of Distributed Artificial Intelligence. In terms of purely structural organization, two generic structures exist: vertical (hierarchical) and horizontal (heterarchical) [26], [27].

In the vertical structure (Fig.5), agent AG1 is at the upper level of the hierarchy and is responsible of all the decisions. If necessary, it can call upon agent AG2, which can give advice.

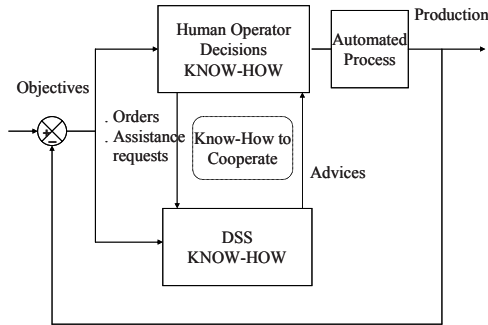


Fig. 5. Vertical Structure for human-machine cooperation.

In the horizontal structure (Fig. 6), both agents are on the same hierarchical level and can behave independently if their respective tasks are independent. Otherwise, they must manage the interferences between their goals using their MI and FG abilities.

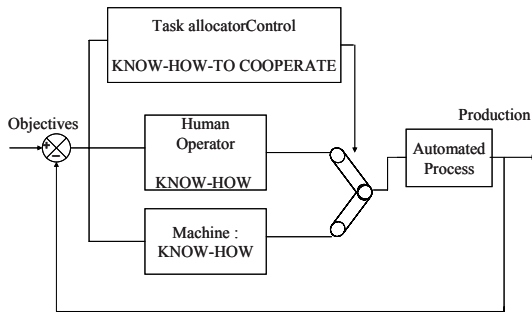


Fig. 6. Horizontal structure for human-machine cooperation.

Several combinations of both structures are also possible, by decomposing the task into several abstraction levels [28] and assigning a dedicated structure to each level. An example will be given below. However, evaluating these structures with realistic experimental platforms showed that choosing a cooperative structure is not sufficient to allow real cooperation between human and machine or, more generally, between two agents [20]. It is necessary to introduce a complementary framework for describing the nature of the cooperative activities and for specifying the agents' KHC.

## 4.4 Cooperative Forms and Know-How-to-Cooperate

Let us consider an agent AG<sub>x</sub> that acquires know-how Kh<sub>x</sub> and know-how-to-cooperate KHC<sub>x</sub>, and is within a structure. The objective is to specify KHC<sub>x</sub> using MI<sub>x</sub> and FG<sub>x</sub> in the different cooperative situations that can be encountered (or built). This can be done by adapting and using the generic typology of cooperative forms proposed by Schmidt [29]: augmentative, debative, integrative.

### 4.4.1 Augmentative

Cooperation is **augmentative** when agents have **similar know-how** but multiple agents are needed to perform a task too demanding for only one agent. Thus, task T must be divided into **similar subtasks ST<sub>i</sub>**.

Interference between the agent activities can result when common resources must be shared or when agents have conflicting goals or sub-goals stemming from their individual ST<sub>i</sub>. Thus, KHC must allow 1) the decomposition of task T into **independent** ST<sub>i</sub> before the task is performed in order to prevent these conflicts, 2) the management of residual conflicts (e.g., about common resources) during ST<sub>i</sub> execution, and 3) the recomposition of the results afterwards if the ST<sub>i</sub> did not result in an action or the rebuilding of the task context else. These activities can be performed by a third agent called the coordinator, or by either AG<sub>x</sub> or AG<sub>y</sub>, which will then play the double role of coordinator and actor.

The coordinator's KH includes the abilities needed to acquire the task context, to build a global plan for the task and to decompose it into ST<sub>i</sub> (sub-plans). The coordinator's KHC includes the abilities needed to acquire other agents' KH (or to infer them from a model) and other agents' workloads (WL) and/or resources for sub-task allocation, to control and recompose the results or the contexts after each ST<sub>i</sub> has been performed, and to manage conflicts about shared common resources. All these KHC abilities are related to MI. The other non-coordinator agents' KHC abilities are related to FG and consist of answering the coordinator's requests.

### 4.4.2 Debative

Cooperation is **debative** when agents have **similar know-how** and are faced with a **single task T** that is not divided into ST<sub>i</sub>. Each agent solves the task and then debates the results (or the partial results) with the other agents. Conflicts can arise and the KHC must allow these conflicts to be solved through explanations based on previous partial results along the problem-solving pathway and on a common frame of reference, for instance [20].

Before task execution, each agent's KH is related to its ability to acquire the task context and build a plan (i.e., establish a goal, sub-goals & means). Each agent's KHC consists of acquiring the other agents' KH either by inferring the other agents' KH models or by asking the other agents for their KH. These inferences and/or requests are part of MI capabilities. The other agents' responses to these requests constitute FG capabilities.

After task execution (complete or partial), each agent transmits its own results to the others, receives results from the other agents and compares them to its own results. In addition to MI (asking for results from others) and FG (transmitting its own results) capabilities, this process requires that agents have specific competencies for understanding the others' results, comparing them to its own results, and deciding whether or not to agree with the others' results. These competences are all included in MI.

In case of conflict, each agent must be able to ask for explanations (e.g., the other agent's view of the task context, its partial results, its goal and/or sub-goals) in order to compare these explanations with its own view-point and to decide whether or not the conflict should continue. In addition, each agent must be able to acknowledge its own errors and learn the lesson needed to avoid such errors in the future. This last ability can have important consequences on agent KH.

#### 4.4.3 Integrative

Cooperation is **integrative** when agents have **different and complementary** know-how and the task T can be **divided into complementary sub-tasks ST<sub>i</sub>** related to each KH. As in the augmentative form of cooperation, a third agent can play the role of coordinator; however, this role could also be played by one of the agents, AG<sub>x</sub> or AG<sub>y</sub>. The coordinator's KHC must allow 1) the elaboration of a common plan (goal, means) and its decomposition into complementary sub-plans (ST<sub>i</sub>, sub-goals) related to each of the respective agents' KH, 2) the control the partial results or the evolving context throughout the agents' execution of the ST<sub>i</sub>, and 3) the recomposition of the results afterwards if the results of the ST<sub>i</sub> were not an action, or the rebuilding of the task context else. The integrative form of cooperation is similar to the augmentative form, except that during ST<sub>i</sub> execution, the possibility of conflictual interactions between the different ST<sub>i</sub> requires that the coordinator be capable of checking each partial result and of ordering corrections, if needed.

A more general and complex case can be imagined, in which both agents, AG<sub>x</sub> and AG<sub>y</sub>, must cooperate in order to build a shared common plan. This case is often studied in the field of Distributed Artificial Intelligence. In such a situation, each agent plays the role of coordinator, first seeking to establish a common frame of reference with respect to the task context and each agent's KH [20] and then working to develop its own comprehensive common plan and comparing it with those of the other agents in **debative** cooperation. Examples of this case can be found in multidisciplinary studies of Human-Human cooperation, for instance.

#### 4.4.4 Human-human Cooperation Examples

As mentioned above, these 3 forms already exist in human-human organizations and are sometimes naturally combined. An example of the augmentative form of

cooperation can be observed in banks, when the line in front of a window is too long, a second window is opened, thus cutting the line in half and reducing the first teller's workload. An example of the debative form is found in the mutual control established between the flying pilot and the co-pilot in the plane cockpit. An example of the integrative form can be seen in the coordination of the different tasks required to build a house. The innovation lies in implementing these forms in human-machine systems.

## 5 Cooperative Forms and Structures in Human-Machine Cooperation

This section presents an analysis of the kind of structure that should be chosen to support the different cooperative forms; the recommended forms are illustrated with examples.

### 5.1 Augmentative Form and Structure, the Example of Air Traffic Control

In this example, both agents have similar KH, and each performs a subtask ST<sub>i</sub> resulting from the division of task T into similar subtasks. In order to prevent conflicts between the agents, the coordinator must decompose T into independent subtasks.

In Air Traffic Control (ATC), the objectives consist of monitoring and controlling the traffic in such a way that the aircraft cross the air space with a maximum level of safety. The air space is divided into geographical sectors, each of them controlled by two controllers. The first one is a tactical controller, called the "radar controller" (RC) who supervises the traffic using a radar screen and dialogues with the aircraft pilots. The supervision task entails detecting possible traffic conflicts between planes that may violate separation norms, resulting in a collision, and then solving them. Conflict resolution usually involves asking to one pilot to modify his/her flight level, heading, or speed.

The second controller is a strategic controller, called the "planning controller" (PC). PC coordinates the traffic in his/her own sector with the traffic in other sectors in order to avoid irreconcilable conflicts on the sector's borders. They are also supposed to anticipate traffic density and regulate the workload of the RC. In addition, in traffic overload conditions, PC assists RC by taking in charge some tactical tasks. To support the RC, a dedicated DSS called SAINTEX has been developed. In this system, each agent (i.e., the RC and SAINTEX) was allowed to perform actions affecting the traffic, and the tasks were dynamically distributed between these two agents based on performance and workload criteria.

To accomplish this dynamic task allocation, a task allocator control system was introduced at the strategic level of the organization [30], which can be:

- a dedicated artificial decisional system with the ability to assess human workload and performance, in which case the **dynamic task allocation** is called **implicit**, or

- the human operator, who plays a second role dealing with strategic and organizational tasks, in which case the **dynamic task allocation** is called **explicit**.

These two task allocation modes were implemented on a realistic Air Traffic Control (ATC) simulator and evaluated by professional Air-Traffic Controllers.

A series of experiments implemented both implicit and explicit dynamic task allocation between the radar controller and SAINTEX. The task allocation depended on the know-how (KH) of the two decision-makers. The SAINTEX KH was limited to simple aircraft conflicts (i.e., between only two planes). The RC's know-how was only limited by the workload. The experiments showed better performance in terms of overall safety and fuel consumption of the traffic, and a better human regulation of the workload in the implicit allocation mode than in the explicit one. However, the responses to the questionnaires showed that the professional Air Traffic Controllers **would not easily accept implicit allocation** in real situations because (a) the different tasks were not completely independent, and (b) they had no control over the tasks assigned to SAINTEX, but retained total responsibility for all tasks.

Thus, it seems that if AGx and AGy are both provided with all the KH and KHC capabilities of a coordinator, a purely horizontal structure like the ones used in Distributed Artificial Intelligence must be envisaged. However, if only one agent, for instance AGx, is assigned the capabilities needed to be a coordinator, the result is a *de facto* hierarchy in which AGx manages the cooperation. AGy will then have FG capabilities and become an assistant in the cooperation. This situation is quite realistic in Human-Machine Cooperation, and the dynamic task allocation aiming for this form of cooperation can be analyzed from this perspective. In the experiment involving only RC and SAINTEX, there was an asymmetry between the KHC of both agents, creating a *de facto* hierarchy in which the RC held the higher position. In the explicit mode, this hierarchy was respected, but in implicit mode, it was reversed, which could explain the RC's refusal of this type of organization. In addition, the sub-tasks were not really independent since solving some traffic conflicts increased the risk of creating new ones. Thus, the cooperative form was not purely augmentative; a purely augmentative form would have required SAINTEX to have other KHC related to the other cooperative forms.

## 5.2 Debative Form and Structure

In this example, both agents have similar KH and are faced with a single task T that is not (or cannot be) divided into sub-tasks. After each agent had performed the task, they compare the results (or the partial results), and when there is a conflict, they debate.

If both agents are given all the KH and KHC abilities, a purely horizontal structure can be imagined. The ability to recognize and acknowledge errors may then depend on trust and self-confidence [22]. On the other hand, giving only one agent full KHC results in a *de facto* hierarchy; if such a hierarchical structure is chosen, the conflict resolution process can be aided (or perturbed) by the hierarchy. This situation is realistic in human-machine cooperation, because the machine capabilities can be reduced to FG capabilities. In this case, the designer of the machine must have simulated the human user's conflict resolution pathway so as to allow the machine to help the human to cooperate with it.

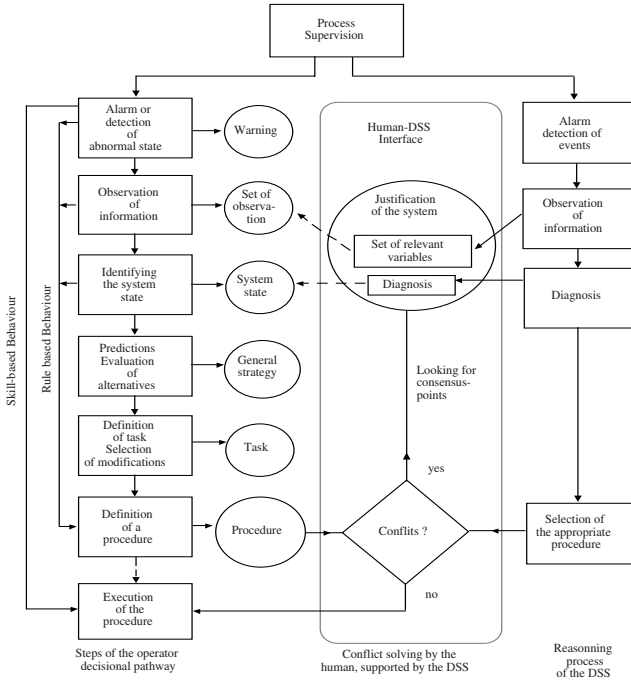


Fig. 7. Synthesis of the decisional conflict resolution pathway in a debative form.

In a study aiming to integrate a DSS into the supervision loop of a continuous production process, a justification graphic interface was designed for that purpose by Taborin et al. [26]. The conflict resolution process was simulated by running both decisional pathways (human and DSS) in parallel, with both paths being inspired by Rasmussen [5].

Conflict resolution consists of looking for “consensus points” in the common deductions of each decision-maker (Fig. 7). For instance, in a conflict resulting from each decision-maker proposing different procedures, the first consensus point would be the previous common deduction, or in other words, each decision-maker’s diagnosis. A second consensus point would be the set of variables used by each decision-maker to make this diagnosis.

### 5.3 Integrative Form and Structure, the Example of Diagnosis in a Telecommunication Network

In this example, both agents have different and complementary KH and each performs a subtask  $St_i$  resulting from the division of  $T$  into complementary subtasks. The task can be decomposed and managed by the coordinator, which can be a third agent or one of the two original agents, all with all KHC capabilities.

As for the other cooperative forms, a horizontal structure, in which each agent has all KHC capabilities, can be imagined. This is generally the case in Human-Human Cooperation, for instance between the pilot and the co-pilot in the plane cockpit.



When the KHC capabilities of one agent are only partial, as is usually the case in Human-Machine Cooperation, the structure is a *de facto* hierarchy, either for reasons of competency or legal responsibilities, or both as is the case in ATC. Thus, the designer must respect this hierarchical organization when creating the structure.

Let us consider the form of cooperation found in the diagnosis task, in which two main tasks are essential for quickly focusing on the failures affecting the system:

- The first task is to interpret the data collected on the system and to generate a set of failure hypotheses. The hypotheses are then crossed to determine a minimal failure set that explains the effects observed.

- The second task is to check the consistency of the hypotheses at each step in the reasoning, according to the system model.

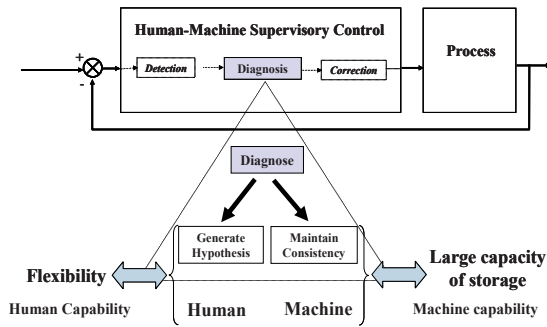


Fig. 8. Task allocation in human-machine diagnosis.

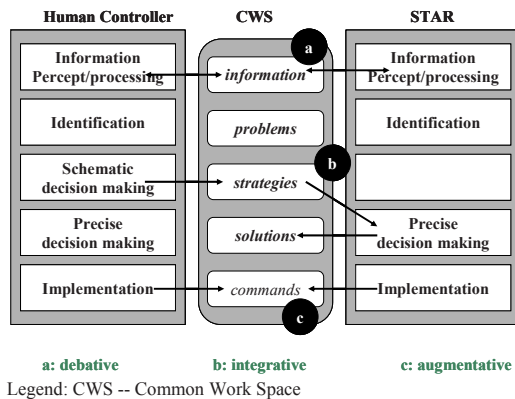
The first task requires a flexible global view of the system in order to quickly generate consistent failure hypotheses. The necessary Know-How resembles human abilities and thus is allocated to the human operator. The second task requires calculating power in order to check the consistency of the hypotheses and to consider multiple alternatives rapidly. The KH needed is best suited to machine abilities and thus is allocated to the machine (Fig. 8). After the tasks have been allocated, the main problem remaining is to define the means for coordinating both decision-makers' activities during the diagnosis process because, in fact, the partial results must be aggregated. As the tasks are shared, the decision-makers must exchange data and interpret them. Furthermore, both decision-makers must share knowledge about the process (e.g., external data); a shared workspace is also needed for coordinating the reasoning processes of the human operator and the machine [31]. The shared knowledge can be represented as a causal network of links between symptoms and the causes of failure.

An example of a diagnosis task was studied in domestic phone network. Customers having difficulties with their phone call a "hotline" service, and an operator must make a diagnosis. The problem can come from the hardware, or a customer mistake, or from a combination of the hardware and the line itself. A DSS was built to assist the operators of such hotlines and was evaluated in well-defined experimental conditions: in the experimental protocol, the network could have 49 possible phone system failures; these failures were linked to 150 symptoms. The result was a causal network with 500 possible links. In less than 3 minutes, hotline operators must find one diagnosis among the possible 49, using knowledge of the

actual symptoms among 150 possible ones. Operators gather information about the symptoms through direct dialogue with the customer and through test devices. The experiments showed that using integrative cooperation with the DSS, the average number of good diagnoses increased from 64% to 82% [32].

## 5.4 Complex Real Case, an Example in Air Traffic Control

Generally, pure cooperative forms do not exist in the real world; most often, a combination of the three forms is encountered. This is the case in Air Traffic Control (ATC). The AMANDA (Automation and MAN-machine Delegation of Action) project has studied a new version of cooperation between Human Controllers and a new tool called STAR in the ATC context. The objective of the project was to build a common frame of reference, called Common Work Space (CWS), using the support system STAR [31]. STAR is able to take controller strategies into account in order to calculate precise solutions and then transmits the corresponding command to the plane pilot. The common frame of reference of air traffic controllers was first identified experimentally by coding the cognitive activities of air traffic controllers [33]. The common workspace (CWS) resulting from this common frame of reference was implemented on the graphic interface of the AMANDA platform [21]. The CWS plays a role similar to a black-board, displaying the problems to be solved cooperatively. As each agent brings pieces of the solution, the CWS displays the evolution of the solution in real time.



**Fig. 9.** Combined cooperative forms in ATC.

The cooperation between STAR and the human controller can take the 3 forms (Fig.9):

a) debative, b) integrative, c) augmentative. The experimental evaluation shows that this cooperative organization allows the controllers to better anticipate air traffic conflicts, thus increasing the safety level. In addition, the common workspace seems to provide a good representation of air traffic conflicts and thus is a good tool for conflict resolution. Furthermore, this organization provides for better task sharing between the two types of controllers (RC and PC), which results in a better regulated workload [21].

## 6 Conclusions

This paper reviews the objectives and the methods used in human engineering to enhance the safety of automated systems, focusing on the parameters related to human-machine interaction—degree of automation, system complexity, the richness and complexity of the human component— among the different classes of parameters that influence safety. One solution approach is to implement cooperation between human and DSS. This paper proposes a framework for integrating human-machine cooperation. Clearly, in order to implement human-machine cooperation, it is necessary to cope not only with the KH of the different agents (human or machine), but also with their respective KHC. Three cooperation forms have been introduced for describing the activities composing the KHC of each agent. These activities can be gathered in too groups: MI corresponding to a coordination activity and FG corresponding to a benevolent behavior for facilitating the other agent's goals. In addition, the appropriate cooperative structure must be chosen. Several examples were presented, regarding each form of cooperation and related to different application fields: Air Traffic Control, production process supervision and Telecommunication networks. In the case of human-machine cooperation the ability of a machine to achieve coordination tasks is discussed in each of these examples.

## References

1. Sheridan T.B., 84, "Supervisory Control of Remote Manipulators, Vehicles and Dynamic Processes: Experiments in Command and Display Aiding", *Advances in Man-machines Systems Researches*, vol. 1(1984)
2. Lemoigne J.L., 84 (reedited 94), "La théorie du système général, théorie de la modélisation" PUF, Paris (1984)
3. Lind M., 90, "Representing Goals and Functions of Complex Systems: an Introduction to Multilevel Flow Modelling", Technical Report 90-D-381 TU Denmark (1990).
4. Lind M. 03, "Making sense of the abstraction hierarchy in the power plant domain", in *Cognition Technology and Work*, vol 5, n°2, (2003) 67-81
5. Rasmussen J. 83, *Skills, Rules and Knowledge: signals, signs and symbols and other distinctions in human performance models: IEEE SMC n°3* (1983)
6. Hoc J.M.: "Supervision et contrôle de processus, la cognition en situation dynamique". Presses Universitaires de Grenoble (1996)
7. Reason J.: "Human error" Cambridge University Press (1990) (Version française traduite par J.M. Hoc, *L'erreur humaine* PUF (1993))
8. Amalberti R. :, "La conduite des systèmes à risques" PUF (1996)
9. Rasmussen J.: "Risk management in a dynamic society: a modelling problem", *Safety Sciences*, 27, 2/3, (1997) 183-213
10. Fadier E., Actigny B. et al.: "Etat de l'art dans le domaine de la fiabilité humaine", ouvrage collectif sous la direction de E. Fadier, Octarès, Paris (1994)
11. Hollnagel E.: "Cognitive Reliability and Errors Analysis Method", CREAM, Elsevier, Amsterdam (1999)
12. Vanderhaegen F.: "APRECIH: a human unreliability analysis method-application to railway system", *Control Engineering Practice*, 7 (1999) 1395-1403
13. Polet P., Vanderhaegen F., Wieringa P.A.: "Theory of Safety-related violations of a System Barriers", *Cognition Technology and Work*, 4 (2002) 171-179

14. Van der Vlugt M., Wieringa P.A.: "Searching for ways to recover from fixation: proposal for a different view-point", Cognitive Science Approach for Process Control CSAPC'03, Amsterdam, September (2003)
15. Sheridan T.: «Forty-Five Years of Man-Machine Systems: History and Trends», 2nd IFAC Conference Analysis, Design and Evaluation of Man-Machine Systems, Varese, september (1985)
16. Millot P.: «Systèmes Homme-Machine et Automatique», Journées Doctorales d'Automatique JDA'99, Conférence Plénière, Nancy, septembre (1999)
17. Fadier E.: «Fiabilité humaine : Méthodes d'analyse et domaines d'application», In J. Leplat et G. de Terssac éditeurs; Les Facteurs humains de la fiabilité dans les systèmes complexes, Edition Octarés, Marseille (1990)
18. Villemeur A.: «Sûreté de fonctionnement des systèmes industriels : fiabilité, facteur humain, informatisation», Eyrolles, Paris (1988)
19. Reason J.: «Intentions, errors and machines: a cognitive science perspective», Aspects of consciousness and awareness, Bielefeld, W. Germany, december (1986)
20. Millot P., Hoc J.M.: "Human-Machine Cooperation: Metaphor or possible reality?" European Conference on Cognitive Sciences, ECCS'97, Manchester UK, April (1997)
21. Millot P., Debernard S.: "An Attempt for conceptual framework for Human-Machine Cooperation", IFAC/IFIP/IFORS/IEA Conference Analysis Design and Evaluation of Human-machine Systems Seoul Korea, September (2007)
22. Moray N., Lee, Muir.: "Trust and Human Intervention in automated Systems", in Hoc, Cacciabue, Hollnagel editors : Expertise and Technology cognition and Human Computer Interaction. Lawrence Erlbaum Publ. (1995)
23. Rajaonah B., Tricot N., Anceaux F., Millot P.: Role of intervening variables in driver-ACC cooperation, International Journal of Human Computer Studies (2006)
24. Millot P.: "Concepts and limits for Human-Machine Cooperation", IEEE SMC CESA'98 Conference, Hammamet, Tunisia, April (1998)
25. Millot P., Lemoine M.P.: "An attempt for generic concepts Toward Human-Machine Cooperation", IEEE SMC, San Diego, USA, October (1998)
26. Millot P., Taborin V., Kamoun A.: «Two approaches for man-computer Cooperation in supervisory Tasks», 4th IFAC Congress on "Analysis Design and Evaluation of man-machine Systems", XiAn China, September (1989)
27. Grislin-Le Strugeon E., Millot P.: «Specifying artificial cooperative agents through a synthesis of several models of cooperation», 7<sup>th</sup> European Conference on Cognitive Science Approach to Process Control CSAPC'99, p. 73-78, Villeneuve d'Ascq, september (1999)
28. Rasmussen J.: "Modelling distributed decision making", in Rasmussen J., Brehmer B., and Leplat J. (Eds), Distributed decision-making: cognitive models for cooperative work pp111-142, John Willey and Sons, Chichester UK(1991)
29. Schmidt K.:« Cooperative Work: a conceptual framework », In J. Rasmussen, B. Brehmer, and J. Leplat (Eds), Distributed decision making: Cognitive models for cooperative work (1991) 75-110
30. Vanderhaegen F., Crévits I., Debernard S., Millot P.: «Human-Machine cooperation: Toward an Activity Regulation Assistance for Different Air Traffic Control Levels », International Journal of Human Computer Interactive, 6(1) (1994) 65-104
31. Pacaux-Lemoine M.P., Debernard S.: "Common work space for Human-Machine Cooperation in Air Traffic Control", Control Engineering and Practice, 10 (2002) 571-576
32. Jouglet D., Millot P.: "Performance improvement of Technical diagnosis provided by human-machine cooperation", IFAC Human-Machine Systems: Analysis Design and Evaluation of Human-Machine Systems, Kassel, Germany, September (2001)
33. Guiost B, Debernard S., Millot P.: "Definition of a Common Work Space". In 10th International Conference of Human-Computer Interaction, Crete, Greece, January (2003) 442-446

# **PART I**

## **Intelligent Control Systems and Optimization**

# Planning of Maintenance Operations for a Motorway Operator based upon Multicriteria Evaluations over a Finite Scale and Sensitivity Analyses

Céline Sanchez<sup>1</sup>, Jacky Montmain<sup>2</sup>, Marc Vinches<sup>2</sup> and Brigitte Mahieu<sup>1</sup>

<sup>1</sup>Service Structure Viabilité Sécurité, Société des Autoroutes Estérel Côtes d'Azur Provence  
Alpes, avenue de Cannes, 06211 Mandelieu Cedex, France  
{cesanchez, bmahieu}@escota.net

<sup>2</sup>Ecole des Mines d'Alès, 6 avenue de Clavières, 30319 Alès Cedex, France  
{jacky.montmain, marc.vinches}@ema.fr

**Abstract.** The Escota Company aims at the formalization and improvement of the decisional process for preventive maintenance in a multi criteria (MC) environment. According to available pieces of knowledge on the infrastructure condition, operations are to be evaluated with regards to (w.r.t.) technical but also to conformity, security and financial criteria. This MC evaluation is modelled as the aggregation of partial scores attributed to an operation w.r.t. a given set of  $n$  criteria. The scores are expressed over a finite scale which can cause some troubles when no attention is paid to the aggregation procedure. This paper deals with the consistency of the evaluation process, where scores are expressed as labels by Escota's experts, whereas the aggregation model is supposed to deal with numerical values and cardinal scales. We try to analyse this curious but common apparent paradox in MC evaluation when engineering contexts are concerned. A robustness study of the evaluation process concludes this paper.

**Keywords.** Multi-criteria decision-making, Multi-criteria aggregation, Finite scale, Decision support system, Motorway infrastructure.

## 1 Escota Decision Process

### 1.1 Context

The Escota Company, founded in 1956, is the leading operator of toll motorways in France. Due to its integration into the Provence-Alpes-Côte d'Azur region, Escota is committed, as every motorway operator, to a sustainable development approach, including the social, economic and environmental aspects of its activities. Every year, specific initiatives are undertaken, or repeated, to include the motorway network in a sustainable development approach. Within this scope, the Escota Company aims at the formalization and improvement of the decisional process for preventive maintenance and property management with the desire to show transparency on decisions relative to property management, personal accountability and justification of decision-making logic in a multi actors and multi criteria (MC) environment [6], [7]. These decisions concern upkeep, improvement and upgrading operations,

involving technical, conformity, security or financial criteria. The operations are related to operating domains such as constructive works, carriageways, vertical road signs and carriageway markings, buildings, prevention of fire risks, open spaces... Managing such a complex infrastructure necessitates a dynamic Information Processing System (IPS) to facilitate the way decision-makers use their reasoning capabilities through adequate information processing procedure.

## 1.2 Valuation of the Infrastructure Condition

Periodic inspections are performed to detect and measure, as early as possible, any malfunction symptoms affecting an element of the infrastructure (EI). The expert in charge of an operating domain then analyses the technical diagnosis relative to the EI. He evaluates the situation seriousness in terms of technical risk analyses. This evaluation relies on a specific set of  $n$  criteria relative to his domain. An aggregation with a weighted arithmetic mean (WAM) is then performed to assess a global degree of emergency to the corresponding maintenance operation. This evaluation is then submitted to the official in charge of the operating network. This latter coordinates the experts' needs and demands for operation planning purposes.

This paper deals more particularly with the MC evaluation process by the expert of an operating domain, i.e. the affectation of an emergency degree to an operation. There exist several methods to identify and perform aggregation process with a WAM. The Analytic Hierarchical Process, AHP, is probably the most famous one in industry [1]. However, because it explicitly guarantees the consistency between the commensurable scales it aggregates and the WAM operator it identifies, the Measuring Attractiveness by a Categorical Based Evaluation Technique method, MACBETH, has got recent successes [2], [3]. In our application, MACBETH is first used to build the valuation scale associated to each emergency criterion of a domain. It is then applied to determine the WAM parameters.

Furthermore, the way experts give their assessment in natural language raises another problem [4]. These labels are commonly converted into numerical values to perform the aggregation process. No particular attention is generally paid to this "translation". However the consequences over the aggregation results are damageable. In civil engineering, the culture of numbers is strongly developed. People commonly manipulate symbolic labels but may convert them into more or less arbitrary numerical values when necessary without further care. This cultural viewpoint explains why an aggregation operator is generally preferred to a rule base whereas appraisals are expressed in terms of symbolic labels [4]. A completely symbolic evaluation over finite scales could be envisaged [5].

Let us illustrate the scales problem with the following example. Let us suppose that the semantic universe of an expert w.r.t. the seriousness of a symptom is:  $\{\textit{insignificant}, \textit{serious}, \textit{alarming}\}$ . We can imagine that a corresponding possible set of discrete numerical values (in  $[0; 1]$ ) could be:  $\{0; 0.5; 1\}$ . There are several assumptions behind this translation concerning the nature of the scale. This point will be discussed later. Let us just note here that the numerical values are commonly chosen equidistant. Now let us consider another semantic universe:  $\{\textit{insignificant}, \textit{minor}, \textit{alarming}\}$ . This time, the associated set of numerical values  $\{0; 0.5; 1\}$  intuitively appears more questionable. The expert should prefer  $\{0; 0.25; 1\}$ . When seriousness degrees of several symptoms are to be aggregated, the result of the WAM

aggregation strongly depends on the choice of the set of numerical values. Furthermore, in any case, the numerical WAM value does not necessarily belong to  $\{0; 0.5; 1\}$  or  $\{0; 0.25; 1\}$ . It must then be converted into the convenient label in return.

The way labels are converted into numerical values (and back) coupled to the commensurability of the scales of the dimensions to be aggregated can entail serious problems when aggregating without any care. In this paper, we propose a methodology to build finite partial valuation scales consistently with WAM aggregation.

The paper is organized as follows. Some considerations are given about the way continuous cardinal scales are constructed with the Escota operating domain experts. Then, it is explained how to build a WAM aggregation operator w.r.t. each operating domain, in order to be consistent with the identified scales. The MACBETH method is the support of these first two steps. The problem related to the finite scales, that the experts use when assigning partial scores to an operation, is then considered. A method is proposed to ensure a logically sound interface between symbolic assessments and numerical computations in the framework of WAM aggregation. Then, a robustness analysis is proposed to determine the potential causes of overestimation or underestimation in the evaluation process of an operation.

## 2 Cardinal Scales of Emergency Degress

### 2.1 Nature of Scales

The purpose of this section is to explain how we have worked with Escota experts of the different operating domains in order to properly identify their emergency scales. There are one emergency scale for each criterion of the domain and one scale for the aggregated emergency value. In the following we will consider the case of the operating domain “*carriageway*”. Eight criteria ( $n=8$ ) are related to it: *security*, *durability*, *regulation*, *comfort*, *public image*, *environment protection*, *sanitary* and *social aspects*.

It has been checked a priori that Escota emergency scales are of cardinal nature: the emergency scale relative to any of the criteria is an interval scale.

Let us consider a finite set  $X$ . When the elements of  $X$  can be ranked w.r.t. to their attractiveness, this is ordinal information. It means that a number  $n(x)$  can be associated to any element  $x$  of  $X$  such that:

$$\forall x, y \in X : [xPy \Leftrightarrow n(x) \succ n(y)] \quad (1)$$

$$\forall x, y \in X : [xIy \Leftrightarrow n(x) = n(y)] \quad (2)$$

where relation  $P$  « *is more attractive than* » is asymmetric and non transitive and relation  $I$  « *is as attractive as* » is an equivalence relation.  $n(x)$  defines an ordinal scale.

Based upon this first level of information, an interval scale can then be built. The next step consists in evaluating the difference of intensity of preference between elements of  $X$ . It implies the following constraints:



$$n(x) - n(y) = k\alpha, k \in \mathbb{N} \tag{3}$$

where  $k$  characterizes the intensity of preference and  $\alpha$  enables to respect the limits of the domain (for example  $[0,1]$ ). The resolution of a system of equations of type (1), (2) and (3) provides an interval scale. That's the principle used in the MACBETH method [2].

### 2.2 Emergency Scales and MACBETH Method

The problem of commensurability of the dimensions to be aggregated is at the heart of the MACBETH method. Aggregation can be envisaged only if the scales relative to the emergency criteria are commensurable [3]. Then, MACBETH guarantees the consistency between the resulting partial scales and the WAM aggregation [2].

First, a training set of operations is constituted. A ranking of the operations in terms of emergency is established w.r.t. each criterion. At this stage, information is purely ordinal. Then, for each criterion, the solutions are compared pair to pair. Two fictive alternatives are introduced in the comparison process; they provide the reference values corresponding to the two emergency degrees: zero and one. The zero (resp. one) emergency degree corresponds to the threshold value under which operations are considered as not urgent at all (resp. highly urgent). The comparison then consists in quantifying the difference of emergency degree for each criterion. This difference is expressed in a finite set of labels: for example, “equivalent”, “weak”, “strong” and “extreme”. The resulting set of constraints defines a linear programming problem. The solution of this problem provides the cardinal scale of emergency associated to one criterion. This step is repeated for each criterion.

Fig. 1 illustrates this process for criterion *security*. The carriageway expert compares 10 operations {A...J} pair to pair. The real names of operations are not given for confidentiality reasons. Two fictive operations *urgent* (highly urgent) and *peu\_urgent* (not urgent at all) complete the training base. The “positive” label in Fig. 1 introduces a more flexible constraint because it simply replaces any label with a higher degree than weak. The resulting cardinal scale is given at the right side of Fig. 1.

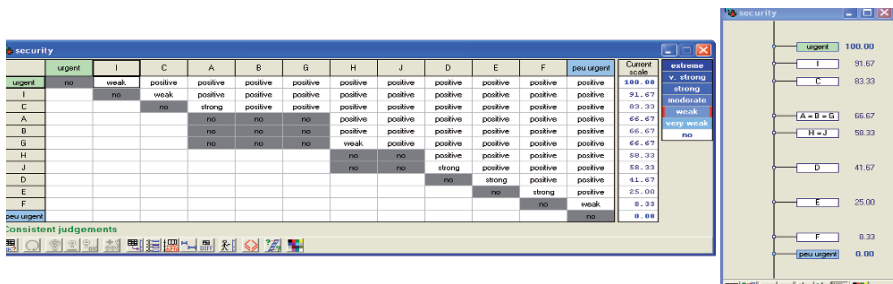


Fig. 1. MACBETH – Pair to pair comparison of operations and cardinal scale for security criterion.

Finally, this procedure is then applied to identify the weights of the WAM operator. The pair to pair comparison is carried out over the eight criteria of the carriageway domain (Fig. 2). The resulting interval scale of weights is given in Fig. 2. Let us note the

weights  $p_i, i = 1..n$  ( $n=8$  for the carriageway domain). At this stage of the modelling, the carriageway expert has identified his 8 emergency scales and his WAM parameters. He is supposed to be able to compute the global degree of emergency of any operation when partial quotations  $u_i$  are available, w.r.t. each criterion:

$$WAM(OP) = \sum_{i=1}^n p_i u_i$$

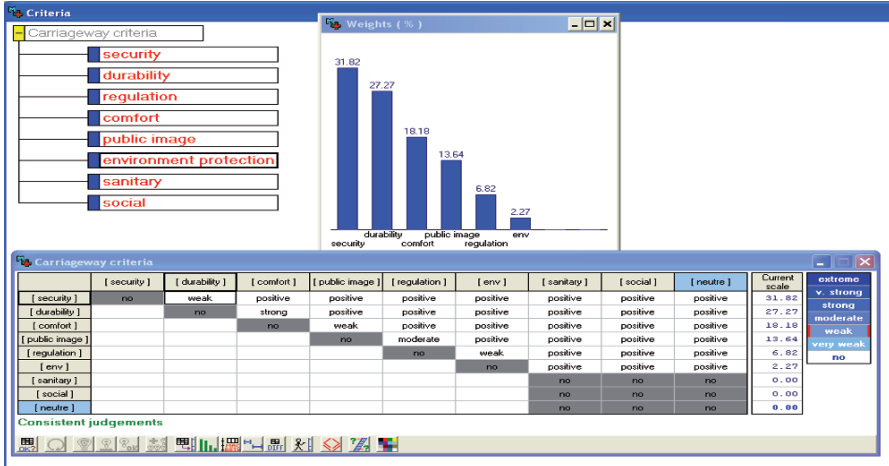


Fig. 2. MACBETH – Pair to pair comparison of carriageway criteria and weights identification.

### 3 Discrete Cardinal Scales of Emergency

Partial scores aggregation does not cause any problem when quotations referred to continuous cardinal scales. As explained in section 1, it is more questionable when partial scores are expressed on a discrete or finite scale. Indeed, Escota experts express their assessment w.r.t. each criterion on a finite set of 3 labels  $\{U_1, U_2, U_3\}$ . The different  $U_i$  define a discrete cardinal scale. However, computing the WAM value necessitates assigning numerical values to each  $U_i$ . In the following, we describe the way this assignment can be achieved in a consistent manner with previous MACBETH identification phases.

A continuous cardinal scale has been identified with MACBETH method for the emergency scale of each criterion. The problem is now to assign a set of numerical values  $\{u_1^i, u_2^i, u_3^i\}$  to  $\{U_1, U_2, U_3\}$  for criterion  $i$ . Let us suppose the continuous cardinal scale for criterion  $i$  has been identified with a training set of  $q$  operations. These operations are grouped into 3 clusters corresponding to  $U_1, U_2, U_3$ . The computation of the clusters and their associated centres is achieved by minimizing the quadratic difference  $\sum_{k=1}^3 \sum_{j=1}^{q_k} (u_k^i - u^i(OP_j))^2$  where  $q_k$  is the number of operations in

class  $U_k$  ( $\sum_{k=1}^3 q_k = q$ ) and  $u^i(OP_j)$ ,  $j=1..q$ , the emergency degree of an operation  $OP_j$  computed with MACBETH (Fig. 1).

In the example of Fig. 1, the computation of clusters gives:  $u_1^{security} = 0.91$ ,  $u_2^{security} = 0.52$  and  $u_3^{security} = 0.11$ .

This assignment is repeated for each criterion relative to the carriageway domain. Then, the WAM can be numerically computed:

- For each criterion  $i$ ,  $i = 1..n$  ( $n = 8$ ), a value  $U_k$  is affected to an operation  $OP$ . Let us note this emergency degree  $U_{k(i)}$ ;
- $OP$  is thus described by its vector of emergency degrees  $[U_{k(1)}, \dots, U_{k(n)}]$ ;
- The corresponding vector of numerical values is:  $\{u_{k(1)}^1, u_{k(2)}^2, \dots, u_{k(n)}^n\}$ ;

$$WAM(OP) = \sum_{i=1}^n p_i \cdot u_{k(i)}^i \quad (4)$$

The last constraint to be satisfied is that the  $WAM$  values must be converted in return into the semantic universe  $\{U_1, U_2, U_3\}$ . The output of the  $WAM$  operator must be discretized in  $\{U_1, U_2, U_3\}$ . The problem is thus to determine the centres of the  $U_k$  clusters of the aggregated emergency scale ( $WAM$  values).

Let us note that the  $WAM$  operator is idempotent. Therefore, we must have:

$$\forall U_k, k \in \{1, 2, 3\}, WAM(U_k, \dots, U_k) = U_k \quad (5)$$

A sufficient condition for (5) is that the centres of the  $U_k$  clusters of the aggregated emergency scale are the images of the corresponding  $U_k$  centres of the partial emergency scales by the  $WAM$  function, i.e.:

$$WAM(u_k^1, \dots, u_k^n) = \sum_{i=1}^n p_i \cdot u_k^i = u_k^{Ag} \quad (6)$$

where  $u_k^{Ag}$  is the centre of class  $U_k$  in the aggregated emergency scale.

Consequently, when an operation is defined by its partial emergency vector  $[U_{k(1)}, \dots, U_{k(n)}]$ , equation (4) provides the numerical value

$$WAM(OP) = \sum_{i=1}^n p_i \cdot u_i \quad (7)$$

Then, the attribution of a class  $U_k$  in the aggregated emergency scale is obtained through the following calculation:

$$\min_k \left| u_k^{Ag} - \sum_{i=1}^n p_i \cdot u_{k(i)}^i \right| \quad (8)$$

The value of  $k$  in  $\{1, 2, 3\}$  that minimizes the expression in (8) provides the class  $U_k$  of operation  $OP$ .

Fig. 3 summarizes the whole evaluation process of an operation  $OP$ . The validation of this process has been carried out with a test base of 23 operations in the carriageway domain. The carriageway expert has analysed each of these operations. For each of them, he has attributed emergency degrees in the Escota normalized semantic universe  $\{U_1, U_2, U_3\}$  w.r.t. every of his 8 criteria.

Then, the aggregated emergency degree in this semantic universe can be computed using the 3-step process described in this paper (white arrows in Fig. 3). Besides these computations, the expert has been asked to directly attribute an overall emergency degree to each of the 23 operations (grey arrow in Fig. 3).

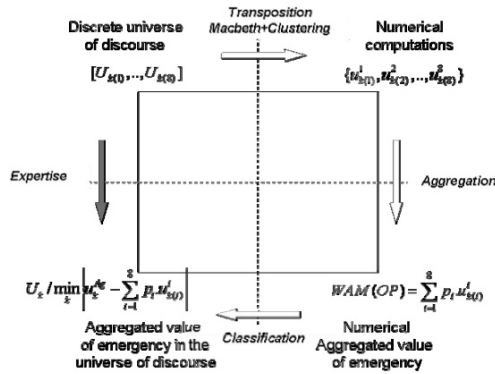


Fig. 3. Evaluation process of an operation.

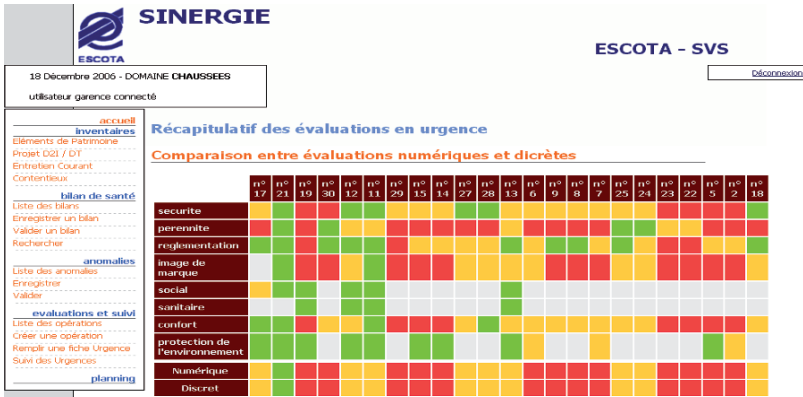


Fig. 4. Tests on the evaluation method over a base of 23 operations U1 U2 U3

Fig. 4 reports these data. The last line corresponds to the direct expert evaluation (grey arrow). The last but one line provides the corresponding computed values with the 3-step method (white arrows). No error has been observed. However, the poor

semantic universe—only 3 labels—implied in our application can also partly explain such a perfect matching.

### 4 The MC Hierarchical Evaluation by Escota

In this paper, the study was focused on the MC evaluation by the expert of an operating domain. However, as evocated in section 1, planning of operations, by Escota, is more complex. The emergency assessment by operating domain experts described here is only part of a hierarchical MC evaluation process. From symptoms detection on elements of infrastructure to operation planning, a similar MC evaluation is carried out at different functional levels in the Escota organization.

The complete information processing used for Escota preventive maintenance can be formalized as the following sequence of risk analysis. Periodic inspections are performed to detect and measure any malfunction symptoms as early as possible. The expert in charge of a domain then analyses these technical diagnoses and evaluates the situation seriousness. The official in charge of the operating network coordinates and ponders the experts’ needs and demands. Each actor of this information processing system participates to a tripartite MC decision-making logic: measurement, evaluation and decision. To each step of this process corresponds a specific set of criteria and an aggregation operator: seriousness of a malfunction results from a prescribed aggregation of the symptoms quotation; the expert’s interpretation of the diagnosis associates an emergency degree to the corresponding maintenance operation w.r.t. the criteria relating to his operating domain (technical risks assessment); finally, the manager attributes a priority degree to the operation on the basis of a set of more strategic criteria (strategic risks analysis).

This hierarchical MC evaluation process enables to breakdown the decision-making into elementary steps. Each step collaborates to the enrichment of information from measures to priority degrees and thus contributes to the final step, i.e. operation planning.

We have developed a dynamic Information Processing System (IPS) to support this hierarchical MC evaluation of the infrastructure condition and facilitate the way decision-makers use their reasoning capabilities through adequate information processing procedure. Fig. 5 illustrates the man machine-interface the expert has at his disposal to fulfil an emergency form relative to an operation.

Fiche d'urgence pour l'A808 km 1 et 2

Titre :	refection couche de roulement				
Description :	fissuration longitudinale sur la voie lente				
	U1	U2	U3	D	Commentaires
securite :	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	zone urbaine et zones de virages
perennite :	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	fissuration longitudinale qui a evol
image de marque :	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	fissuration visible
confort :	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	
protection de l'environnement :	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	
reglementation :	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	
social :	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	
santaire :	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	
montant estime (€) :	0,0				

Fig. 5. Keyboarding of an emergency form.

Finally, the emergency evaluation synthesis (Fig. 6) can be consulted by the official in charge of the operation network before he proceeds to his own MC evaluation.

**Fiche d'urgence pour l'A808 km 1 et 2**

**Titre :** reftection couche de roulement

**Description :** fissuration longitudinale sur la voie lente

Critère	U1	U2	U3	D	Commentaires
securite :	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	zone urbaine et zones de virages
perennite :	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	fissuration longitudinale qui a evol
image de marque :	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	fissuration visible
confort :	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	
protection de l'environnement :	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	
reglementation :	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	
social :	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	
sanitaire :	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	
montant estimé (k€) :	0,0				

**Fig. 6.** Emergency evaluation synthesis.

## 5 The Robustness Analysis of the Evaluation Process

Let us now consider a last step in the evaluation process: assessment of the risk of erroneous estimation w.r.t. the emergency of an operation, i.e., the risk of underestimation or overestimation of the aggregated emergency score of an operation. It relies on a robustness analysis of the evaluation procedure based upon the WAM. Two aims are assigned to this step, it must answer the following questions: 1) when an erroneous partial estimation is done w.r.t. criterion  $i$ , what is the risk the aggregated emergency degree to be affected? 2) when an operation appears to be underestimated (resp. overestimated), which criteria could most likely explain this faulty result? The first question corresponds to *an a priori risk estimation* of erroneous evaluation; the second question is related to *a diagnosis analysis*.

Let us first define the notion of neighbourhood of a vector of emergency degrees  $[U_{k(1)}, \dots, U_{k(n)}]$  associated to an operation  $OP$ . The vectors of the neighbourhood of  $[U_{k(1)}, \dots, U_{k(n)}]$  are all the vectors  $[U'_{k(1)}, \dots, U'_{k(n)}]$  such that:  $\forall i \in \{1..n\}, U'_{k(i)} = U_{k(i)}$  or  $U'_{k(i)}$  is the value just above (resp. below)  $U_{k(i)}$  (when defined; indeed, there is no value below zero and no value above  $U_1$ ). The neighbourhood is a set of vectors denoted  $\mathcal{N}([U_{k(1)}, \dots, U_{k(n)}])$ . In the example in dimension 2 in Fig. 7,  $U_{k(1)} = U_2$  and  $U_{k(2)} = U_2$ . The values of component  $i$  ( $i = 1 \text{ or } 2$ ) of a neighbour vector may be  $U_2$ ,  $U_1$  or  $U_3$ . There are 8 neighbours. In the general case, the maximal number of neighbours is  $3^n - 1$ .

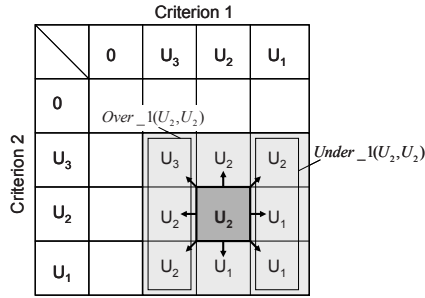


Fig. 7. Neighbourhood of the vector of emergency degrees (U<sub>2</sub>, U<sub>2</sub>) in dimension 2.

### 5.1 Risk of Erroneous Estimation

The risk of misclassification of an operation due to an overestimation (resp. underestimation) w.r.t. a criterion *i* enables the expert in charge of a domain to assess the impact of an evaluation error w.r.t. criterion *i* on the overall emergency degree of the operation. The higher, the more carefully the partial appraisal w.r.t. criterion *i* must be carried out. The lower, the weaker the impact of the criterion to the global emergency degree. The risk analysis is based upon the following algorithm. We'll first consider the risk of underestimation for sake of simplicity. We consider that a value  $U_{k(i)}$  is underestimated (resp. overestimated) when it should take the value just above  $U_{k(i)}$  (resp. just below  $U_{k(i)}$ ). This assumption means that the worst appraisal error w.r.t. one criterion can only correspond to the value just below or just above for this criterion.

Let's consider a vector  $U = [U_{k(1)}, \dots, U_{k(n)}]$

Compute  $WAM(U)$

For each criterion *i* :

- Find all the vectors  $U' = [U'_{k(1)}, \dots, U'_{k(n)}]$  in  $N(U)$  such that  $U'_{k(i)}$  takes the value just above  $U_{k(i)}$  (when defined, else  $U_{k(i)} = U_1$  and there is no risk of underestimation w.r.t. criterion *i* in this case). Note this set:  $Under\_i(U)$
- Count the numbers of vectors  $U'$  in  $Under\_i(U)$  such that  $WAM(U')$  is higher than  $WAM(U)$ . Note this number  $n_{under\_i}$
- The risk of underestimation induced by criterion *i* for an operation characterized by  $U$  is then:

$$risk\_under(i) = \frac{n_{under\_i}}{|Under\_i(U)|}$$

In the example in Fig. 7, let us consider an assumption of underestimation w.r.t. criterion 1. The set  $Under\_1(U_2, U_2)$  is represented in the figure.  $|Under\_1(U_2, U_2)| = 3$ ; only  $(U_1, U_2)$  and  $(U_1, U_1)$  lead to an overall underestimation

(the operation is evaluated  $U_2$  whereas it should be  $U_1$ ). Then,  $n_{under\_1} = 2$  and  $risk\_under(1) = 2/3$ . It means that an underestimation w.r.t. criterion 1 for an operation characterized by  $(U_2, U_2)$  leads to an underestimation of the overall degree of emergency of the operation in 66% of the cases.

The algorithm is the same for the risk of overestimation. Nevertheless, in this case, when  $U_{k(i)} = 0$ , the risk of overestimation w.r.t. criterion  $i$  is null. Fig. 8 and Figure 9 provide the results for the risk analysis when underestimation (Fig. 8) and when overestimation (Fig. 9) for all the vectors in Fig. 4.

## 5.2 Diagnosis Analyses

When the degree of emergency of an operation is suspected to be overestimated (resp. underestimated), the diagnosis analysis consists in determining the most likely causes, i.e., the criteria that the most frequently entail an overestimation (resp. underestimation) of the operation when they're overestimated (resp. underestimated) themselves. The possibility that criterion  $i$  is a cause of overestimation (resp. underestimation) assuming an overestimation (resp. underestimation) of the overall emergency degree of the operation— is computed in the diagnosis step.

Let us consider the algorithm in case of underestimation (resp. overestimation).  
 Let's consider a vector  $U = [U_{k(1)}, \dots, U_{k(n)}]$   
 Compute  $WAM(U)$   
 Compute  $N(U)$  and its cardinal  $|N(U)|$

- Compute  $WAM(U')$  for each  $U' = [U'_{k(1)}, \dots, U'_{k(n)}]$  in  $N(U)$
- Let us note  $Higher\_N(U)$  (resp.  $Lower\_N(U)$ ), the set of vectors  $U'$  in  $N(U)$  such that  $WAM(U') > WAM(U)$  (resp.  $WAM(U') < WAM(U)$ )
- For each criterion  $i$ , count the number  $n'_{under\_i}$  (resp.  $n'_{over\_i}$ ) of times criterion  $i$  is underestimated (resp. overestimated) in a vector of  $Higher\_N(U)$  (resp.  $Lower\_N(U)$ ), i.e.,  $U'_{k(i)}$  takes the value just above  $U_{k(i)}$  (resp. just below  $U_{k(i)}$ ) in  $Higher\_N(U)$  (resp.  $Lower\_N(U)$ )
- Compute for each criterion  $i$ :  

$$Diag\_under(i) = \frac{n'_{under\_i}}{|Higher\_N(U)|}$$
  

$$Diag\_over(i) = \frac{n'_{over\_i}}{|Lower\_N(U)|}$$
 (resp.

$Diag\_under(i)$  gives the rate that an underestimation w.r.t. criterion  $i$  be a potential cause of underestimation of the overall emergency degree of an operation (idem for overestimation).

Fig. 10 concerns underestimation diagnosis and Fig. 11 overestimation diagnosis for the base of operations in Fig. 4. A rate indicates the possibility a criterion is



underestimated itself (resp. overestimated) when the overall emergency degree of the concerned operation is underestimated (resp. overestimated).

	17	21	19	30	12	11	29	15	14	27	28	13	6	9	8	7	25	24	23	22	5	2	18
env	11.0%	44.0%	0%	0%	15.0%	2.0%	0%	0%	0%	26.0%	13.0%	41.0%	0%	0%	0%	0%	18.0%	21.0%	0%	0%	0%	0%	21.0%
sanitary	11.0%	42.0%	0%	0%	14.0%	1.0%	0%	0%	0%	26.0%	13.0%	39.0%	0%	0%	0%	0%	18.0%	21.0%	0%	0%	0%	0%	21.0%
comfort	25.0%	58.0%	0%	0%	25.0%	4.0%	0%	0%	0%	40.0%	27.0%	55.0%	0%	0%	0%	0%	28.0%	32.0%	0%	0%	0%	0%	37.0%
regulation	11.0%	45.0%	0%	0%	16.0%	2.0%	0%	0%	0%	34.0%	20.0%	43.0%	0%	0%	0%	0%	20.0%	25.0%	0%	0%	0%	0%	23.0%
security	32.0%	67.0%	0%	0%	37.0%	4.0%	0%	0%	0%	62.0%	35.0%	76.0%	0%	0%	0%	0%	41.0%	46.0%	0%	0%	0%	0%	55.0%
durability	15.0%	73.0%	0%	0%	28.0%	4.0%	0%	0%	0%	32.0%	17.0%	62.0%	0%	0%	0%	0%	39.0%	44.0%	0%	0%	0%	0%	27.0%
social	11.0%	42.0%	0%	0%	14.0%	1.0%	0%	0%	0%	26.0%	13.0%	39.0%	0%	0%	0%	0%	18.0%	21.0%	0%	0%	0%	0%	21.0%
public image	18.0%	56.0%	0%	0%	26.0%	4.0%	0%	0%	0%	44.0%	27.0%	58.0%	0%	0%	0%	0%	29.0%	34.0%	0%	0%	0%	0%	39.0%

Fig. 8. Risk of overall underestimation of the operations induced by partial underestimations w.r.t. criteria.

	17	21	19	30	12	11	29	15	14	27	28	13	6	9	8	7	25	24	23	22	5	2	18
env	2.0%	3.0%	0.0%	53.0%	7.0%	30.0%	12.0%	17.0%	17.0%	1.0%	6.0%	0.0%	41.0%	34.0%	34.0%	28.0%	13.0%	11.0%	6.0%	6.0%	0.0%	0.0%	1.0%
sanitary	2.0%	3.0%	0.0%	53.0%	6.0%	29.0%	12.0%	16.0%	16.0%	1.0%	6.0%	0.0%	34.0%	34.0%	34.0%	22.0%	13.0%	11.0%	6.0%	6.0%	0.0%	0.0%	1.0%
comfort	8.0%	7.0%	1.0%	69.0%	13.0%	46.0%	19.0%	24.0%	24.0%	3.0%	14.0%	2.0%	51.0%	51.0%	51.0%	39.0%	20.0%	18.0%	12.0%	12.0%	2.0%	1.0%	3.0%
regulation	3.0%	3.0%	1.0%	58.0%	9.0%	32.0%	19.0%	23.0%	23.0%	1.0%	6.0%	1.0%	40.0%	35.0%	35.0%	28.0%	14.0%	14.0%	7.0%	7.0%	1.0%	0.0%	1.0%
security	8.0%	9.0%	1.0%	72.0%	18.0%	63.0%	35.0%	47.0%	47.0%	3.0%	18.0%	2.0%	74.0%	80.0%	80.0%	56.0%	30.0%	27.0%	19.0%	19.0%	2.0%	1.0%	3.0%
durability	8.0%	9.0%	1.0%	96.0%	17.0%	49.0%	24.0%	30.0%	30.0%	3.0%	11.0%	2.0%	48.0%	48.0%	48.0%	36.0%	39.0%	34.0%	18.0%	18.0%	2.0%	1.0%	3.0%
social	2.0%	3.0%	0.0%	53.0%	6.0%	29.0%	12.0%	16.0%	16.0%	1.0%	6.0%	0.0%	34.0%	34.0%	34.0%	22.0%	13.0%	11.0%	6.0%	6.0%	0.0%	0.0%	1.0%
public image	3.0%	5.0%	1.0%	60.0%	15.0%	44.0%	24.0%	29.0%	29.0%	3.0%	14.0%	2.0%	53.0%	48.0%	48.0%	33.0%	20.0%	18.0%	12.0%	12.0%	2.0%	1.0%	3.0%

Fig. 9. Risk of overall overestimation of the operations induced by partial overestimations w.r.t. criteria.

	17	21	19	30	12	11	29	15	14	27	28	13	6	9	8	7	25	24	23	22	5	2	18
durability	45%	57%	0%	0%	67%	100%	0%	0%	0%	40%	42%	52%	0%	0%	0%	0%	71%	67%	0%	0%	0%	0%	41%
security	97%	53%	0%	0%	86%	100%	0%	0%	0%	79%	87%	64%	0%	0%	0%	0%	75%	71%	0%	0%	0%	0%	84%
comfort	77%	45%	0%	0%	59%	90%	0%	0%	0%	51%	66%	46%	0%	0%	0%	0%	51%	49%	0%	0%	0%	0%	56%
public image	55%	44%	0%	0%	62%	90%	0%	0%	0%	56%	68%	48%	0%	0%	0%	0%	53%	52%	0%	0%	0%	0%	60%
env	35%	35%	0%	0%	36%	45%	0%	0%	0%	33%	33%	35%	0%	0%	0%	0%	33%	33%	0%	0%	0%	0%	33%
regulation	35%	35%	0%	0%	37%	45%	0%	0%	0%	43%	51%	36%	0%	0%	0%	0%	37%	39%	0%	0%	0%	0%	35%
social	33%	33%	0%	0%	33%	33%	0%	0%	0%	33%	33%	33%	0%	0%	0%	0%	33%	33%	0%	0%	0%	0%	33%
sanitary	33%	33%	0%	0%	33%	33%	0%	0%	0%	33%	33%	33%	0%	0%	0%	0%	33%	33%	0%	0%	0%	0%	33%

Fig. 10. Rates of causes of underestimation diagnoses.

	17	21	19	30	12	11	29	15	14	27	28	13	6	9	8	7	25	24	23	22	5	2	18
durability	100%	100%	100%	60%	82%	57%	66%	61%	61%	100%	60%	100%	46%	46%	46%	53%	94%	96%	93%	93%	100%	100%	100%
security	100%	100%	100%	46%	90%	72%	96%	95%	95%	100%	100%	100%	71%	78%	78%	84%	73%	75%	100%	100%	100%	100%	100%
comfort	100%	78%	100%	44%	66%	53%	53%	48%	48%	100%	80%	100%	50%	50%	50%	57%	50%	51%	62%	62%	100%	100%	100%
public image	45%	60%	100%	38%	74%	50%	66%	58%	58%	100%	80%	100%	51%	46%	46%	50%	62%	62%	100%	100%	100%	100%	100%
env	35%	34%	33%	33%	35%	34%	33%	35%	35%	33%	33%	40%	40%	33%	33%	42%	33%	33%	33%	33%	33%	50%	33%
regulation	40%	39%	100%	35%	43%	36%	53%	46%	46%	33%	33%	60%	39%	34%	34%	42%	35%	41%	37%	37%	50%	50%	33%
social	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%
sanitary	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%	33%

Fig. 11. Rates of causes of overestimation diagnoses.

## 6 Conclusions

In civil engineering, the culture of numbers is strongly developed. People commonly manipulate symbolic labels but attribute them numerical values when necessary without further care. A typical case is when aggregation procedures are required. We have proposed a methodology that enables 1) experts to express their judgement

values in their own discrete semantic universe, 2) to convert the labels in adequate numerical values using the MACBETH method and clustering techniques, 3) to compute the WAM based aggregated value and convert it in return into the experts' semantic universe 4) to carry out a robustness analysis of the evaluation process to assess the risk of misclassification of the operations and to diagnose these misclassifications. This method is implemented in an IPS—SINERGIE—that supports decisions concerning maintenance operations planning by the motorway operator Escota.

## References

1. Saaty, T.L.: The Analytic Hierarchy Process. McGraw-Hill, New York (1980)
2. Bana e Costa, C.A., Vansnick, J.C.: MACBETH - an interactive path towards the construction of cardinal value functions. *International transactions in Operational Research*, vol. 1, pp. 489–500 (1994)
3. Clivillé, V. : Approche Systémique et méthode multicritère pour la définition d'un système d'indicateurs de performance. Thèse de l'Université de Savoie, Annecy (2004)
4. Jullien, S., Mauris, G., Valet, L., Bolon, Ph.: Decision aiding tools for Animated film selection from a mean aggregation of criteria preferences over a finite scale. 11th Int. Conference on Information processing and Management of uncertainty in Knowledge-Based Systems, IPMU, Paris, France (2006)
5. Grabisch, M.: Representation of preferences over a finite scale by a mean operator. *Mathematical Social Sciences*, vol. 52, pp. 131–151 (2006)
6. Akharraz A., Montmain J., Mauris G.: A project decision support system based on an elucidative fusion system, Fusion 2002, 5th International Conference on Information Fusion, Annapolis, Maryland, USA (2002)
7. Akharraz A., Montmain J., Denguir A., Mauris G., Information System and Decisional Risk Control for a Cybernetic Modeling of Project Management. 5<sup>th</sup> international conference on computer science (MCO 04), Metz, France, pp. 407–414 (2004).

# A Multiple Sensor Fault Detection Method based on Fuzzy Parametric Approach

Frederic Lafont, Nathalie Pessel and Jean-Francois Balmat

University of South-Toulon-Var, LSIS UMR CNRS 6168

B.P 20132, 83957 La Garde Cedex, France

lafont, nathalie.pessel, balmat@univ-tln.fr

**Abstract.** This paper presents a new approach for the model-based diagnosis. The model is based on an adaptation with a variable forgetting factor. The variation of this factor is managed thanks to fuzzy logic. Thus, we propose a design method of a diagnosis system for the sensor defaults. In this study, the adaptive model is developed theoretically for the Multiple-Input Multiple-Output (MIMO) systems. We present the design stages of the fuzzy adaptive model and we give details of the Fault Detection and Isolation (FDI) principle. This approach is validated with a benchmark: a hydraulic process with three tanks. Different defaults (sensors) are simulated with the fuzzy adaptive model and the fuzzy approach for the diagnosis is compared with the residues method. The method is efficient to detect and isolate one or more defaults. The results obtained are promising and seem applicable to a set of MIMO systems.

**Keywords.** Adaptive model, fuzzy system models, diagnosis, Fault Detection and Isolation (FDI).

## 1 Introduction

The automatic control of technical systems requires a fault detection to improve reliability, safety and economy. The diagnosis is the detection, the isolation and the identification of the type as well as the probable cause of a failure using a logical reasoning based on a set of information coming from an inspection, a control or a test (AFNOR, CEI) [1], [2]. The model-based diagnosis is largely studied in the literature [3], [4], [5]. These methods are based on parameter estimation, parity equations or state observers [3], [4], [6]. The goal is to generate the indicators of defaults through the generation of residues [7] (Fig. 1).

This paper deals with the problem of the model-based diagnosis by using a parametric estimation method. We particularly focus our study on an approach with an adaptive model. Many methods exist which enable the design of these adaptive models [3].

Many works tackle the model-based diagnosis from a fuzzy model of the processes [8], [9], [10], [11], [12].

Sala et al. [13] notices that Higher decision levels in process control also use rule bases for decision support. Supervision, diagnosis and condition monitoring are examples of successful application domains for fuzzy reasoning strategies.

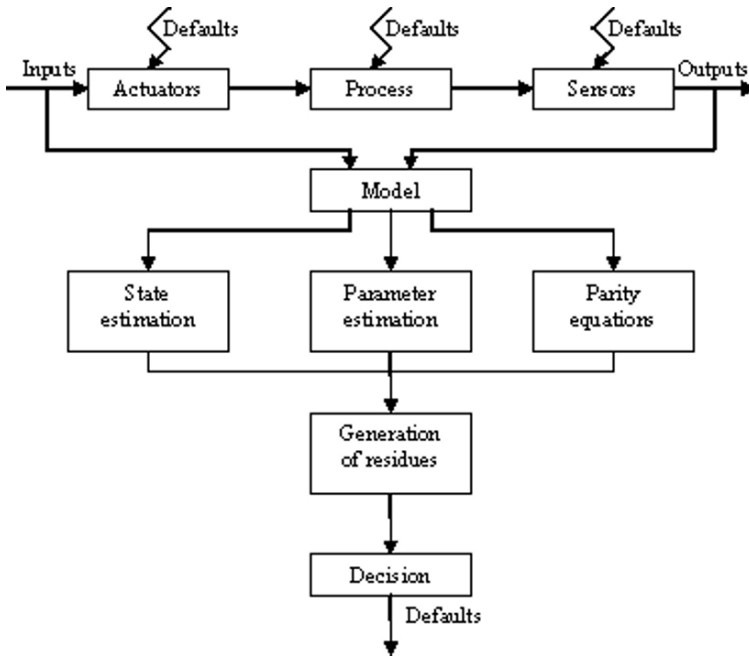


Fig. 1. Model-based diagnosis.

In our work, unlike these approaches, fuzzy logic is used to design the parametric model.

In all cases, for the model-based approaches, the quality of the fault detection and isolation depends on the quality of the model.

It is possible to improve the model identification by implementing an original method based on a parameters adjustment by using a Fuzzy Forgetting Factor (FFF) [14]. The idea, in this study, is to use the variations of the fuzzy forgetting factors for the fault detection and isolation. Thus, we propose an original method based on a fuzzy adaptation of the parameter adjustments by introducing a fuzzy forgetting factor. From these factors (one by output), we can generate residues for the fault detection and isolation.

This paper is an extension of the approach presented at the fourth International conference on Informatics in Control, Automation and Robotics [15] to detect and isolate several defaults. It is now developed to detect several simultaneous defaults. The paper is organised as follow: first, we present the principle of the fuzzy forgetting factor. Then, we summarize the different stages of generation of residues and decision-making. In section 4, we present the application of the method to diagnosis of a hydraulic process. A numerical example, with several types of sensor defaults (the bias and the calibration default), is presented to show the performances of this method.

## 2 A New Approach: The Fuzzy Forgetting Factor Method

In this section, after having presented the classical approach for the on-line identification, we present a new method of adaptation based on the fuzzy forgetting factor variation [15].

We consider a modeling of non-linear and non-stationary systems. Consequently, an on-line adaptation is necessary to obtain a valid model capable of describing the process and allowing to realize an adaptive command [16]. A common technique for estimating the unknown parameters is the Recursive Least Squares (RLS) algorithm with forgetting factor [17], [18], [19].

At each moment  $k$ , we obtain a model, such as:

$$y(k+1) = A(k)y(k) + B(k)u(k) \quad (1)$$

with  $y$  the outputs vector and  $u$  the command vector,

$$\varphi(k) = (y(k)u(k))^T \quad (2)$$

$$\hat{y}(k+1) = \hat{\theta}^T(k)\varphi(k) \quad (3)$$

$$\hat{\theta}(k+1) = \hat{\theta}(k) + m(k+1)P(k)\varphi^T(k)\epsilon(k+1) \quad (4)$$

$$\epsilon(k+1) = y(k+1) - \hat{y}(k+1) \quad (5)$$

$$P(k+1) = \frac{1}{\lambda(k)} \left[ P(k) - \frac{P(k)\varphi(k)\varphi^T(k)P(k)}{\lambda(k) + \varphi^T(k)P(k)\varphi(k)} \right] \quad (6)$$

with  $\hat{\theta}(k)$  the estimated parameters vector (initialized with the least-squares algorithm),  $\varphi(k)$  the regression vector,  $\epsilon(k+1)$  the a-posterior error,  $P(k)$  the gain matrix of regular adaptation and  $\lambda(k)$  the forgetting factor.

If the process is slightly excited, the gain matrix  $P(k)$  increases like an exponential [20]. To avoid this problem, and the drift of parameters, a measure  $m(k)$  is introduced as:

$$m(k+1) = 1 \text{ if } \left| \frac{u(k+1) - u(k)}{u_{max}} \right| > S_u \quad (7)$$

$$\text{or if } \left| \frac{y(k+1) - \hat{\theta}^T(k)\varphi(k)}{y_n} \right| > S_y \quad (8)$$

$$m(k+1) = 0 \text{ if } \left| \frac{u(k+1) - u(k)}{u_{max}} \right| < S_u \quad (9)$$

$$\text{and if } \left| \frac{y(k+1) - \hat{\theta}^T(k)\varphi(k)}{y_n} \right| < S_y \quad (10)$$

with  $y_n$  the nominal value of  $y$ .

The adaptation is suspended as soon as the input becomes practically constant and/or as soon as the output  $y$  reaches a predefined tolerance area from the thresholds  $S_y$  and/or  $S_u$ . In the opposite case, and/or when a disturbance is detected on the input, the adaptation resumes with  $m(k)=1$ .

The adaptation gain can be interpreted like a measurement of the parametric error. When an initial estimation of the parameters is available, the initial gain matrix is:

$$P(0) = GI \quad (11)$$

With  $G \ll 1$  or  $Trace < 1$  and  $I$ : identity matrix.

We choose as initial values:

$$P(0) = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \quad (12)$$

$$\lambda(0) = 0.96 \quad (13)$$

## 2.1 Methods of the Forgetting Factor Variation

The considered class of the system imposes to use a method with a variable forgetting factor in order to take into account the non-stationarity of the process.

Generally, the adaptation of a model is obtained by using a RLS algorithm with forgetting factor. The forgetting factor can be constant or variable.

There are different classical methods of the forgetting factor variation as, for example, the exponential forgetting factor. The variation of  $\lambda$  is defined as:

$$\lambda(k+1) = \lambda_0 \lambda(k) + (1 - \lambda_0) \quad (14)$$

where

$$0 < \lambda_0 < 1 \quad (15)$$

with the typical values:

$$\lambda_0 = 0.95 \cdots 0.99 \quad \lambda(0) = 0.95 \cdots 0.99 \quad (16)$$

This method consists in increasing  $\lambda$  to 1 rapidly.

Andersson proposes to modify the gain matrix  $P(k+1)$  of the RLS algorithm to improve the model [21]. This method introduces an Adaptive forgetting Factor through Multiple Models (AFMM) in considering the RLS algorithm as a special case of the Kalman filter.  $\hat{\theta}(k+1)$  is approximated with a sum of many Gaussian density functions. Moreover, when the process is subjected to jumps, this method enables us to reduce the importance of the gain matrix  $P(k+1)$  in adjusting a parameter.

A new identification algorithm, inspired by these two methods (exponential and Andersson), is proposed. This approach presents a Fuzzy Forgetting Factor [14].

### 2.2 The Proposed Approach

We use fuzzy logic to modify the forgetting factor in an automatic and optimal way [22]. Thus, we have defined a fuzzy box of Mamdani type by using the following variables:  $\lambda(k)$  and  $\Delta\epsilon(k)$  in input and  $\lambda(k+1)$  in output (Fig. 2).

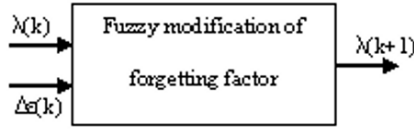


Fig. 2. Fuzzy box.

$\Delta\epsilon(k)$  represents the variation of the mean error on the  $N$  last samples:

$$\Delta\epsilon(k) = \frac{1}{N} \sum_{j=k-N+1}^k (\epsilon(j) - \epsilon(j-1)) \tag{17}$$

$\Delta\epsilon(k)$  is defined with three membership functions: one for the negative error, one for the null error and one for the positive error (Fig. 3). A study of observed process allows to determine the values:  $\{-\eta_{max}; -\eta_{min}; \eta_{min}; \eta_{max}\}$ .

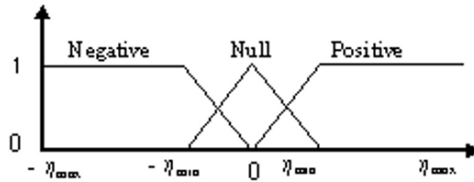


Fig. 3. Fuzzyfication of the error variation.

$$\mu_{negative} : [-\eta_{max}, \eta_{max}] \rightarrow [0, 1] \tag{18}$$

$$v \rightarrow \left\{ \begin{array}{l} \mu_{negative}(v) = 1 \text{ if } v \leq -\eta_{min} \\ \mu_{negative}(v) = \frac{-1}{\eta_{min}} * v \text{ if } -\eta_{min} < v < 0 \\ \mu_{negative}(v) = 0 \text{ if } v \geq 0 \end{array} \right\} \tag{19}$$

$$\mu_{null} : [-\eta_{max}, \eta_{max}] \rightarrow [0, 1] \tag{20}$$

$$v \rightarrow \left\{ \begin{array}{l} \mu_{null}(v) = 0 \text{ if } v \leq -\eta_{min} \\ \mu_{null}(v) = \frac{1}{\eta_{min}} * v + 1 \text{ if } -\eta_{min} < v < 0 \\ \mu_{null}(v) = \frac{-1}{\eta_{min}} * v + 1 \text{ if } 0 \leq v < \eta_{min} \\ \mu_{null}(v) = 0 \text{ if } v \geq \eta_{min} \end{array} \right\} \tag{21}$$

$$\mu_{positive} : [-\eta_{max}, \eta_{max}] \rightarrow [0, 1] \quad (22)$$

$$v \rightarrow \left\{ \begin{array}{l} \mu_{positive}(v) = 0 \text{ if } v \leq 0 \\ \mu_{positive}(v) = \frac{1}{\eta_{min}} * v \text{ if } 0 < v < \eta_{min} \\ \mu_{positive}(v) = 1 \text{ if } v \geq \eta_{min} \end{array} \right\} \quad (23)$$

The membership functions of the input  $\lambda(k)$  and the output  $\lambda(k + 1)$  are identical (Fig. 4).

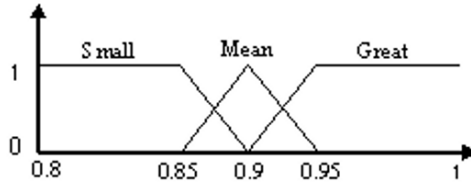


Fig. 4. Fuzzyfication of the lambda.

$$\mu_{small} : [0.8, 1] \rightarrow [0, 1] \quad (24)$$

$$v \rightarrow \left\{ \begin{array}{l} \mu_{small}(v) = 1 \text{ if } v \leq 0.85 \\ \mu_{small}(v) = -20 * v + 18 \text{ if } 0.85 < v < 0.9 \\ \mu_{small}(v) = 0 \text{ if } v \geq 0.9 \end{array} \right\} \quad (25)$$

$$\mu_{mean} : [0.8, 1] \rightarrow [0, 1] \quad (26)$$

$$v \rightarrow \left\{ \begin{array}{l} \mu_{mean}(v) = 0 \text{ if } v \leq 0.85 \\ \mu_{mean}(v) = 20 * v - 17 \text{ if } 0.85 < v < 0.9 \\ \mu_{mean}(v) = -20 * v + 19 \text{ if } 0.9 \leq v < 0.95 \\ \mu_{mean}(v) = 0 \text{ if } v \geq 0.95 \end{array} \right\} \quad (27)$$

$$\mu_{great} : [0.8, 1] \rightarrow [0, 1] \quad (28)$$

$$v \rightarrow \left\{ \begin{array}{l} \mu_{great}(v) = 0 \text{ if } v \leq 0.9 \\ \mu_{great}(v) = 20 * v - 18 \text{ if } 0.9 < v < 0.95 \\ \mu_{great}(v) = 1 \text{ if } v \geq 0.95 \end{array} \right\} \quad (29)$$

According to the application, the bounds [0.8 ; 1] can be reduced.



The inference rules are based on the variation method of the exponential forgetting factor. In this case, the forgetting factor must be maximum when the modeling of the system is correct (small error variation). Also, we have been inspired by Andersson’s work. When there is an important non-stationarity, the forgetting factor must decrease.

If  $\lambda(k)$  is  $F_{n_\lambda}^1$  and  $\Delta\epsilon(k)$  is  $F_{n_\epsilon}^2$  then  $\lambda(k+1)$  is  $F_{n'_\lambda}^3$ , where  $F_{n_\lambda}^1 \in \{F_1^1, F_2^1, F_3^1\}$  is the set of membership functions of the input variable  $\lambda(k)$ ,  $F_{n_\epsilon}^2 \in \{F_1^2, F_2^2, F_3^2\}$  is the set of membership functions of the input variable  $\Delta\epsilon(k)$  and  $F_{n'_\lambda}^3 \in \{F_1^3, F_2^3, F_3^3\}$  is the set of membership functions of  $\lambda(k+1)$ .

The rules for the output  $\lambda(k+1)$  are defined in table 1.

**Table 1.** Rules for the variation of the forgetting factor.

$\Delta\epsilon(k) \lambda(k)$	Small	Mean	Great
Negative	Small	Mean	Great
Null	Mean	Great	Great
Positive	Small	Small	Mean

The inference method is based on the max-min and the defuzzification is the centre of gravity.

$$\mu(z) = \max(\min(\min(\mu_{F_{n_\lambda}^1}(v), \mu_{F_{n_\epsilon}^2}(v))), \mu_{F_{n'_\lambda}^3}(z)) \tag{30}$$

With  $n_\lambda=1$  to 3,  $n_\epsilon= 1$  to 3 and  $n'_\lambda= 1$  to 3.

$$\lambda(k+1) = \frac{\int \mu(z)zdz}{\int \mu(z)dz} \tag{31}$$

The number of forgetting factors is equal to the number of model outputs.

### 3 Generation of Residues and Decision-Making

#### 3.1 Classical Method

The residuals are analytical redundancy generated measurements representing the difference between the observed and the expected system behaviour. When a fault occurs, the residual signal allows to evaluate the difference with the normal operating conditions (Fig. 5).

The residuals are processed and examined under certain decision rules to determine the change of the system status. Thus, the fault is detected, isolated (to distinguish the abnormal behaviours and determine the faulty component) and identified (to characterize the duration of the default and the amplitude in order to deduce its severity) (Fig. 6).

A threshold between the outputs of the system and the estimated outputs is chosen in order to proceed to the decision-making.

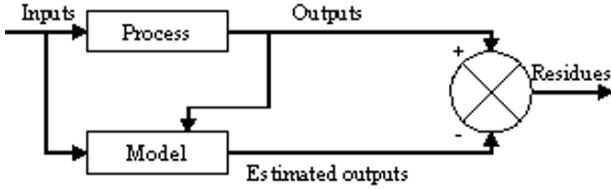


Fig. 5. Generation of residues.

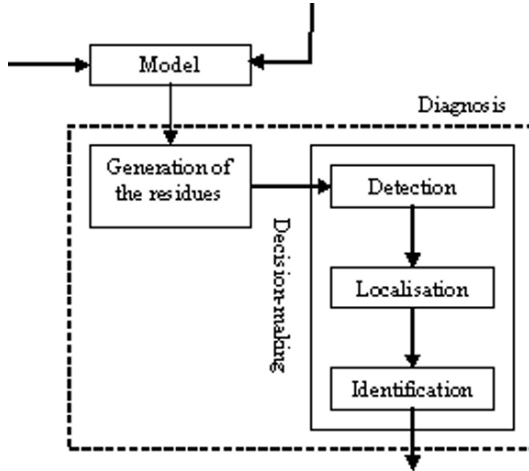


Fig. 6. Decision-making.

The residues  $r_j = |(y_j - \hat{y}_j)|$  are calculated to estimate the case where there is no failure and the case of sensor default. A threshold  $t$  is taken: if  $r_j \leq t$  then  $r_j = 0$ . At each instant  $k$ , the different  $r_j$  are checked in order to establish a diagnosis.

### 3.2 Approach with Fuzzy Lambda

Our method uses the fuzzy lambda to detect and isolate a default on a sensor. For the MIMO system, the algorithm generates one lambda for each output.

Let  $\lambda_j$ , with  $j = 1$  to  $n$ ,  $n$ : number of outputs.

The residues  $r'_j = 1 - \lambda_j$  are calculated to estimate the case where there is no failure and the case of sensor default. A threshold  $t'$  is taken: if  $r'_j \leq t'$  then  $r'_j = 0$ .

At each instant  $k$ , the different  $r'_j$  are checked in order to establish a diagnosis as shown in table 2.

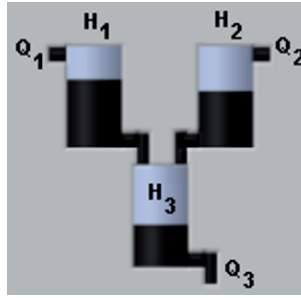
**Table 2.** Analysis of residues.

Analysis of residues	Diagnosis
$\forall j, r'_j = 0$	<i>No failure</i>
<i>If <math>r'_j \neq 0</math></i>	<i>Sensor default j</i>

## 4 Application

### 4.1 Benchmark Example: A Hydraulic Process [23]

The approach proposed previously has been validated on a benchmark: a hydraulic process. This system is a hydraulic process composed of three tanks (Fig. 7). The objective of the regulation is to be able to have a constant volume of the fluid. The three tanks have the same section:  $S$ .

**Fig. 7.** A hydraulic process.

The physical model of this system is obtained with the difference between the entering and outgoing flows which make evolve the level of each tank.

The state model is described by:

$$\begin{aligned} \dot{X} &= AX + BU \\ Y &= CX + DU \end{aligned} \quad (32)$$

$$X = [h_1 \ h_2 \ h_3]^T, U = [q_1 \ q_2]^T \text{ and } Y = X \quad (33)$$

The vector of outputs is the same as the state vector and, thus, the observation matrix  $C$  is an identity matrix with a size  $3 \times 3$ . This system, considered as linear around a running point, has been identified in using an ARX structure. The discrete model is obtained by using a sample period equal to 0.68 seconds.

The model describes the dynamical behaviour of the system in terms of inputs/outputs variations around the running point  $(U_0 \ Y_0)$ .

$$U_0 = (0.8 \ 1)^T \quad Y_0 = (400 \ 300 \ 200)^T$$

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) + D_d u(k) + n_o(k) \end{aligned} \quad (34)$$

The sensors noise  $no(k)$  considered is a normal distribution with mean zero and variance one.

This system is completely observable and controllable.

A quadratic linear control, associated to an integrator, enables to calculate the feedback gain matrix  $K$  from the minimization of the following cost function:

$$J = \frac{1}{2} \sum_{k=0}^N (x^T(k)Qx(k) + u^T(k)Ru(k)) \quad (35)$$

$$u(k) = -Kx(k) \quad (36)$$

As shown in section 2, for each output, a forgetting factor is assigned.  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  vary independently in function of the error between the process outputs and the model outputs.

For this application, the values  $\eta_{min}$  and  $\eta_{max}$ , described in section 2.2, are respectively 1.25 and 10. The model is adapted to follow the process behaviour.

## 4.2 Results

Two types of defaults have been tested: the bias and the calibration default. The simulation of the bias default has been carried out by subtracting a constant value  $\beta$  from the real value: for example  $h_{1_{real}} = h_{1_{sensor}} - \beta$ .

The simulation of the calibration default is obtained by multiplying the real value by a coefficient  $\gamma$ : for example  $h_{1_{real}} = h_{1_{sensor}} * \gamma$ .

For each type of default, we present two cases of simulated default: - a fault simulated in a sensor, - a same fault simulated in two or three sensors at same time.

The environment of the supervision enables to see the good detection of defaults. As soon as a failure is detected, the algorithm stops and indicates which sensor has a default (Fig. 8).

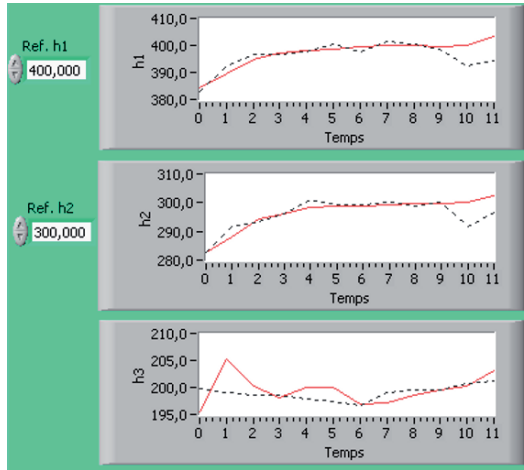
The physical model is represented by the dotted line curve and the parametric model by the solid line curve. For this example, the default is simulated, at sample 10, on the sensor  $h_1$ . The algorithm has detected the default at sample 12.

**Case 1.** For these first experiments, we simulated a sensor default by type of default. We have simulated the classical method and our approach with the bias default and the calibration default for the three sensors ( $\lambda_1, \lambda_2, \lambda_3$ ). To compare these two methods, we vary the values  $\beta$  and  $\gamma$ .

In table 3 and table 4, we show the performances of the two methods. For this, we define a rate which is the percentage of detection on 100 tests.

We can note that the fuzzy method gives better results. Indeed, when the default is weak ( $\beta < 7$  or  $\gamma > 0.97$ ), the rate of detection is more important.

On the other hand, the results are similar. To improve the detection with the classical method, the threshold  $t$  could be decreased but that implies an important rate of false alarm. Indeed, if the threshold is weaker than the importance of the noise, the algorithm stops in an inopportune way.



**Fig. 8.** Supervision.

**Table 3.** Rate of detection for the bias default.

<i>Bias default</i>		$\beta$					
		4	5	6	7	8	
<i>Classical method</i>	$h_1$	6	42	88	98	100	
	$h_2$	6	42	80	100	100	
<i>Threshold <math>t = 5.5</math></i>		$h_3$	22	42	80	90	100
<i>Fuzzy method</i>	$h_1$	64	84	100	100	100	
	$h_2$	76	92	98	100	100	
<i>Threshold <math>t' = 0.1</math></i>		$h_3$	86	92	98	100	100

**Table 4.** Rate of detection for the calibration default.

<i>Calibration default</i>		$\gamma$				
		0.99	0.98	0.97	0.96	
<i>Classical method</i>	$h_1$	24	100	100	100	
	$h_2$	2	84	100	100	
<i>Threshold <math>t = 5.5</math></i>		$h_3$	0	8	80	98
<i>Fuzzy method</i>	$h_1$	70	100	100	100	
	$h_2$	48	100	100	100	
<i>Threshold <math>t' = 0.1</math></i>		$h_3$	36	68	92	100

**Case 2.** In this case, for each type of default, several faults are simulated in several sensors at same time. Table 5 shows the performances of the proposed method when  $h_1$  and  $h_2$ , or  $h_1$  and  $h_3$ , or  $h_2$  and  $h_3$ , or  $h_1$ ,  $h_2$  and  $h_3$  sensors present a default.

The rate of detection obtained is weaker than in the case 1, nevertheless, the FFF method allows to detect several defaults simulated at same time (principally when  $\beta \geq 8$  or  $\gamma \leq 0.96$ ).

**Table 5.** Rate of detection for several defaults for the fuzzy method.

<i>Sensor defaults</i>	$\beta$		$\gamma$	
<i>Threshold <math>t' = 0.1</math></i>	4	8	0.99	0.96
$h_1 h_2$	38	80	18	92
$h_1 h_3$	34	88	16	92
$h_2 h_3$	34	90	4	88
$h_1 h_2 h_3$	14	88	4	94

### 4.3 Sensitivity to the Measure Noise

The measure noise has a great significance on the fault detection. The presented values are the minimal values which the method can detect.

In the case where the measure noise is more important, these results can be upgraded by modifying the values  $\eta_{min}$  and  $\eta_{max}$  defined in section 2.2. If the measure noise is very large, it is necessary to increase these initial values. By doing that, a tolerance compared with the noise is admitted. A compromise should be found between the noise level and the variation of  $\eta_{min}$  and  $\eta_{max}$ . Indeed, the algorithm can detect a false alarm.

## 5 Conclusions

This paper presents an original method of model-based diagnosis with a fuzzy parametric approach. This method is applicable to all non-linear MIMO systems for which the knowledge of the physical model is not required. We define a Fuzzy Forgetting Factor which allows to improve the estimation of model parameters, and to detect and isolate several types of faults. Thus, the fuzzy adaptation of the forgetting factors is used to detect and isolate the sensor faults. The results are illustrated by a benchmark system (a hydraulic process) and comparisons between the classical method and the FFF method is depicted in table 3 and table 4. Moreover, the method has been evaluated for several sensor defaults and results presented in table 5.

In conclusion, the method is efficient to detect and isolate one or more defaults simultaneously. The proposed approach is able to detect faults which correspond to a bias and a calibration default for each sensor.

A possible extension would be to determine the values  $\eta_{min}$  and  $\eta_{max}$ , described in section 2.2, in an automatic way according to the sensor noise. Moreover, it would be interesting to develop the FFF method for the actuator defaults.

## References

1. Noura, H.: Methodes d'accommodation aux defaults: Theorie et application. Memoire d'Habilitation a Diriger des Recherches. University Henri Poincare, Nancy 1, (2002)
2. Szederkenyi, G.: Model-Based Fault Detection of Heat Exchangers. Department of Applied Computer Science. University of Veszprem, (1998)
3. Ripoll, P.: Conception d'un systeme de diagnostic flou appliqu au moteur automobile. Thesis. University of Savoie, (1999)

4. Maquin, D.: Diagnostic a base de modeles des systemes technologiques. Memoire d'Habilitation a Diriger des Recherches. Institut National Polytechnique de Lorraine, (1997)
5. Isermann, R.: Supervision, fault-detection and fault-diagnosis methods - Advanced methods and applications. In: Proc. of the IMEKO world congress, New Measurements - Challenges and Visions. Vol. 1. 4. Tampere, Finland (1997) 1–28
6. Isermann, R.: Model-based fault detection and diagnosis - Status and applications. In: Annual Reviews in Control. Vol. 28. 1. Elsevier Ltd. (2005) 71–85
7. Isermann, R.: Process fault detection based on modelling and estimation methods - A survey. In: Automatica. Vol. 20. 4. (1984) 387–404
8. Querelle, R., Mary, R., Kiupel, N., Frank, P.M.: Use of qualitative modelling and fuzzy clustering for fault diagnosis. In: Proc. of world Automation Congress WAC'96. Vol. 5. 4. Montpellier, France (1996) 527–532
9. Kroll, A.: Identification of functional fuzzy models using multidimensional reference fuzzy sets. In: Fuzzy Sets and Systems. Vol. 8. (1996) 149–158
10. Liu, G., Toncich, D.J., Harvey, E.C., Yuan, F.: Diagnostic technique for laser micromachining of multi-layer thin films. Int. J. of Machine Tools and Manufacture. **45** (2005) 583–589
11. Evsukoff, A., Gentil, S., Montmain, J.: Fuzzy reasoning in co-operative supervision systems. In: Control Engineering Practice. 8. (2000) 389–407
12. Carrasco, E.F., Rodriguez, J., Punal, A., Roca, E., Lema, J.M.: Diagnosis of acidification states in an anaerobic wastewater treatment plant using a fuzzy-based expert system. In: Control Engineering Practice. 12. (2004) 59–64
13. Sala, A., Guerra, T.M., Babuska, R.: Perspectives of fuzzy systems and control. In: Fuzzy Sets and Systems. Vol. 156. (2005) 432–444
14. Lafont, F., Balmat, J.F., Taurines, M.: Fuzzy forgetting factor for system identification. In: Proc. of third International Conference on Systems, Signals and Devices. Systems analysis and automatic control. Vol. 1. Sousse, Tunisia (2005)
15. Lafont, F., Pessel, N., Balmat, J.F.: A fuzzy parametric approach for the model-based diagnosis. In: Proc. of fourth International Conference on Informatics in Control, Automation and Robotics. Angers, France (2007)
16. Fink, A., Fischer, M., Nelles, O.: Supervision of Non-linear Adaptive Controllers Based on Fuzzy Models. In: Control Engineering Practice. Vol. 8. 10. (2000) 1093–1105
17. Campi, M.: Performance of RLS Identification Algorithms with Forgetting Factor. A  $\phi$ -Mixing Approach. In: Journal of Mathematical Systems, Estimation, and Control. Vol. 4. 3. (1994) 1–25
18. Uhl, T.: Identification of modal parameters for non-stationary mechanical systems. In: Arch. Appl. Mech. 74. (2005) 878–889
19. Trabelsi, A., Lafont, F., Kamoun, M., Enea, G.: Identification of non-linear multi-variable systems by adaptive Fuzzy Takagi-Sugeno model. In: International Journal of Computational Cognition, ISBN 1542-5908. Vol. 2. 3. (2004) 137–153
20. Slama-Belkhodja, I., De Fornel, B.: Commande adaptative d'une machine asynchrone. In: J. Phys. III. Vol. 6. (1996) 779–796
21. Andersson, P.: Adaptive forgetting in recursive identification through multiple. In: J. Control. (1985) 1175–1193
22. Jager, R.: Fuzzy Logic in Control. Thesis, ISBN 90-9008318-9. Technische Universiteit Delft, (1995)
23. Jamouli, H.: Generation de residus directionnels pour le diagnostic des systemes lineaires stochastiques et la commande toleranteaux fautes. Thesis. University Henri Poincare, Nancy 1, (2003)

# The Shape of a Local Minimum and the Probability of its Detection in Random Search

Boris Kryzhanovsky, Vladimir Kryzhanovsky and Andrey Mikaelian

Center of Optical Neural Technologies, SR Institute of System Analysis RAS  
44/2 Vavilov Str, Moscow 119333, Russia  
kryzhanov@mail.ru

**Abstract.** The problem of binary optimization is discussed. By analyzing the generalized Hopfield model we obtain expressions describing the relationship between the depth of a local minimum and the size of the basin of attraction. The shape of local minima landscape is described. Based on this, we present the probability of finding a local minimum as a function of the depth of the minimum. Such a relation can be used in optimization applications: it allows one, basing on a series of already found minima, to estimate the probability of finding a deeper minimum, and to decide in favor of or against further running the program. It is shown, that the deepest minimum is defined with the greatest probability in random search. The theory is in a good agreement with experimental results.

**Keywords.** Binary optimization, neural networks, local minimum.

## 1 Introduction

Usually a neural system of associative memory is considered as a system performing a recognition or retrieval task. However it can also be considered as a system that solves an optimization problem: the network is expected to find a configuration minimizes an energy function [1]. This property of a neural network can be used to solve different *NP*-complete problems. A conventional approach consists in finding such an architecture and parameters of a neural network, at which the objective function or cost function represents the neural network energy. Successful application of neural networks to the traveling salesman problem [2] had initiated extensive investigations of neural network approaches for the graph bipartition problem [3], neural network optimization of the image processing [4] and many other applications. This subfield of the neural network theory is developing rapidly at the moment [5-11].

The aforementioned investigations have the same common feature: the overwhelming majority of neural network optimization algorithms contain the Hopfield model in their core, and the optimization process is reduced to finding the global minimum of some quadratic functional (the energy) constructed on a given  $N \times N$  matrix in an  $N$ -dimensional configuration space [12-13]. The standard neural network approach to such a problem consists in a random search of an optimal solution. The procedure consists of two stages. During the first stage the neural



network is initialized at random, and during the second stage the neural network relaxes into one of the possible stable states, i.e. it optimizes the energy value. Since the sought result is unknown and the search is done at random, the neural network is to be initialized many times in order to find as deep an energy minimum as possible. But the question about the reasonable number of such random starts and whether the result of the search can be regarded as successful always remains open.

In this paper we have obtained expressions that have demonstrated the relationship between the depth of a local minimum of energy and the size of the basin of attraction [14]. Based on this expressions, we presented the probability of finding a local minimum as a function of the depth of the minimum. Such a relation can be used in optimization applications: it allows one, based on a series of already found minima, to estimate the probability of finding a deeper minimum, and to decide in favor of or against further running of the program. Our expressions are obtained from the analysis of generalized Hopfield model, namely, of a neural network with Hebbian matrix. They are however valid for any matrices, because any kind of matrix can be represented as a Hebbian one, constructed on arbitrary number of patterns. A good agreement between our theory and experiment is obtained.

## 2 Description of the Model

Let us consider Hopfield model, i.e. a system of  $N$  Ising spins-neurons  $s_i = \pm 1$ ,  $i = 1, 2, \dots, N$ . A state of such a neural network can be characterized by a configuration  $\mathbf{S} = (s_1, s_2, \dots, s_N)$ . Here we consider a generalized model, in which the connection matrix:

$$T_{ij} = \sum_{m=1}^M r_m s_i^{(m)} s_j^{(m)}, \quad \sum r_m^2 = 1 \quad (1)$$

is constructed following Hebbian rule on  $M$  binary  $N$ -dimensional patterns  $\mathbf{S}_m = (s_1^{(m)}, s_2^{(m)}, \dots, s_N^{(m)})$ ,  $m = 1, \overline{M}$ . The diagonal matrix elements are equal to zero ( $T_{ii} = 0$ ). The generalization consists in the fact, that each pattern  $\mathbf{S}_m$  is added to the matrix  $T_{ij}$  with its statistical weight  $r_m$ . We normalize the statistical weights to simplify the expressions without loss of generality. Such a slight modification of the model turns out to be essential, since in contrast to the conventional model it allows one to describe a neural network with a non-degenerate spectrum of minima.

We have to note that all our expressions described below are obtained from the analysis of Hopfield model with generalized Hebbian matrix. Nevertheless they are valid for any random matrices, because it is proved by us [15] that any kind of symmetric matrix can be represented as a Hebbian one (1), constructed on arbitrary number of patterns.

The energy of the neural network is given by the expression:

$$E = -\frac{1}{2} \sum_{i,j=1}^N s_i T_{ij} s_j \quad (2)$$

and its (asynchronous) dynamics consist in the following. Let  $\mathcal{S}$  be an initial state of the network. Then the local field  $h_i = -\partial E / \partial s_i$ , which acts on a randomly chosen  $i$ -th spin, can be calculated, and the energy of the spin in this field  $\varepsilon_i = -s_i h_i$  can be determined. If the direction of the spin coincides with the direction of the local field ( $\varepsilon_i < 0$ ), then its state is stable, and in the subsequent moment ( $t+1$ ) its state will undergo no changes. In the opposite case ( $\varepsilon_i > 0$ ) the state of the spin is unstable and it flips along the direction of the local field, so that  $s_i(t+1) = -s_i(t)$  with the energy  $\varepsilon_i(t+1) < 0$ . Such a procedure is to be sequentially applied to all the spins of the neural network. Each spin flip is accompanied by a lowering of the neural network energy. It means that after a finite number of steps the network will relax to a stable state, which corresponds to a local energy minimum.

### 3 Basin of Attraction

Let us examine at which conditions the pattern  $\mathcal{S}_m$  embedded in the matrix (1) will be a stable point, at which the energy  $E$  of the system reaches its (local) minimum  $E_m$ . In order to obtain correct estimates we consider the asymptotic limit  $N \rightarrow \infty$ . We determine the basin of attraction of a pattern  $\mathcal{S}_m$  as a set of the points of  $N$ -dimensional space, from which the neural network relaxes into the configuration  $\mathcal{S}_m$ . Let us try to estimate the size of this basin. Let the initial state of the network  $\mathcal{S}$  be located in a vicinity of the pattern  $\mathcal{S}_m$ . Then the probability of the network convergence into the point  $\mathcal{S}_m$  is given by the expression:

$$\mathbf{Pr} = \left( \frac{1 + \mathbf{erf} \gamma}{2} \right)^N \quad (3)$$

where  $\mathbf{erf} \gamma$  is the error function of the variable  $\gamma$ :

$$\gamma = \frac{r_m \sqrt{N}}{\sqrt{2(1-r_m^2)}} \left( 1 - \frac{2n}{N} \right) \quad (4)$$

and  $n$  is Hamming distance between  $\mathcal{S}_m$  and  $\mathcal{S}$  ( $N - 2n = \mathcal{S}_m \mathcal{S}$ ). The expression (3) can be obtained with the help of the methods of probability theory, repeating the well-known calculation [16] for conventional Hopfield model.

It follows from (3) that the basin of attraction is determined as the set of the points of the configuration space close to  $\mathcal{S}_m$ , for which  $n \leq n_m$ :

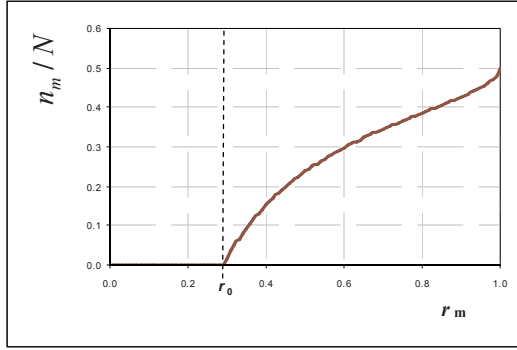
$$n_m = \frac{N}{2} \left( 1 - \frac{r_0 \sqrt{1-r_m^2}}{r_m \sqrt{1-r_0^2}} \right) \quad (5)$$

where

$$r_0 = \sqrt{2 \ln N / N} \quad (6)$$

Indeed, if  $n \leq n_m$  we have  $\mathbf{Pr} \rightarrow 1$  for  $N \rightarrow \infty$ , i.e. the probability of the convergence to the point  $\mathcal{S}_m$  asymptotically tends to 1. In the opposite case ( $n > n_m$ ) we have  $\mathbf{Pr} \rightarrow 0$ . It means that the quantity  $n_m$  can be considered as the radius of the basin of attraction of the local minimum  $E_m$ .

It follows from (5) that the radius of basin of attraction tends to zero when  $r_m \rightarrow r_0$  (Fig.1). It means that the patterns added to the matrix (1), whose statistical weight is smaller than  $r_0$ , simply do not form local minima. Local minima exist only in those points  $\mathcal{S}_m$ , whose statistical weight is relatively large:  $r_m > r_0$ .



**Fig. 1.** A typical dependence of the width of the basin of attraction  $n_m$  on the statistical weight of the pattern  $r_m$ . A local minimum exists only for those patterns, whose statistical weight is greater than  $r_0$ . For  $r_m \rightarrow r_0$  the size of the basin of attraction tends to zero, i.e. the patterns whose statistical weight  $r_m \leq r_0$  do not form local minima.

## 4 Shape of Local Minimum

From analysis of Eq. (2) it follows that the energy of a local minimum  $E_m$  can be represented in the form:

$$E_m = -r_m N^2 \quad (7)$$

with the accuracy up to an insignificant fluctuation of the order of

$$\sigma_m = N \sqrt{1 - r_m^2} \quad (8)$$

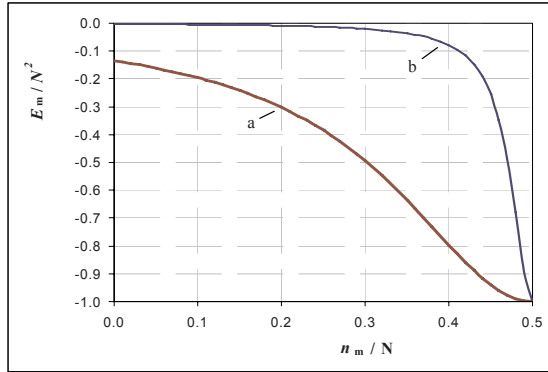
Then, taking into account Eqs. (5) and (7), one can easily obtain the following expression:

$$E_m = E_{\min} \frac{1}{\sqrt{(1 - 2n_m / N)^2 + E_{\min}^2 / E_{\max}^2}} \quad (9)$$

where

$$E_{\min} = -N \sqrt{2N \ln N}, \quad E_{\max} = \left( \sum_{m=1}^M E_m^2 \right)^{1/2} \quad (10)$$

which yield a relationship between the depth of the local minimum and the size of its basin of attraction. One can see that the wider the basin of attraction, the deeper the local minimum and vice versa: the deeper the minimum, the wider its basin of attraction (see Fig.2).



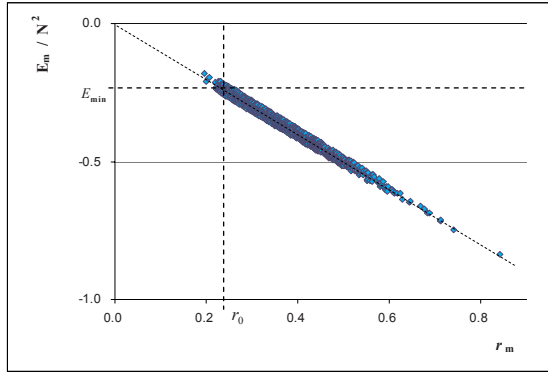
**Fig. 2.** The dependence of the energy of a local minimum on the size of the basin of attraction: a)  $N=50$ ; b)  $N=5000$ .

We have introduced here also a constant  $E_{max}$ , which we make use of in what follows. It denotes the maximal possible depth of a local minimum. In the adopted normalization, there is no special need to introduce this new notation, since it follows from (7)-(9) that  $E_{max} = -N^2$ . However for other normalizations some other dependencies of  $E_{max}$  on  $N$  are possible, which can lead to a misunderstanding.

The quantity  $E_{min}$  introduced in (10) characterizes simultaneously two parameters of the neural network. First, it determines the half-width of the Lorentzian distribution (9). Second, it follows from (9) that:

$$E_{max} \leq E_m \leq E_{min} \tag{11}$$

i.e.  $E_{min}$  is the upper boundary of the local minimum spectrum and characterizes the minimal possible depth of the local minimum. These results are in a good agreement with the results of computer experiments aimed to check whether there is a local minimum at the point  $\mathcal{S}_m$  or not. The results of one of these experiments ( $N=500$ ,  $M=25$ ) are shown in Fig.3. One can see a good linear dependence of the energy of the local minimum on the value of the statistical weight of the pattern. Note that the overwhelming number of the experimental points corresponding to the local minima are situated in the right lower quadrant, where  $r_m > r_0$  and  $E_m < E_{min}$ . One can also see from Fig.3 that, in accordance with (8), the dispersion of the energies of the minima decreases with the increase of the statistical weight.



**Fig. 3.** The dependence of the energy  $E_m$  of a local minimum on the statistical weight  $r_m$  of the pattern.

Now let us describe the shape of local minimum landscape. Let  $\mathbf{S}_m^{(n)}$  is any point in  $n$ -vicinity of local minima  $\mathbf{S}_m$  ( $n$  is Hamming distance between  $\mathbf{S}_m^{(n)}$  and  $\mathbf{S}_m$ ). It follows from (2) that the energy  $E_m^{(n)}$  in the state  $\mathbf{S}_m^{(n)}$  can be described as

$$E_m^{(n)} = -r_m N^2 (1 - 2n/N)^2 + D = E_m (1 - 2n/N)^2 + D \quad (12)$$

where

$$D = 4 \sum_{\mu \neq m}^M \sum_{i=1}^n \sum_{j=n+1}^N r_\mu S_{mi} S_{\mu i} S_{mj} S_{mj} \quad (13)$$

is Gaussian random number with average 0 and a relatively small variance

$$\sigma_D = 4 \sqrt{n(N-n)(1-r_m^2)} \quad (14)$$

It follows from (12) that the average value of energy in  $n$ -vicinity, e.i. the shape of local minima, can be presented as

$$\langle E_m^{(n)} \rangle = E_m (1 - 2n/N)^2 \quad (15)$$

As we see all the minima have the same shape described by Eqs. (15) which is independent of matrix type. It can be proved in experiment by random matrix generating and randomly chosen minima shape investigation (see Fig.4). As we see the experiment is in a good agreement with theory.

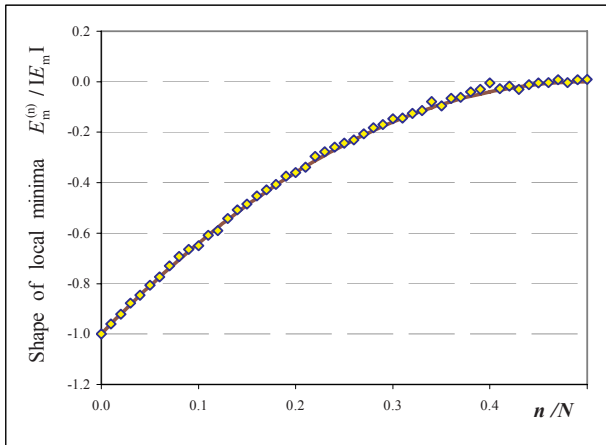


Fig. 4. The shape of randomly detected minima: curve – theory (15), marks – experiment.

### 5 The Probability of Finding the Minimum

Let us find the probability  $W$  of finding a local minimum  $E_m$  at a random search. By definition, this probability coincides with the probability for a randomly chosen initial configuration to get to the basin of attraction of the pattern  $S_m$ . Consequently, the quantity  $W = W(n_m)$  is the number of points in a sphere of a radius  $n_m$ , reduced to the total number of the points in the  $N$ -dimensional space:

$$W = 2^{-N} \sum_{n=1}^{n_m} \binom{N}{n} \tag{16}$$

Equations (5) and (16) define implicitly a connection between the depth of the local minimum and the probability of its finding. Applying asymptotical Stirling expansion to the binomial coefficients and passing from summation to integration one can represent (16) as

$$W = W_0 e^{-Nh} \tag{17}$$

where  $h$  is generalized Shannon function

$$h = \frac{n_m}{N} \ln \frac{n_m}{N} + \left(1 - \frac{n_m}{N}\right) \ln \left(1 - \frac{n_m}{N}\right) + \ln 2 \tag{18}$$

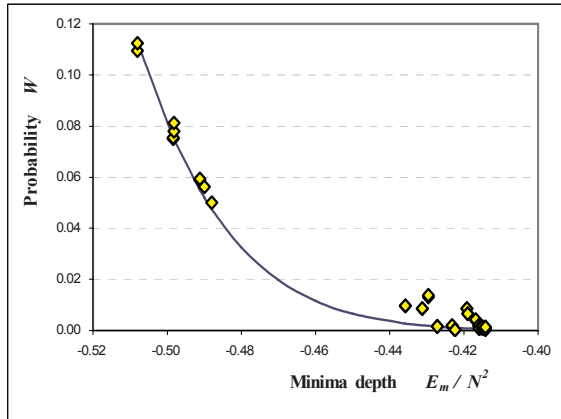
Here  $W_0$  is an insignificant for the further analysis slow function of  $E_m$ . It can be obtained from the asymptotic estimate (17) under the condition  $n_m \gg 1$ , and the dependence  $W = W(n_m)$  is determined completely by the fast exponent.

It follows from (18) that the probability of finding a local minimum of a small depth ( $E_m \sim E_{\min}$ ) is small and decreases as  $W \sim 2^{-N}$ . The probability  $W$  becomes

visibly non-zero only for deep enough minima  $|E_m| \gg |E_{\min}|$ , whose basin of attraction sizes are comparable with  $N/2$ . Taking into account (9), the expression (18) can be transformed in this case to a dependence  $W = W(E_m)$  given by

$$W = W_0 \exp \left[ -NE_{\min}^2 \left( \frac{1}{E_m^2} - \frac{1}{E_{\max}^2} \right) \right] \quad (19)$$

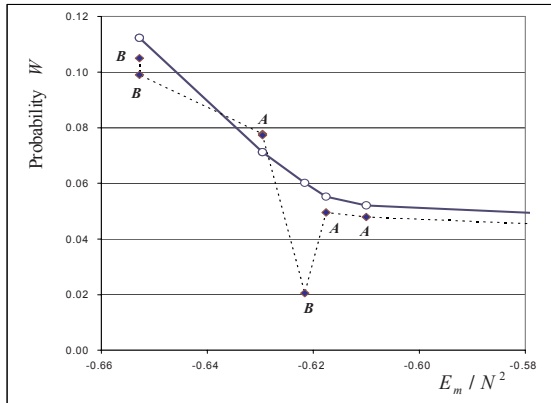
It follows from (18) that the probability to find a minimum increases with the increase of its depth. This dependence “the deeper minimum  $\rightarrow$  the larger the basin of attraction  $\rightarrow$  the larger the probability to get to this minimum” is confirmed by the results of numerous experiments. In Fig.5 the solid line is computed from Eq. (17), and the points correspond to the experiment (Hebbian matrix with a small loading parameter  $M/N \leq 0.1$ ). One can see that a good agreement is achieved first of all for the deepest minima, which correspond to the patterns  $\mathcal{S}_m$  (the energy interval  $E_m \leq -0.49N^2$  in Fig.5). The experimentally found minima of small depth (the points in the region  $E_m > -0.44N^2$ ) are the so-called “chimeras”. In standard Hopfield model ( $r_m \equiv 1/\sqrt{M}$ ) they appear at relatively large loading parameter  $M/N > 0.05$ . In the more general case, which we consider here, they can appear also earlier. The reasons leading to their appearance are well examined with the help of the methods of statistical physics in [17], where it was shown that the chimeras appear as a consequence of interference of the minima of  $\mathcal{S}_m$ . At a small loading parameter the chimeras are separated from the minima of  $\mathcal{S}_m$  by an energy gap clearly seen in Fig.5.



**Fig. 5.** The dependence of the probability  $W$  to find a local minimum on its depth  $E_m$ : theory - solid line, experiment - points.

## 6 Discussion

Our analysis shows that the properties of the generalized model are described by three parameters  $r_0$ ,  $E_{\min}$  and  $E_{\max}$ . The first determines the minimal value of the statistical weight at which the pattern forms a local minimum. The second and third parameters are accordingly the minimal and the maximal depth of the local minima. It is important that these parameters are independent from the number of embedded patterns  $M$ .

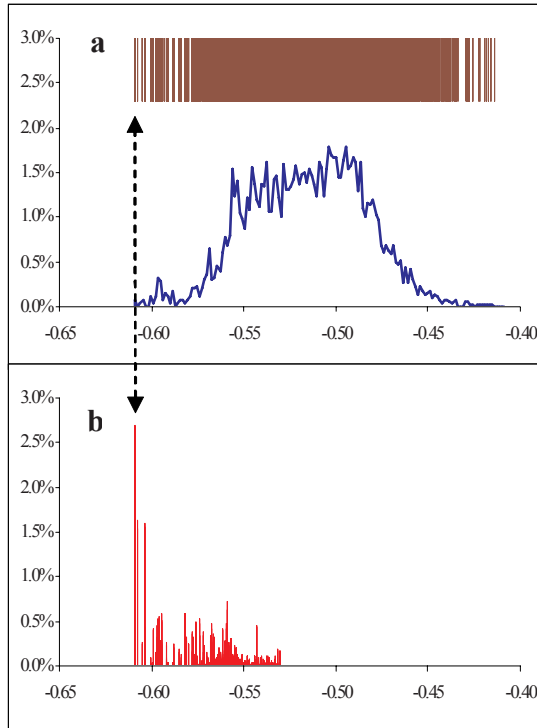


**Fig. 6.** The comparison of the predicted probabilities (solid line) and the experimentally found values (points connected with the dashed line).

Now we are able to formulate a heuristic approach of finding the global minimum of the functional (2) for any given matrix (not necessarily Hebbian one). The idea is to use the expression (19) with unknown parameters  $W_0$ ,  $E_{\min}$  and  $E_{\max}$ . To do this one starts the procedure of the random search and finds some minima. Using the obtained data, one determines typical values of  $E_{\min}$  and  $E_{\max}$  and the fitting parameter  $W_0$  for the given matrix. Substituting these values into (19) one can estimate the probability of finding an unknown deeper minimum  $E_m$  (if it exists) and decide in favor or against (if the estimate is a pessimistic one) the further running of the program.

This approach was tested with Hebbian matrices at relatively large values of the loading parameter ( $M / N \geq 0.2 \div 10$ ). The result of one of the experiments is shown in Fig.6. In this experiment with the aid of the found minima (the points  $A$ ) the parameters  $W_0$ ,  $E_{\min}$  and  $E_{\max}$  were calculated, and the dependence  $W = W(E_m)$  (solid line) was found. After repeating the procedure of the random search over and over again ( $\sim 10^5$  random starts) other minima (points  $B$ ) and the precise probabilities of getting into them were found. One can see that although some dispersion is present, the predicted values in the order of magnitude are in a good agreement with the precise probabilities.





**Fig. 7.** The case of matrix with a quasi-continuous type of spectrum. a) The upper part of the figure shows the spectrum of minima distribution – each vertical line corresponds to a particular minimum. The solid line denotes the spectral density of minima (the number of minima at length  $\Delta E$ ). The Y-axis presents spectral density and the X-axis is the normalized values of energy minima  $E/N^2$ . b) Probability of finding a minimum with energy  $E$ . The Y-axis is the probability of finding a particular minimum (%) and the X-axis is the normalized values of energy minima.

In conclusion we stress once again that any given symmetric matrix can be performed in the form of Hebbian matrix (1) constructed on an arbitrary number of patterns (for instance,  $M \rightarrow \infty$ ) with arbitrary statistical weights. It means that the dependence “the deeper minimum  $\leftrightarrow$  the larger the basin of attraction  $\leftrightarrow$  the larger the probability to get to this minimum” as well as all other results obtained in this paper are valid for all kinds of matrices. To prove this dependence, we have generated random matrices, with uniformly distributed elements on  $[-1,1]$  segment. The results of a local minima search on one of such matrices are shown in Fig. 7. The value of normalized energy is shown on the X-scale and on the Y-scale the spectral density is noted. As we can see, there are a lot of local minima, and most of them concentrated in central part of spectrum (Fig 7.a). Despite of such a complex view of the spectrum of minima, the deepest minimum is found with maximum probability (Fig 7.b). The same perfect accordance of the theory and the experimental results are also obtained in the case of random matrices, the elements of which are subjected to the Gaussian distribution with a zero mean.

The work supported by RFBR grant # 06-01-00109.

## References

1. Hopfield, J.J.: Neural Networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. USA.* v.79, pp.2554-2558 (1982)
2. Hopfield, J.J., Tank, D.W.: Neural computation of decisions in optimization problems. *Biological Cybernetics*, v.52, pp.141-152 (1985)
3. Fu, Y., Anderson, P.W.: Application of statistical mechanics to NP-complete problems in combinatorial optimization. *Journal of Physics A.*, v.19, pp.1605-1620 (1986)
4. Poggio, T., Girosi, F.: Regularization algorithms for learning that are equivalent to multilayer networks. *Science* 247, pp.978-982 (1990)
5. Smith, K.A.: Neural Networks for Combinatorial Optimization: A Review of More Than a Decade of Research. *INFORMS Journal on Computing* v.11 (1), pp.15-34 (1999)
6. Hartmann, A.K., Rieger, H.: *New Optimization Algorithms in Physics*, Wiley-VCH, Berlin (2004)
7. Huajin Tang; Tan, K.C.; Zhang Yi: A Columnar Competitive Model for Solving Combinatorial optimization problems. *IEEE Trans. Neural Networks* v.15, pp.1568–1574 (2004)
8. Kwok, T., Smith, K.A.: A noisy self-organizing neural network with bifurcation dynamics for combinatorial optimization. *IEEE Trans. Neural Networks* v.15, pp.84 – 98 (2004)
9. Salcedo-Sanz, S.; Santiago-Mozos, R.; Bousono-Calzon, C.: A hybrid Hopfield network-simulated annealing approach for frequency assignment in satellite communications systems. *IEEE Trans. Systems, Man and Cybernetics*, v. 34, 1108 – 1116 (2004)
10. Wang, L.P., Li, S., Tian F.Y, Fu, X.J.: A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing. *IEEE Trans. System, Man, Cybern, Part B - Cybernetics* v.34, pp. 2119-2125 (2004)
11. Wang, L.P., Shi, H.: A gradual noisy chaotic neural network for solving the broadcast scheduling problem in packet radio networks. *IEEE Trans. Neural Networks*, vol.17, pp.989 – 1000 (2006)
12. Joya, G., Atencia, M., Sandoval, F.: Hopfield Neural Networks for Optimization: Study of the Different Dynamics. *Neurocomputing*, v.43, pp. 219-237 (2002)
13. Kryzhanovsky, B., Magomedov, B.: Application of domain neural network to optimization tasks. *Proc. of ICANN'2005. Warsaw. LNCS 3697, Part II*, pp.397-403 (2005)
14. Kryzhanovsky, B., Magomedov, B., Fonarev, A.: On the Probability of Finding Local Minima in Optimization Problems. *Proc. of International Joint Conf. on Neural Networks IJCNN-2006 Vancouver*, pp.5882-5887 (2006)
15. Kryzhanovsky, B.V.: Expansion of a matrix in terms of external products of configuration vectors. *Optical Memory & Neural Networks*, v. 17, No.1, pp. 17-26 (2008)
16. Perez-Vincente, C.J.: Finite capacity of sparse-coding model. *Europhys. Lett*, v.10, pp. 627-631 (1989)
17. Amit, D.J., Gutfreund, H., Sompolinsky, H.: Spin-glass models of neural networks. *Physical Review A*, v.32, pp.1007-1018 (1985)

# Rapidly-exploring Sorted Random Tree: A Self Adaptive Random Motion Planning Algorithm

Nicolas Jouandeau

Université Paris 8, LIASD, 2, rue de la Liberté, 93526 Saint-Denis Cedex, France  
n@ai.univ-paris8.fr  
<http://www.ai.univ-paris8.fr/~n/>

**Abstract.** We present a novel single shot random algorithm, named *RSRT*, for *Rapidly-exploring Sorted Random Tree* and based on inherent relations analysis between *RRT* components. Experimental results are realized with a wide set of path planning problems involving a free flying object in a static environment. The results show that our *RSRT* algorithm is faster than existing ones. These results can also stand as a starting point of a massive motion planning benchmark.

## 1 Motion Planning with Rapidly Exploring Random Trees

The problem of motion planning turns out to be solved only by high computational systems due to its inherent complexity [1]. As the main goal of the discipline is to develop practical and efficient solvers that produce automatically motions, random sampling searches successfully reduce the determinist-polynomial complexity of the resolution [2]. In compensation, the resolution that consists in exploring the space, produce non-determinist solution [3]. Principal alternatives of this search are realized in configuration space  $C$  [4], in state space  $X$  [5] and in state-time space  $ST$  [6].  $C$  is intended to motion planning in static environments.  $X$  adds differential constraints.  $ST$  adds the possibility of a dynamic environment. The concept of high-dimensional configuration spaces is initiated by J. Barraquand *et al.* [7] to use a manipulator with 31 degrees of freedom. P. Cheng [8] uses these methods with a 12 dimensional state space involving rotating rigid objects in 3D space. S. M. LaValle [9] presents such a space with a hundred dimensions for a robot manipulator or a couple of mobiles. S.M. LaValle [10] is based on the construction of a tree  $T$  in the considered space  $\mathcal{S}$ . Starting from the initial position  $q_{init}$ , the construction of the tree is carried out by integrating control commands iteratively. Each iteration aims at bringing closer the mobile  $\mathcal{M}$  to an element  $e$  randomly selected in  $\mathcal{S}$ . To avoid cycles, two elements  $e$  of  $T$  cannot be identical. In practice, *RRT* is used to solve various problems such as negotiating narrow passages made of obstacles [11], finding motions that satisfy obstacle-avoidance and dynamic balances constraints [12], making Mars exploration vehicles strategies [13], searching hidden objects [14], rallying a set of points or playing hide-and-seek with another mobile [15] and many others mentioned in [9]. Thus the *RRT* method can be considered as the most general one by their efficiency to solve a large set of problems.

In its initial formulation, *RRT* algorithms are defined without goal. The exploration tree covers the surrounding space and progress blindly towards free space. A geometrical path planning problem aims generally at joining a final configuration  $q_{obj}$ . To solve the path planning problem, the *RRT* method searches a solution by building a tree (Alg. 1) rooted at the initial configuration  $q_{init}$ . Each node of the tree results from the mobile constraints integration. Its edges are commands that are applied to move the mobile from a configuration to another.

**Algorithm 1:** Basic *RRT* building algorithm.

```

rrt( $q_{init}, k, \Delta t, C$ )
1  init( $q_{init}, T$ );
2  for  $i \leftarrow 1$  to  $k$ 
3       $q_{rand} \leftarrow \text{randomState}(C)$ ;
4       $q_{prox} \leftarrow \text{nearbyState}(q_{rand}, T)$ ;
5       $q_{new} \leftarrow \text{newState}(q_{prox}, q_{rand}, \Delta t)$ ;
6      addState( $q_{new}, T$ );
7      addLink( $q_{prox}, q_{new}, T$ );
8  return  $T$ ;

```

The *RRT* method is a random incremental search which could be casting in the same framework of Las Vegas Algorithms (*LVA*). It repeats successively a loop made of three phases: generating a random configuration  $q_{rand}$ , selecting the nearest configuration  $q_{prox}$ , generating a new configuration  $q_{new}$  obtained by numerical integration over a fixed time step  $\Delta t$ . The mobile  $\mathcal{M}$  and its constraints are not explicitly specified. Therefore, modifications for additional constraints (such as non-holonomic) are considered minor in the algorithm formulation.

In this first version,  $C$  is presented without obstacle in an arbitrary space dimension. At each iteration, a local planner is used to connect each couples ( $q_{new}, q_{prox}$ ) in  $C$ . The distance between two configurations in  $T$  is defined by the time-step  $\Delta t$ . The local planner is composed by temporal and geometrical integration constraints. The resulting solution accuracy is mainly due to the chosen local planner.  $k$  defines the maximum depth of the search. If no solution is found after  $k$  iterations, the search can be restarted with the previous  $T$  without re-executing the init function (Alg. 1 line 1).

The *RRT* method, inspired by traditional Artificial Intelligent techniques for finding sequences between an initial and a final element (*i.e.*  $q_{init}$  and  $q_{obj}$ ) in a well-known environment, can become a bidirectional search (shortened *Bi-RRT* [16]). Its principle is based on the simultaneous construction of two trees (called  $T_{init}$  and  $T_{obj}$ ) in which the first grows from  $q_{init}$  and the second from  $q_{obj}$ . The two trees are developed towards each other while no connection is established between them. This bidirectional search is justified because the meeting configuration of the two trees is nearly the half-course of the configuration space separating  $q_{init}$  and  $q_{obj}$ . Therefore, the resolution time complexity is reduced [17].

*RRT-Connect* [18] is a variation of *Bi-RRT* that consequently increase the *Bi-RRT* convergence towards a solution thanks to the enhancement of the two trees convergence. This has been settled to:

- ensure a fast resolution for “simple” problems (in a space without obstacle, the *RRT* growth should be faster than in a space with many obstacles);
- maintain the probabilistic convergence property. Using heuristics modify the probability convergence towards the goal and also should modify its evolving distribution. Modifying the random sampling can create local minima that could slow down the algorithm convergence.

**Algorithm 2:** Connecting a configuration  $q$  to a graph  $T$  with *RRT-Connect*.

```

connectT( $q, \Delta t, T$ )
1   $r \leftarrow$  ADVANCED;
2  while  $r =$  ADVANCED
3     $|r \leftarrow$  expandT( $q, \Delta t, T$ );
4  return  $r$ ;

```

In *RRT-Connect*, the two graphs previously called  $T_{init}$  and  $T_{obj}$  are called now  $T_a$  and  $T_b$  (Alg. 3).  $T_a$  (respectively  $T_b$ ) replaces  $T_{init}$  and  $T_{obj}$  alternatively (respectively  $T_{obj}$  and  $T_{init}$ ). The main contribution of *RRT-Connect* is the ConnectT function which move towards the same configuration as long as possible (*i.e.* without collision). As the incremental nature algorithm is reduced, this variation is designed for non-differential constraints. This is iteratively realized by the expansion function (Alg. 2). A connection is defined as a succession of successful extensions. An expansion towards a configuration  $q$  becomes either an extension or a connection. After connecting successfully  $q_{new}$  to  $T_a$ , the algorithm tries as many extensions as possible towards  $q_{new}$  to  $T_b$ . The configuration  $q_{new}$  becomes the convergence configuration  $q_{co}$  ( Alg. 3 lines 8 and 10). Inherent relations inside the adequate construction of  $T$  in  $C_{free}$  shown in previous works are:

- the deviation of random sampling in the variations *Bi-RRT* and *RRT-Connect*. Variations include in *RRT-Connect* are called *RRT-ExtCon*, *RRT-ConCon* and *RRT-ExtExt*; they modify the construction strategy of one of the two trees of the method *RRT-Connect* by changing priorities of the extension and connection phases [19].
- the well-adapted  $q_{prox}$  element selected according to its collision probability in the variation *CVP* and the integration of collision detection since  $q_{prox}$  generation [20].
- the adaptation of  $C$  to the vicinity accessibility of  $q_{prox}$  in the variation *RC-RRT* [21].
- the parallel execution of growing operations for  $n$  distinct graphs in the variation *OR parallel Bi-RRT* and the growing of a shared graph with a parallel  $q_{new}$  sampling in the variation *embarrassingly parallel Bi-RRT* [22].
- the sampling adaptation to the *RRT* growth [23–27].

**Algorithm 3:** Expanding two graphs  $T_a$  and  $T_b$  towards themselves with *RRT-Connect*.  $q_{new}$  mentioned line 10 corresponds to the  $q_{new}$  variable mentioned line 9 Alg. 4.

```

rrtConnect( $q_{init}, q_{obj}, k, \Delta t, C$ )
1  init( $q_{init}, T_a$ );
2  init( $q_{obj}, T_b$ );
3  for  $i \leftarrow 1$  to  $k$ 
4       $q_{rand} \leftarrow \text{randomState}(C)$ ;
5       $r \leftarrow \text{expandT}(q_{rand}, \Delta t, T_a)$ ;
6      if  $r \neq \text{TRAPPED}$ 
7          if  $r = \text{REACHED}$ 
8               $q_{co} \leftarrow q_{rand}$ ;
9          else
10              $q_{co} \leftarrow q_{new}$ ;
11             if connectT( $q_{co}, \Delta t, T_b$ ) =
                REACHED
12                 sol  $\leftarrow \text{plan}(q_{co}, T_a, T_b)$ ;
13                 return sol;
14             swapT( $T_a, T_b$ );
15 return TRAPPED;

```

**Algorithm 4:** Expanding  $T$  with obstacles.

```

expandT( $q, \Delta t, T$ )
1   $q_{prox} \leftarrow \text{nearbyState}(q, T)$ ;
2   $d_{min} \leftarrow \rho(q_{prox}, q)$ ;
3   $success \leftarrow \text{FALSE}$ ;
4  foreach  $u \in U$ 
5       $q_{tmp} \leftarrow \text{integrate}(q, u, \Delta t)$ ;
6      if isCollisionFree( $q_{tmp}, q_{prox}, \mathcal{M}, C$ )
7           $d \leftarrow \rho(q_{tmp}, q_{rand})$ ;
8          if  $d < d_{min}$ 
9               $q_{new} \leftarrow q_{tmp}$ ;
10              $d_{min} \leftarrow d$ ;
11              $success \leftarrow \text{TRUE}$ ;
12 if  $success = \text{TRUE}$ 
13     insertState( $q_{prox}, q_{new}, T$ );
14     if  $q_{new} = q$ 
15         return REACHED;
14     return ADVANCED;
17 return TRAPPED;

```

By adding the collision detection in the given space  $S$  during the expansion phase, the selection of nearest neighbor  $q_{prox}$  is realized in  $S \cap C_{free}$  (Alg. 4). Although the

collision detection is expensive in computing time, the distance metric evaluation  $\rho$  is subordinate to the collision detector.  $U$  defines the set of admissible orders available to the mobile  $\mathcal{M}$ . For each expansion, the function `expandT` (Alg. 4) returns three possible values: REACHED if the configuration  $q_{new}$  is connected to  $T$ , ADVANCED if  $q$  is only an extension of  $q_{new}$  which is not connected to  $T$ , and TRAPPED if  $q$  cannot accept any successor configuration  $q_{new}$ .

In the next section, we examine in detail some justifications of our algorithm and the inherent relations in the various components used. This study enables to synthesize a new algorithm named Rapidly exploring Sorted Random Tree (*RSRT*), based on reducing collision detector calls without modification of the classical random sampling strategy.

## 2 RSRT Algorithm

Variations of *RRT* method presented in the previous section is based on the following sequence :

- generating  $q_{rand}$ ;
- selecting  $q_{prox}$  in  $T$ ;
- generating each successor of  $q_{prox}$  defined in  $U$ .
- realizing a colliding test for each successor previously defined;
- selecting a configuration called  $q_{new}$  that is the closest to  $q_{rand}$  among successors previously defined; This selected configuration has to be collision free.

The construction of  $T$  corresponds to the repetition of such a sequence. The collision detection discriminates the two possible results of each sequence:

- the insertion of  $q_{new}$  in  $T$  (*i.e.* without obstacle along the path between  $q_{prox}$  and  $q_{new}$ );
- the rejection of each  $q_{prox}$  successors (*i.e.* due to the presence of at least one obstacle along each successors path rooted at  $q_{prox}$ ).

The rejection of  $q_{new}$  induces an expansion probability related to its vicinity (and then also to  $q_{prox}$  vicinity); the more the configuration  $q_{prox}$  is close to obstacles, the more its expansion probability is weak. It reminds one of fundamentals *RRT* paradigm: free spaces are made of configurations that admit various number of available successors; good configurations admit many successors and bad configurations admit only few ones. Therefore, the more good configurations are inserted in  $T$ , the better the *RRT* expansion will be. The problem is that we do not previously know which good and bad configurations are needed during *RRT* construction, because the solution of the considered problem is not yet known. This problem is also underlined by the parallel variation *OR Bi-RRT* [22] (*i.e.* to define the depth of a search in a specific vicinity). For a path planning problem  $p$  with a solution  $s$  available after  $n$  integrations starting from  $q_{init}$ , the question is to maximize the probability of finding a solution; According to the concept of “rational action”, the response of *P3* class to adapt a on-line search can be solved by the definition of a formula that defines the cost of the search in terms of

“local effects” and “propagations” [28]. These problems find a way in the tuning of the behavior algorithm like *CVP* did [20].

**Algorithm 5:** Expanding  $T$  and reducing the collision detection.

```

newExpandT( $q, \Delta t, T$ )
1   $q_{prox} \leftarrow \text{nearbyState}(q, T)$ ;
2   $S \leftarrow \emptyset$ ;
3  foreach  $u \in U$ 
4       $q \leftarrow \text{integrate}(q_{prox}, u, \Delta t)$ ;
5       $d \leftarrow \rho(q, q_{rand})$ ;
6       $S \leftarrow S + \{(q, d)\}$ ;
7  qsort( $S, d$ );
8   $n \leftarrow 0$ ;
10 while  $n < \text{Card}(S)$ 
11      $s \leftarrow \text{getTupleIn}(n, S)$ ;
12      $q_{new} \leftarrow \text{firstElementOf}(s)$ ;
13     if isCollisionFree( $q_{new}, q_{prox}, \mathcal{M}, C$ )
14         insertState( $q_{prox}, q_{new}, T$ );
15         if  $q_{new} = q$ 
16             return REACHED;
17         return ADVANCED;
18      $n \leftarrow n + 1$ ;
19 return TRAPPED;

```

In the case of a space made of a single narrow passage, the use of bad configurations (which successors generally collide) is necessary to resolve such problem. The weak probability of such configurations extension is one of the weakness of the *RRT* method.

To bypass this weakness, we propose to reduce research from the closest element (Alg. 4) to the first free element of  $C_{free}$ . This is realized by reversing the relation between collision detection and distance metric; the solution of each iteration is validated by subordinating collision tests to the distance metric; the first success call to the collision detector validates a solution. This inversion induces:

- a reduction of the number of calls to the collision detector proportionally to the nature and the dimension of  $U$ ; Its goal is to connect the collision detector and the derivative function that produce each  $q_{prox}$  successor.
- an equiprobability expansion of each node independently of their relationship with obstacles;

The  $T$  construction is now based on the following sequence:

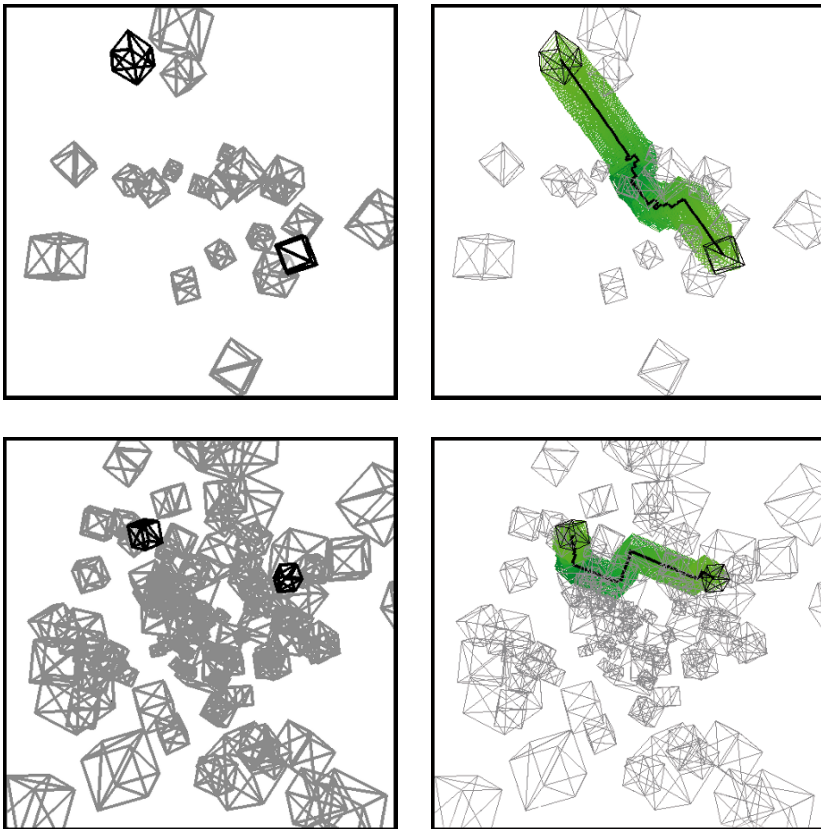
1. generating a random configuration  $q_{rand}$  in  $C$ ;
2. selecting  $q_{prox}$  the nearest configuration to  $q_{rand}$  in  $T$  (Alg. 5 line 1);
3. generating each successors of  $q_{prox}$  (Alg. 5 lines 3 to 6); each successor is associated with its distance metric from  $q_{rand}$ . It produces a couple called  $s$  stored in  $S$ ;



4. sorting  $s$  elements by distance ( Alg. 5 lines 7);
5. selecting the first collision-free element of  $S$  and breaking the loop as soon as this first element is discovered (Alg. 5 lines 16 and 17);

### 3 Experiments

This section presents experiments performed on a Redhat Linux Cluster that consists of 8 Dual Core processor 2.8 GHz Pentium 4 (5583 bogomips) with 512 MB DDR Ram.



**Fig. 1.** 20 obstacles problem and its solution (upper couple). 100 obstacles problem and its solution (lower couple).

To perform the run-time behavior analysis for our algorithm, we have generated series of problems that gradually contains more 3D-obstacles. For each problem, we have randomly generated ten different instances. The number of obstacles is defined by the sequence 20, 40, 60,  $\dots$ , 200, 220. In each instance, all obstacles are cubes and their sizes are randomly varying between (5, 5, 5) and (20, 20, 20). The mobile is

a cube with a fixed size (10, 10, 10). Obstacles and mobile coordinates are varying between  $(-100, -100, -100)$  and  $(100, 100, 100)$ . For each instance, a set of 120  $q_{init}$  and 120  $q_{obj}$  are generated in  $C_{free}$ . By combining each  $q_{init}$  and each  $q_{obj}$ , 14400 configuration-tuples are available for each instance of each problem. For all that, our benchmark is made of more than 1.5 million problems. An instance with 20 obstacles is shown in Fig. 1 on the lower part and another instance with 100 obstacles in Fig. 1 on the left part. On these two examples,  $q_{init}$  and  $q_{obj}$  are also visible. We used the Proximity Query Package (*PQP*) library presented in [29] to perform the collision detection. The mobile is a free-flying object controlled by a discretized command that contains 25 different inputs uniformly dispatched over translations and rotations. The performance was compared between *RRT-Connect* (using the *RRT-ExtCon* strategy) and our *RSRT* algorithm (Alg. 5).

The choice of the distance metric implies important consequences on configurations' connexity in  $C_{free}$ . It defines the next convergence node  $q_{co}$  for the local planner. The metric distance must be selected according to the behavior of the local planner to limit its failures. The local planner chosen is the straight line in  $C$ . To validate the toughness of our algorithm regarding to *RRT-Connect*, we had use three different distance metrics. Used distance metrics are:

- the Euclidean distance (mentioned *Eucl* in Fig. 2 to 4)

$$d(q, q') = \left( \sum_{k=0}^i (c_k - c'_k)^2 + nf^2 \sum_{k=0}^j (\alpha_k - \alpha'_k)^2 \right)^{\frac{1}{2}}$$

where  $nf$  is the normalization factor that is equal to the maximum of  $c_k$  range values.

- the scaled Euclidean distance metric (mentioned *Eucl2* in Fig. 2 to 4)

$$d(q, q') = \left( s \sum_{k=0}^i (c_k - c'_k)^2 + nf^2 (1 - s) \sum_{k=0}^j (\alpha_k - \alpha'_k)^2 \right)^{\frac{1}{2}}$$

where  $s$  is a fixed value 0.9;

- the Manhattan distance metric (mentioned *Manh* in Fig. 2 to 4)

$$d(q, q') = \sum_{k=0}^i \|c_k - c'_k\| + nf \sum_{k=0}^j \|\alpha_k - \alpha'_k\|$$

where  $c_k$  are axis coordinates and  $\alpha_k$  are angular coordinates.

For each instance, we compute the first thousand successful trials to establish average resolving times (Fig. 2), standard deviation resolving times (Fig. 3) and midpoint resolving times (Fig. 4). These trials are initiated with a fixed random set of seed. Those fixed seed assume that tested random suite are different between each other and are the same between instances of all problems. As each instance is associated to one thousand trials, each point of each graph is the average over ten instances (and then over ten thousands trials). On each graph, the number of obstacles is on x-axis and resolving time in *sec.* is on y-axis.

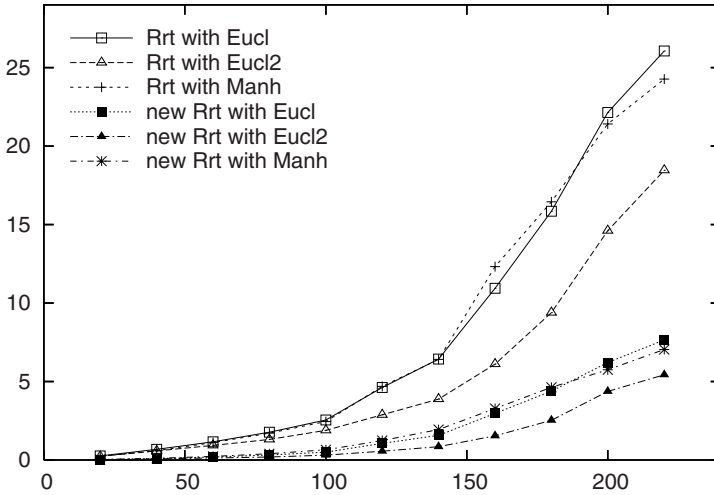


Fig. 2. Averages resolving times.

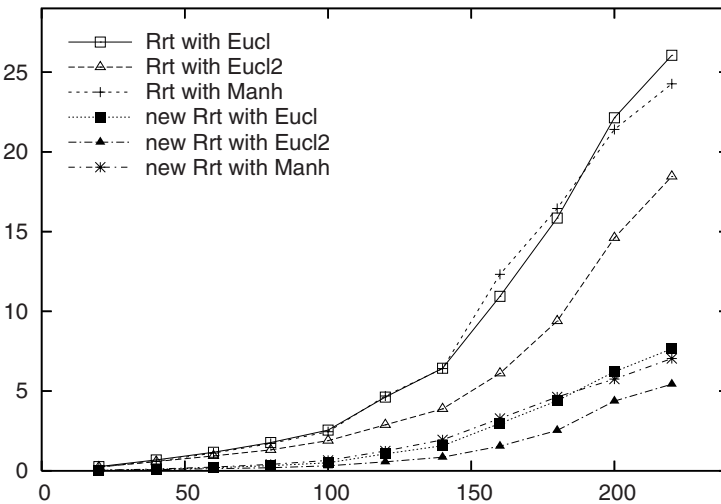


Fig. 3. Standard deviation resolving times.

Figure 2 shows that average resolving time of our algorithm oscillates between 10 and 4 times faster than the original *RRT-Connect* algorithm. As the space obstruction grows linearly, the resolving time of *RRT-Connect* grows exponentially while *RSRT* algorithm grows linearly. Figure 3 shows that the standard deviation follows the same profile. It shows that *RSRT* algorithm is more robust than *RRT-Connect*. Figure 4 shows that midpoints' distributions follow the average resolving time behavior. This is a reinforcement of the success of the *RSRT* algorithm. This assumes that half part of time distribution are 10 to 4 times faster than *RRT-Connect*.

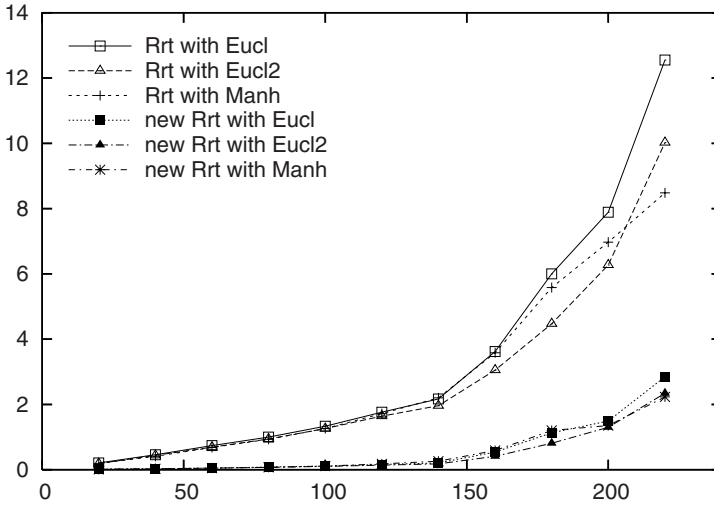


Fig. 4. Midpoint resolving times.

## 4 Conclusions

We have described a new *RRT* algorithm, called *RSRT* algorithm, to solve motion planning problems in static environments. *RSRT* algorithm accelerates consequently the resulting resolving time. The experiments show the practical performances of the *RSRT* algorithm, and the results reflect its classical behavior. The results given above (have been evaluated on a cluster which provide a massive experiment analysis. The challenging goal is now to extend the benchmark that is proposed to every motion planning method. The proposed benchmark will be enhanced to specific situations that allow *RRT* to deal with motion planning strategies based on statistical analysis.

## References

1. Canny, J.: The complexity of robot motion planning. PhD thesis, Massachusetts Institute of Technology. Artificial Intelligence Laboratory. (1987)
2. Schwartz, J., Sharir, M.: On the piano movers problem:I, II, III, IV, V. Technical report, New York University, Courant Institute, Department of Computer Sciences (1983)
3. Latombe, J.: Robot Motion Planning (4th edition). Kluwer Academic (1991)
4. Lozano-Prez, T.: Spatial Planning: A Configuration Space Approach. In: Trans. on Computers. (1983)
5. Donald, B., Xavier, P., Canny, J., Reif, J.: Kinodynamic Motion Planning. Journal of the ACM (1993)
6. Fraichard, T.: Dynamic trajectory planning with dynamic constraints: a "state-time space" approach. In: Int. Conf. Robotics and Automation (ICRA'93). (1993)
7. Barraquand, J., Latombe, J.: A Monte-Carlo Algorithm for Path Planning with many degrees of Freedom. In: Int. Conf. on Robotics and Automation (ICRA'90). (1990)

8. Cheng, P.: Reducing rrt metric sensitivity for motion planning with differential constraints. Master's thesis, Iowa State University (2001)
9. LaValle, S.: Planning Algorithms. [on-line book] (2004)  
<http://msl.cs.uiuc.edu/planning/>.
10. LaValle, S.: Rapidly-exploring random trees: A new tool for path planning. Technical Report 98-11, Dept. of Computer Science, Iowa State University (1998)
11. Ferr, E., Laumond, J.: An iterative diffusion algorithm for part disassembly. In: Int. Conf. Robotics and Automation (ICRA'04). (2004)
12. Kuffner, J., Nishiwaki, K., Kagami, S., Inaba, M., Inoue, H.: Motion planning for humanoid robots. In: Int'l Symp. Robotics Research (ISRR'03). (2003)
13. Williams, B.C., B.C., Kim, P., Hofbaur, M., How, J., Kennell, J., Loy, J., Ragno, R., Stedl, J., Walcott, A.: Model-based reactive programming of cooperative vehicles for mars exploration. In: Int. Symp. on Artificial Intelligence, Robotics and Automation in Space. 2001
14. Tovar, B., LaValle, S., Murrieta, R.: Optimal navigation and object finding without geometric maps or localization. In: Int. Conf. on Robotics and Automation (ICRA'03). (2003)
15. Simov, B., LaValle, S., Slutzki, G.: A complete pursuit-evasion algorithm for two pursuers using beam detection. In: Int. Conf. on Robotics and Automation (ICRA'02). (2002)
16. LaValle, S., Kuffner, J.: Randomized kinodynamic planning. In: Int. Conf. on Robotics and Automation (ICRA'99). (1999)
17. Russell, S., Norvig, P.: Artificial Intelligence, A Modern Approach (2me dition). Prentice Hall (2003)
18. Kuffner, J., LaValle, S.: RRT-Connect: An efficient approach to single-query path planning. In: Int. Conf. on Robotics and Automation (ICRA'00). (2000)
19. LaValle, S., Kuffner, J.: Rapidly-exploring random trees: Progress and prospects. In: Workshop on the Algorithmic Foundations of Robotics (WAFR'00). (2000)
20. Cheng, P., LaValle, S.: Reducing Metric Sensitivity in Randomized Trajectory Design. In: Int. Conf. on Intelligent Robots and Systems (IROS'01). (2001)
21. Cheng, P., LaValle, S.: Resolution Complete Rapidly-Exploring Random Trees. In: Int. Conf. on Robotics and Automation (ICRA'02). (2002)
22. Carpin, S., Pagello, E.: On Parallel RRTs for Multi-robot Systems. In: 8th Conf. of the Italian Association for Artificial Intelligence (AI\*IA'02). (2002)
23. Jouandeau, N., Chrif, A.A.: Fast Approximation to gaussian random sampling for randomized motion planning. In: Int. Symp. on Intelligent Autonomous Vehicules (IAV'04). (2004)
24. Corts, J., Simon, T.: Sampling-based motion planning under kinematic loop-closure constraints. In: Workshop on the Algorithmic Foundations of Robotics (WAFR'04). (2004)
25. Lindemann, S.R., LaValle, S.M.: Current issues in sampling-based motion planning. In: Int. Symp. on Robotics Research (ISRR'03). (2003)
26. Lindemann, S., LaValle, S.: Incrementally reducing dispersion by increasing Voronoi bias in RRTs. In: Int. Conf. on Robotics and Automation (ICRA'04). (2004)
27. Yershova, A., Jaillet, L., Simeon, T., LaValle, S.M.: Dynamic-domain rrts: Efficient exploration by controlling the sampling domain. In: Int. Conf. on Robotics and Automation (ICRA'05). (2005)
28. Russell, S.: Rationality and Intelligence. In Press, O.U., ed.: Common sense, reasoning, and rationality. (2002)
29. Gottschalk, S., Lin, M., Manocha, D.: Obb-tree: A hierarchical structure for rapid interference detection. In: Proc. of ACM Siggraph'96. (1996)

# Applying an Intensification Strategy on Vehicle Routing Problem

Etiene P. L. Simas and Arthur Tórigo Gómez

Universidade do Vale do Rio dos Sinos  
Programa Interdisciplinar de Pós Graduação em Computação Aplicada  
Av. Unisinos, 950, São Leopoldo, RS, Brazil  
{etiene.simas@terra.com.br, breno@unisinos.br}

**Abstract.** In this paper we propose a Tabu Search algorithm to solve the Vehicle Routing Problem. The Vehicle Routing Problem is usually defined as the problem that concerns in creation of least cost routs to serve a set of clients by a fleet of vehicles. We develop an intensification strategy to diversify the neighbors generated and to increase the neighborhood size. We had done experiments using and not using the intensification strategy to compare the performance of the search. The experiments we had done showed that an intensification strategy allows an increase on the solutions quality.

**Keywords.** Vehicle Routing Problem; Tabu Search; Intensification Strategy.

## 1 Introduction

The Vehicle Routing Problem (VRP) that is a NP-Hard problem [1] is usually dealt within the logistic context [2],[3]. It can be described as a set of customers that have to be served by a fleet of vehicles, satisfying some constraints [4],[3]. The transport is one of the most costly activities in logistic, typically varying in one or two thirds of the total costs [5]. Therefore, the necessity of improving the efficiency of this activity has great importance. A small percentage saved with this activity could result in a substantial saving total [6]. There many variants and constraints that can be considered, i.e. it that can be considered the fleet may be heterogeneous, the vehicles must execute collections and deliveries, there may exist more than one depot, etc. In this paper we are dealing with the classic version of this problem, were just the vehicle capacity constraint are considered.

## 2 The Vehicle Routing Problem

A classical definition is presented in Barbarasoglu and Ozgur [7]. The VRP is defined in a complete, undirected graph  $G=(V,A)$  where a fleet of  $N_v$  vehicle of homogeneous capacity is located. All remaining vertices are customers to be served. A non-negative matrix  $C=(c_{ij})$  is defined on  $A$  representing the distance between the vertices. The costs are the same in both directions. A non-negative demand  $d_i$  is associated with

each vertex representing the customer demand at  $v_i$ . The routes must start and finish at the depot. The clients must be visited just once, by only one vehicle and the total demand of the route can't exceed the capacity  $Q_v$  of the vehicle. In some cases, there is a limitation on the total route duration. In this case,  $t_{ij}$  represents the travel time for each  $(v_i, v_j)$ ,  $t_i$  represents the service time at vertex  $v_i$  and is required that the total time duration of any route should not exceed  $T_v$ . A typical formulation based on [7] is used in this paper:

$$\text{Minimize } \sum_i \sum_j \sum_v c_{ij} X_{ij}^v \cdot \tag{1}$$

$$\sum_i \sum_v X_{ij}^v = 1 \text{ for all } j. \tag{2}$$

$$\sum_j \sum_v X_{ij}^v = 1 \text{ for all } i. \tag{3}$$

$$\sum_i X_{ip}^v - \sum_j X_{pj}^v = 0 \text{ for all } p, v. \tag{4}$$

$$\sum_i d_i \left( \sum_j X_{ij}^v \right) \leq Q_v \text{ for all } v. \tag{5}$$

$$\sum_{j=1}^n X_{0j}^v \leq 1 \text{ for all } v. \tag{6}$$

$$\sum_{i=1}^n X_{i0}^v \leq 1 \text{ for all } v. \tag{7}$$

$$\sum_{i=1}^n X_{i0}^v \leq 1 \text{ for all } i, j \in v. \tag{8}$$

Where  $X_{ij}$  are binary variables indicating if  $\text{arc}(v_i, v_j)$  is transversed by vehicle  $v$ . The objective function of distance/cost/time is expressed by eq. (1). Constraints in eqs (2) and (3) together state that each demand vertex is served by exactly one vehicle. The eq. (4) guarantees that a vehicle leaves the demand vertex as soon as it has served the vertex. Vehicle capacity is expressed by (5) where  $Q_v$  is the capacity. Constraints (6) and (7) express that vehicle availability can't be exceeded. The subtour elimination constraints are given in eq.(8) where  $Z$  can be defined by:

$$Z = \left\{ (X_{ij}^v) : \sum_{i \in B} \sum_{j \in B} X_{ij}^v \leq |B| - 1 \text{ for } B \subseteq V \setminus \{0\}; |B| \geq 2 \right\} \tag{9}$$

### 3 Resolutions Methods

Since VRP is Np-Hard to obtain good solutions in an acceptable time, heuristics are used and this is the reason why the majority of researchers and scientists direct their

efforts in heuristics development [8],[9],[3]. Osman and Laporte [10] define heuristic as a technique, which seeks good solutions at a reasonable computational cost without being able to guarantee the optimality. Laporte et al [11] define two main groups of heuristics: classical heuristics, developed mostly between 1960 and 1990, and metaheuristics. The classical heuristics are divided in three groups: constructor methods, two-phase methods and improvement methods. Since 1990, the metaheuristics have been applied to the VRP problem. To Osman and Laporte [10] a metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space in order to find efficiently near-optimal solutions. Several metaheuristics have been proposed to solve the VRP problem. Among these ones, Tabu Search are considered the best metaheuristic for VRP. To review some works with Tabu Search and others metaheuristics some readings are suggested [12],[13].

### 3.1 Tabu Search

It was proposed by Glover [14] and had its concepts detailed in Glover and Laguna [15]. It's a technique to solve optimization combinatorial problems [14] that consists in an iterative routine to construct neighborhoods emphasizing the prohibition of stopping in an optimum local. The process that Tabu Search searches for the best solution is through an aggressive exploration [15], choosing the best movement for each iteration, not depending on if this movement improves or not the value of the actual solution. In Tabu Search development, intensification and diversification strategies are alternated through the tabu attributes analysis. Diversification strategies direct the search to new regions, aiming to reach whole search space while the intensification strategies reinforce the search in the neighborhood of a solution historically good [15]. The stop criterion makes it possible to stop the search. It can be defined as the interaction where the best results were found or as the maximum number of iteration without an improvement in the value of the objective function. The tabu list is a structure that keeps some solution's attributes that are considered tabu. The objective of this list is to forbid the use of some solutions during some defined time.

## 4 VRP Application

The application developed is divided into three modules:

a) Net Generation Module: This module generates the nets that will be used in the application using vertices coordinates and demands given.

b) Initial Solution Module: This module generates the initial solutions of the nets. The initial solutions are created thought the use of an algorithm implementing the Nearest Insertion heuristic [16], [17].

c) Tabu Search Module: This module performs the tabu search algorithm. The Tabu Search elements that were used are now detailed. The stop criterion adopted is the maximum number of iterations without any improvement in the value of the objective value. The tabu list keeps all the routes and the cost of the solution, forbidden these routes to be used together during the tabu tenure defined. An elite



solution list is used to keep the best results that were found during the search. It was proposed an intensification strategy to be used every time when the search executes 15 iterations without an improvement in the objective function value. In this strategy we visit every solution that is in elite list generating a big neighborhood for each one. There were defined two movements to neighborhood generation. V1 that makes the exchange of vertices and V2 that makes the relocation of vertices. In V1, one route  $r1$  is selected and than one vertex of this route is chosen. We try to exchange this vertex with every vertex of all the other routes. The exchange is done if the addition of the two new demands doesn't exceed the vehicle's capacity of both routes. This procedure is done for every vertex of the route  $r1$ . To every exchange that is made, one neighbor is generated. In V2, we select one route and choose one vertex and then we try to reallocate it into all others routes, if it doesn't exceed the vehicle capacity of the route. When a vertex can be inserting into a route, we try to insert it into all possible positions inside this route. To every position that a vertex is inserted, one neighbor is generated.

When these movements are used in the search with intensification, they are called V1' and V2' because with intensification, not only one route is selected like in V1 and V2, but also all routes of the solution are chosen. Aiming increase the neighborhood size and the diversification between the solutions, we proposed to use the movements alone and together.

## 5 Computational Experience

The computational experiments were conducted on problems 1, 2, 3, 4 and 5 of Christofides Mingozzi and Toth [18]. These problems contain 50, 75, 100, 150 and 199 vertices and one depot respectively and they are frequently used in papers for tests purposes. The objective of the experiments was to compare the search process using and not using intensification strategy using the different movements proposed. There were proposed 9 values to Nbmax {100, 250, 500, 750, 1000, 1250, 1500, 1750, and 2000} and 6 values to Tabu List size {10, 25, 50, 75, 100, and 200}. For every problem, the experiments were divided into 6 groups according with the used movements. Table 1 shows these groups.

**Table 1.** Groups of experiments divided by movements.

<b>Search Mode</b>	<b>Used Movement</b>
Using Intensification	V1
Using Intensification	V2
Using Intensification	V1,V2
Not Using Intensification	V1 + V1'
Not Using Intensification	V2 + V2'
Not Using Intensification	V1V2 + V1'V2'

There were generated 54 experiments for each group combining all values proposed to Nbmax with all Tabu List size. So, for each problem there were generated 162 experiments using intensification and 162 experiments not using it. Two types of analyses were done. In one type it was evaluated the best result obtained for a fixed value of Nbmax used with all Tabu List size and in other type it was evaluated the

best results obtained for a fixed size of Tabu List used with all values proposed to Nbmax. Analyses had also been made comparing the best result found for each group of experiment, in this case comparing the quality of the different movements.

### 5.1 Analyzing the Nbmax Variation for each Tabu List Size

By analyzing the results in this perspective it will be evaluate the variation of the Nbmax for each Tabu List size. The objective is verified if big values of Nbmax can improved the quality of Tabu Search process. We create a “lower average” for the average from results obtained with Nbmax = 100 and Nbmax = 250 and an “upper average” for the average from results obtained with Nbmax = 1750 and 2000. For all problems, analyzing each one of the 6 groups of experiments done, the “upper average” was always lesser than the “lower average”, indicating that big values of Nbmax can improve the search quality. The experiments shown that an increase in Nbmax value can improve the search quality, by decreasing the results costs of the solutions.

Table 2 shows the number of best results that were found in each Nbmax value:

**Table 2.** Quantity and localization of the best results found for problems 1,2,3,4 and 5.

Search	500	750	1000	1250	1500	1750	2000
<b>Problem 1</b>							
Using Intens.	1	1	3	3	3	4	3
Not using Intens.	0	0	1	2	4	6	5
<b>Problem 2</b>							
Using Intens.	1	0	1	3	4	2	7
Not using Intens.	0	0	1	2	3	6	6
<b>Problem 3</b>							
Using Intens.	0	1	0	2	1	4	10
Not using Intens.	0	1	0	1	2	1	12
<b>Problem 4</b>							
Using Intens.	0	0	1	2	3	1	11
Not using Intens.	0	0	1	4	1	2	10
<b>Problem 5</b>							
Using Intens.	0	0	2	3	0	1	12
Not using Intens.	0	0	0	1	1	5	11

This table shown that most best results were found when the search used big values of Nbmax.

### 5.2 Analyzing the Tabu List Size Variation for each Nbmax Value

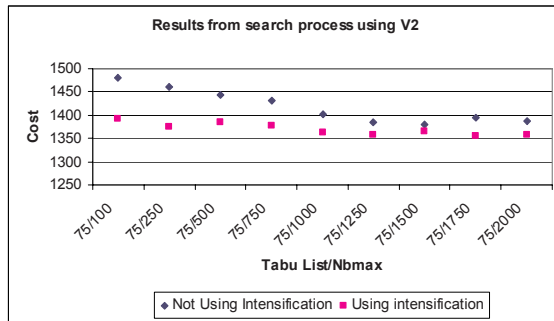
By analyzing the results in this perspective it will be evaluate the variation of the tabu list size for each Nbmax value. The objective is verified if big tabu list size can improved the quality of Tabu Search process. For problem 1, not using intensification, 66,66% of the results were found with Tabu list size  $\geq 75$ . Using intensification it was 48,14%. For problem 2 the percentage was 55,55% and 59,25%, not using and using intensification. For problem 3 the percentage was 77,77% and 96,29% not using and

using intensification. For problem 4 these percentage were 74,05% and 77,77% and for problem 5 they were 63,88% and 92,59%. So by analyzing these results we can see that big tabu list size can improve the quality of the search process.

### 5.3 Comparing the Search Process Using and not Using Intensification

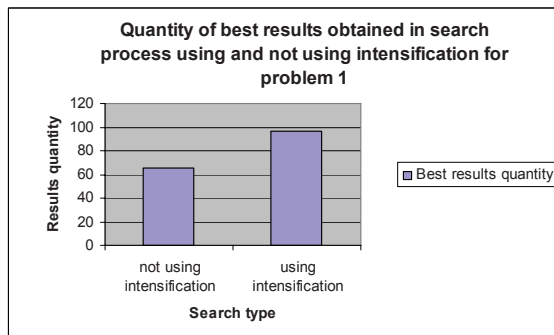
This analysis intends to compare the search process using and not using the intensification strategy to see if it can improve the results generated.

Figure 1 shows an example of the graphics done with the results obtained in both search process to compare the quality of the different search process.



**Fig. 1.** Costs obtained from both search process for Tabu List size = 75 and using V2 for problem 5.

This figure shows that an intensification strategy increase all results of the search process using V2 for problem 5. A comparison with the results generated by the search process using and not using intensification was done. Figures 2 to 6 shows the percentage of results that were improved with the intensification strategy. Figure 3 shows that for problem 1, from 162 results that were generated 97 were improved with intensification.



**Fig. 2.** Improve caused by the intensification search for problem 1.

For problem 2, from 162 results, 121 were improved using intensification strategy.

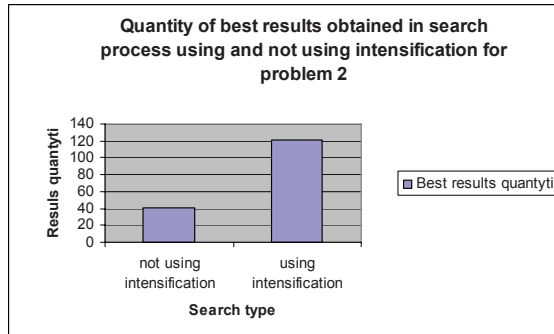


Fig. 3. Improve caused by the intensification search for problem 2.

For problem 3, from 162 results 102 were improved using intensification strategy.

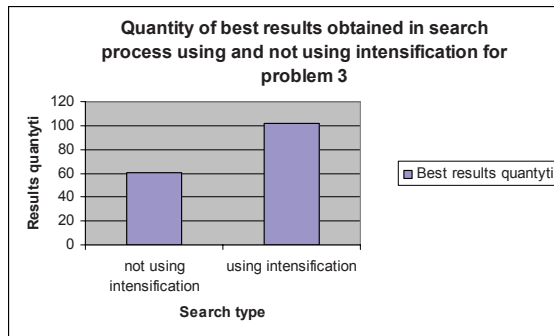


Fig. 4. Improve caused by the intensification search for problem 3.

For problem 4, from 162 results 117 were improved using intensification strategy.

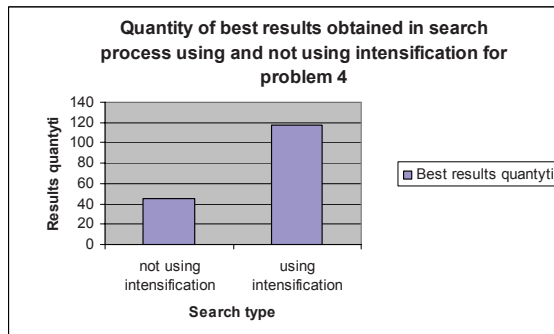


Fig. 5. Improve caused by the intensification search for problem 4.

For problem 5, from 162 results 135 were improved using intensification strategy.

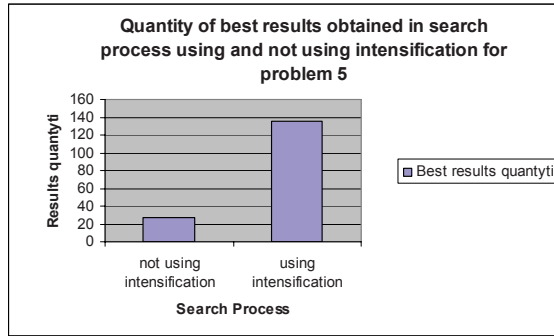


Fig. 6. Improve caused by the intensification search for problem 5.

The figures shows that an increase in solution quality of, at least, 50% happens when intensification strategy is used.

The average results and the standard deviation are shown for problem 1, 2, 3, 4 and 5 in tables 3 to 7.

Table 3. Average results and standard deviation for problem 1.

Problem 1				
	Average		Standard Deviation	
	Not using Intensif.	Using Intensif.	Not using Intensif.	Using Intensif.
V1	657,55	650,95	24,79	28,30
V2	582,08	570,30	21,16	22,21
V1,V2	537,36	542,01	8,56	13,62

Table 4. Average results and standard deviation for problem 2.

Problem 2				
	Average		Standard Deviation	
	Not using Intensif.	Using Intensif.	Not using Intensif.	Using Intensif.
V1	951,07	943,02	21,68	25,84
V2	895,75	883,03	20,67	21,71
V1,V2	867,96	863,06	13,00	14,70

Table 5. Average results and standard deviation for problem 3.

Problem 3				
	Average		Standard Deviation	
	Not using Intensif.	Using Intensif.	Not using Intensif.	Using Intensif.
V1	954,01	948,12	19,60	16,95
V2	903,63	901,17	14,50	17,84
V1,V2	879,90	870,10	15,05	20,73

**Table 6.** Average results and standard deviation for problem 4.

<b>Problem 4</b>				
	<b>Average</b>		<b>Standard Deviation</b>	
	<b>Not using Intensif.</b>	<b>Using Intensif.</b>	<b>Not using Intensif.</b>	<b>Using Intensif.</b>
V1	1215,35	1210,79	12,79	10,70
V2	1124,93	1118,83	15,66	12,52
V1,V2	1087,72	1079,54	18,15	16,70

**Table 7.** Average results and standard deviation for problem 5.

<b>Problem 5</b>				
	<b>Average</b>		<b>Standard Deviation</b>	
	<b>Not using Intensif.</b>	<b>Using Intensif.</b>	<b>Not using Intensif.</b>	<b>Using Intensif.</b>
V1	1569,11	1561,59	10,12	16,22
V2	1421,41	1393,56	34,40	12,75
V1,V2	1387,93	1377,82	26,64	16,37

**Table 8.** Best results obtained.

<b>Problem</b>	<b>Best Results</b>
1	525,42
2	847,82
3	837,79
4	1061,07
5	1352,74

From the results presented in tables 3 to 7 we can see that the results generated by the movements grouped are better than the results obtained using the movements alone. The reason for this is that when movements are used together the size of the neighborhood generated is bigger than the neighborhood generated by V1 or V2 alone. The movements together also cause an increase of the diversification of the solutions. And when the search generates more results, it is doing a deeper search in the space. Of course, as was shown, the intensification strategy helps the search to produce more qualified results.

When comparing the V1 and V2 movements, we can see that V2 produce results more qualified. If we analyze the policy behind the movement, we can say that V2 is more flexible than V1. V1 needs that two constraints are satisfied to generate one neighbor. While in V2, just one demand capacity must be verified (the capacity of the vehicle that serve the route where the vertex are being allocated) in V1, both routes must be verified to see if the vehicles capacities aren't exceeded.

Next table shows the best results obtained for each problem. All the best results were obtained during the search using movements V1 and V2 together and with the intensification strategy.

### 5.4 Comparisons

Aiming to evaluate the quality of the application developed, some papers were selected from the literature to compare the results. There were selected some classical heuristic and some papers that also used Tabu Search to solve the VRP. The paper selected were: {WL}Willard [19], {PF}Pureza and França [20], {OM1}Osman [21] , {OM2} Osman [22] , {RG} Rego [23], {GHL} Gendreau, Hertz and Laporte [24], {BO} Barbarasoglu and Ozgur [7], {XK} Xu and Kelly [3], {TV} Toth and Vigo [25], {CW} Clarke and Wright [26], {GM} Gillet and Miller [27] , {MJ} Mole and Jamenson [28] , {CMT} Christofides, Mingozzi and Toth[18].

Table 9 shows the comparison done with the results from the papers. The results were obtained in Barbarasolgu and Ozgur [7] and in Gendreau, Hertz and Laporte[24]. In the first columns the paper used is shown. Columns 2 and 4 present the best results from the paper and columns 3 and 5 shows the difference in percentage from the results obtained in this paper to the paper compared. This difference was called “gap”. The (+) indicate that our result is that percentage more than the result from the paper. The (-) indicate that our results is that percentage minor than the result from the paper.

**Table 9.** Best results and gap for problem 1 and 2.

	Problem 1		Problem 2	
	Best	%Gap	Best	%Gap
WL	588	11,91(-)	893	5,33(-)
RG	557,86	6,17(-)	847	0,10(+)
PF	536	2,10(-)	842	0,69(+)
OM1	524,61	0,15(+)	844	0,45(+)
OM2	524,61	0,15(+)	844	0,45(+)
GHL	524,61	0,15(+)	835,77	1,42(+)
BO	524,61	0,15(+)	836,71	1,31(+)
XK	524,61	0,15(+)	835,26	1,48(+)
TV	524,61	0,15(+)	838,60	1,09(+)
CW	578,56	10,11(-)	888,04	4,74(-)
GM	546	3,92(-)	865	2,03(-)
MJ	575	9,44(-)	910	7,33(-)
CMT	534	1,63(-)	871	2,73(-)

**Table 10.** Best results and gap for problem 3 and 4.

	Problem 3		Problem 4		Problem 5	
	Best	%Gap	Best	%Gap	Best	%Gap
WL	906	8,14(-)	-	-	-	-
RG	832,04	0,69(+)	1074,21	1,31(+)	1352,88	0,014(-)
PF	851	1,58(-)	1081	1,88(-)	-	-
OM1	835	0,33(+)	1052	0,85(+)	1354	0,09(-)
OM2	838	0,03(-)	1044,35	1,58(+)	1334,55	1,34(+)
GHL	829,45	1,00(+)	1036,16	2,35(+)	1322,65	2,22(+)
BO	828,72	1,08(+)	1043,89	1,62(+)	1306,16	3,44(+)
XK	826,14	1,39(+)	1029,56	2,97(+)	1298,58	4,00(+)
TV	828,56	1,10(+)	1028,42	3,08(+)	1291,45	4,53(+)
CW	878,70	4,88(-)	1204	13,47(-)	1540	13,84(-)
GM	862	2,89(-)	1079	1,69(-)	1389	2,68(-)
MJ	882	5,28(-)	1259	18,65(-)	1545	14,21(-)
CMT	851	1,58(-)	1093	3,01(-)	1418	4,82(-)

By analyzing these tables we can see that our application produce more qualified results than all the classical heuristics used in comparison because our result was better than all of the heuristic results. Comparing with other tabu search algorithm, we can say that our algorithm is very competitive. It dominates at least 2 results from the 9 used for each problem. Moreover, the results generated were less than 5% of the other results for all cases. And in 25 cases out of 45 this percentage is minor than 2%.

## 6 Final Considerations

In this paper it was proposed an application using Tabu Search to solve the vehicle routing problem. This application was divided into 3 modules: a net generation module, an initial solution module and tabu search module. We used two movements based in relocation of vertices and exchange of vertices to create the neighborhood. We use the movements alone and together, intending to diversify the solutions. We used an elite list solution to keep the best results found during the search. We propose an intensification strategy to use every time the search executes 15 iterations without improvement in objective value. We proposed some experiments to test if the solution quality increases or not with the increase in Nbmax value and in Tabu List size. We also compare the search process using and not using intensifications intending to see if this solution's quality is improved with the Intensification strategy. The experiments showed that big values to Nbmax and Tabu list size could improve the results. From the experiments we also can see that an intensification strategy can improve the quality of the search.

## References

1. Lenstra, J.K., Rinnoy K., G.: Complexity of Vehicle Routing and Scheduling Problems. *Networks* 11, 221-227 (1981)
2. Ho, S.C., Haugland, D.: A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers & Operations Research* 31, 1947-1964 (2004)
3. Xu, J., Kelly, James P.: A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem. *Transportation Science* 30, 379-393 (1996)
4. Laporte, G. The Vehicle Routing Problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59, 345-458 (1992)
5. Ballou, R.H. 2001 Gerenciamento da cadeia de Suprimentos – Planejamento, Organização Logística Empresarial, 4Ed, Porto Alegre: Bookman (2001)
6. Bodin, L.D, Golden, B.L., Assad, A.A., Ball, M.O.: Routing and Scheduling of vehicles and crews: The State of the Art. *Computers and Operations Research* 10, 69-211 (1983)
7. Barbarasoglu, G., Ozgur, D.: A tabu search algorithm for the vehicle routing problem. *Computers & Operations Research* 26, 255-270 (1999)
8. Thangiah, S.R., Petrovik, P. 1997 Introduction to Genetic Heuristics and vehicle Routing Problems with Complex Constraints. In: Woodruff, David, L. *Advances in Computational and Stochastic Optimization, Logic programming, and Heuristic search: Interfaces in Computer Science and Operations research*. Kluwer Academic Publishers. (1997)
9. Nelson, Marvin D; Nygard, Kendall E.; Griffin, John H.; Shreve, Warren E.: Implementing Techniques for the vehicle routing problem. *Computers & Operations Research* 12, 273-283 (1985)



10. Osman, I; Laporte,G.: Metaheuristics: A bibliography. *Annals of Operations Research* 63, 513-628 (1996)
11. Laporte,G., Gendreau,M., Potvin,J., Semet, F.: Classical and modern heuristics for the vehicle routing problem. *Intl.Trans. in Op. Res* 7, 285-300 (2000)
12. Cordeau, J-F., Gendreau, M., Laporte, G., Potvin, J.-Y., & Semet, F.: A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53, 512-522 (2002)
13. Tarantilis, C.D; Ioannou, G; Prastacos, G.: Advanced vehicle routing algorithms for operations management problems. *Journal of Food Engineering*, 70, 455-471 (2005)
14. Glover,F.: Tabu Search – parte 1. *ORSA Journal on Computing* v.1, n.3. (1989)
15. Glover,F.,Laguna,M. Tabu Search. Kluwer Academic Publishers. (1997)
16. Tyagi, M.: A Practical Method for the Truck Dispatching Problem. *Journal. of the Operations Research Society of Japan*, 10, 76-92 (1968)
17. Cook,W.J, Cunningham, W.H, Pulleyblank, W.R, Schrijver, A. *Combinatorial Optimization*. Willey (1998)
18. Christofides, N.; Mingozzi, A.; Toth, P.1979. The Vehicle Routing Problem. In: Christofides, Nicos. *Combinatorial Optimization*, UMI, (1979)
19. Willard, A.G.: Vehicle routing using r-optimal tabu search. MSc.Dissertation, The Management School, Imperial College, London (1989)
20. Pureza V.M., França, P.M.: Vehicle routing problems via tabu search metaheuristic, Publication CRT-747, Centre de recherché sur les transports, Montreal (1991)
21. Osman, I.H.: Simulated annealing and tabu search algorithms for combinatorial optimization problems. Ph.D.Dissertation, The Management School, Imperial College, London, 1991
22. Osman, I.H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41, 421-451 (1993)
23. Rego, C.: A Subpath Ejection Method for the Vehicle Routing Problem. *Management Science*, 44, 1447-1459 (1998)
24. Gendreau, M., Hertz, A.,Laporte, G.: A Tabu Search Heuristic for the Vehicle Routing Problem. *Management Science*, 40, 1276-1290 (1994)
25. Toth, P., Vigo, D.: Models, relaxations and exact approach for the capacitated vehicle routing problem. *Discrete Applied Mathematics* 123, 487-512, (2003)
26. Clarke, G, Wright, J.W. : Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12: 568-581 (1964)
27. Gillet, B.E, Miller, L.R.: A heuristic algorithm for the vehicle dispatch problem. *Operations Research* 22, 240-349 (1974)
28. Mole,R.H, Jamenson, S.R.: A sequential route-building algorithm employing a generalized savings criterion. *Operations Research Quarterly* 27, 503-511 (1976).

# Detection of Correct Moment to Model Update

Heli Koskimäki, Ilmari Juutilainen, Perttu Laurinen and Juha Röning

Intelligent Systems Group, University of Oulu  
PO BOX 4500, 90014 University of Oulu, Finland  
{heli.koskimaki, ilmari.juutilainen, perttu.laurinen,  
juha.roning}@ee.oulu.fi  
<http://www.ee.oulu.fi/research/isc>

**Abstract.** When new data are obtained or simply when time goes by, the prediction accuracy of models in use may decrease. However, the question is when prediction accuracy has dropped to a level where the model can be considered out of date and in need of updating. This article describes a method that was developed for detecting the need for a model update. The method was applied in the steel industry, and the models whose need of updating is under study are two regression models, a property model and a deviation model, developed to facilitate planning of optimal process settings by predicting the yield strength of steel plates beforehand. To decide on the need for updating, information from similar past cases was utilized by introducing a limit called an exception limit for both models. The limits were used to indicate when a new observation was from an area of the model input space where the results of the models are exceptional. Moreover, an additional limit was formed to indicate when too many exceedings of the exception limit have occurred within a certain time scale. These two limits were then used to decide when to update the model.

**Keywords.** Adaptive model update, similar past cases, error in neighborhood, process data.

## 1 Introduction

At Ruukki's steel works in Raahе, Finland, liquid steel is cast into steel slabs that are then rolled into steel plates. Many different variables and mechanisms affect the mechanical properties of the final steel plates. The desired specifications of the mechanical properties of the plates vary, and to fulfil the specifications, different treatments are required. Some of these treatments are complicated and expensive, so it is possible to optimize the process by predicting the mechanical properties beforehand on the basis of planned production settings [1].

Regression models have been developed for Ruukki to help development engineers control mechanical properties such as yield strength, tensile strength, and elongation of the metal plates [2]. Two different regression models are used for every mechanical property: a property model and a deviation model. The first one tells the predicted quality value, while the second one tells the actual working limits around this value. However, acquirement of new data and the passing of time decrease the reliability of

these models, which can result in economical losses to the plant. For example, when mechanical properties required by the customer are not satisfied in qualification tests, the testing lot in question need to be reproduced. If retesting also gives an unsatisfactory result, the whole order has to be produced again. Because of the volumes produced in a steel mill, this can cause huge losses. Thus, updating of the models emerges as an important step in improving modelling in the long run. This study is a follow-up to article [3], in which the need to update the regression model developed to model the yield strength of steel plates was studied from the point of view of the property model. However, in this article the deviation model is added to the study, making the solution more complete.

In practice, because the model is used in advance to plan process settings and since the employees know well how to produce common steel plate products, modelling of rare and new events becomes the most important aspect. However, to make new or rarely manufactured products, a reliable model is needed. Thus, when comparing the improvement in the model's performance, rare events are emphasized.

In this study model adaptation was approached by searching for the exact time when the performance of the model has decreased too much. In practice, model adaptation means retraining the model at optimally selected intervals. However, because the system has to adapt quickly to a new situation in order to avoid losses to the plant, periodic retraining, used in many methods ([4], [5]), is not considered the best approach. Moreover, there are also disadvantages if retraining is done unnecessarily. For example, extra work is needed to take a new model into use in the actual application environment. In the worst case, this can result in coding errors that affect the actual accuracy of the model.

Some other studies, for example [6], have considered model adaptation as the model's ability to learn behaviour in areas from which information has not been acquired. In this study, adaptation of the model is considered to be the ability to react to time-dependent changes in the modelled causality. In spite of extensive literature searches, studies comparable with the approach used in this article were not found. Thus, it can be assumed that the approach is new, at least in an actual industrial application.

## 2 Data Set and Regression Model

The data for this study were collected from Ruukki's steel works production database between July 2001 and April 2006. The whole data set consisted of approximately 250,000 observations. Information was gathered from element concentrations of actual ladle analyses, normalization indicators, rolling variables, steel plate thicknesses, and other process-related variables [7]. The observations were gathered during actual product manufacturing. The volumes of the products varied, but if there were more than 500 observations from one product, the product was considered a common product. Products with less than 50 observations were categorized as rare products.

In the studied prediction model, the response variable used in the regression modelling was the Box-Cox-transformed yield strength of the steel plates. The Box-Cox transformation was selected to produce a Gaussian-distributed error term. The deviation in yield strength also depended strongly on input variables. Thus, the studied prediction

model included separate link-linear models for both mean and variance

$$\begin{aligned} y_i &\sim N(\mu_i, \sigma_i^2) \\ \mu_i &= f(x_i' \beta) \\ \sigma_i &= g(z_i' \tau). \end{aligned} \quad (1)$$

The length of the parameter vector of mean model  $\beta$  was 130 and the length of the parameter vector of variance model  $\tau$  was 30. The input vectors  $x_i$  and  $z_i$  included 100 carefully chosen non-linear transformations of the 30 original input variables; for example, many of these transformations were products of two or three original inputs. The link functions  $f$  and  $g$  that were used were power transformations selected to maximize the fit with data. The results are presented in the original (nontransformed) scale of the response variable [2].

### 3 Neighborhood, *APEN* and *ASSRN*

In this study the need for a model update was approached using information from previous cases. To be precise, the update was based on the average prediction errors of similar past cases and on the average of the squared standardized residuals of deviations. Thus, for every new observation, a neighbourhood containing similar past cases was formed and an average prediction error and an average of squared standardized residuals inside the neighbourhood were calculated.

The neighbourhoods were defined using a Euclidian distance measure and the distance calculation was done only for previous observations to resemble the actual operating environment. The input variables were weighted using gradient-based scaling, so the weighting was relative to the importance of the variables in the regression model [8]. A numerical value of 3.5 was considered for the maximum distance inside which the neighbouring observations were selected. The value was selected using prior knowledge of the input variable values. Thus, a significant difference in certain variable values with the defined weighting resulted in Euclidean distances of over 3.5. In addition to this, the maximum count of the selected neighbours was restricted to 500.

After the neighbourhood for a new observation was defined, the average prediction error of the neighbourhood ( $= APEN$ ) was calculated as the distance-weighted mean of the prediction errors of observations belonging to the neighbourhood:

$$APEN = \frac{\sum_{i=1}^n [(1 - \frac{d_i}{max(d)}) \cdot \hat{\varepsilon}_i]}{\sum_{i=1}^n (1 - \frac{d_i}{max(d)})}, \quad (2)$$

where

- $n$  = number of observations in a neighbourhood,
- $\varepsilon(i)$  = the prediction error of the  $i$ th observation of the neighbourhood,
- $d_i$  = the Euclidian distance from the new observation to the  $i$ th observation of the neighbourhood,
- $max(d)$  = the maximum allowed Euclidian distance between the new observation and the previous observations in the neighbourhood ( $= 3.5$ ).

The average of the squared standardized residuals in the neighbourhood (= *ASSRN*) was achieved using:

$$ASSRN = \frac{\sum_{i=1}^n [(1 - \frac{d_i}{max(d)}) \cdot \frac{\hat{\varepsilon}_i^2}{\sigma^2}]}{\sum_{i=1}^n (1 - \frac{d_i}{max(d)})}, \quad (3)$$

where

$\sigma$  = deviation achieved from the regression model

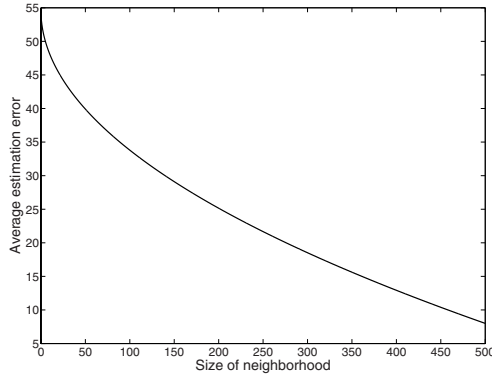
## 4 Study

The method used to observe the need for a model update was based on a combination of the *APEN* and *ASSRN* terms. To be accurate, in both cases a limit, called the exception limit, was used to decide when the value of a term differed too much from the expected and it should be called an exception. The exception limits were developed differently for the property and deviation models. However, because the property and deviation models were both used to solve the final working limits of product manufacturing, it was considered that they should be updated at the same time moments. Thus, the exceptions were combined to find the actual update moment.

In the property model case, it was considered an exception if the model's average prediction error in the neighbourhood of a new observation (*APEN*) differed from zero too much compared with the amount of similar past cases. When there are plenty of accurately predicted similar past cases, the *APEN* is always near zero. When the amount of similar past cases decreases, the sensitivity of the *APEN* (in relation to measurement variation) increases, also in situations when the actual model would be accurate. In other words, the relationship between the sensitivity of the *APEN* and the number of neighbours is negatively correlated. The exception limit for the *APEN* is shown in Figure 1. It defines how high the absolute value of the average prediction error of a neighbourhood (=  $|APEN|$ ) has to be in relation to the size of the neighbourhood before it can be considered an exception. This design was introduced to avoid possible sensitivity issues of the *APEN*. In practice, if the size of the neighbourhood was 500 (the area is well known), prediction errors higher than 8 were defined as exceptions, while with a neighbourhood whose size was 5, the error had to be over 50. The values of the prediction errors used were decided by relating them to the average predicted deviation,  $\hat{\sigma}_i$  ( $\approx 14.4$ ).

In the deviation model case the exceptions were related to the average of the squared standardized residuals in the neighbourhood. Two different boundary values were formed so that if the *ASSRN* was smaller than the first value or greater than the second value, it was considered an exception. The actual exceptions were thus achieved using the equation:

$$ev(i) = \begin{cases} 1, & \text{when } ASSRN < \frac{1}{3} \\ 1, & \text{when } ASSRN > 3 \\ 0, & \text{else} \end{cases} \quad (4)$$



**Fig. 1.** Exception limit.

A second limit, the update limit, was formed to merge the information on exceptions. The limit was defined as being exceeded if the sum of the exceptions within a certain time interval was more than 10 percent of the amount of observation in the interval. The chosen interval was 1000 observations, which represents measurements from approximately one week of production. Thus, the model was retrained every time if the sum of the exception values exceeded a value of 100 during the preceding 1000 observations.

The study was started by training the parameters of the regression model using the first 50,000 observations (approximately one year). After that the trained model was used to give the *APENs* and *ASSRN*s of new observations. The point where the update limit was exceeded the first time was located and the parameters of the model were updated using all the data acquired by then. The study was carried on by studying the accuracy of the model after each update and repeating the steps iteratively until the whole data set was passed.

## 5 Results

The effect of the update was studied by comparing the results of the actual prediction error and the deviation achieved using our approach to the case without an update. The accuracy of the models was measured in the property prediction case using the weighted mean absolute prediction error

$$\text{MAE} = \frac{1}{\sum_{i=1}^N w(i)} \sum_{i=1}^N w(i) |y_i - \hat{\mu}_i|. \quad (5)$$

In the variation model case, a robustified negative log-likelihood was employed to take into account the variance

$$\text{robL} = \frac{1}{\sum_{i=1}^N w(i)} \sum_i w(i) \left( \log(\hat{\sigma}_i^2) + \rho \left[ \frac{(y_i - \hat{\mu}_i)^2}{\hat{\sigma}_i^2} \right] \right). \quad (6)$$

Here, the function  $\rho(\cdot)$  is a robust function, this study employed

$$\rho(t) = \begin{cases} t, & \text{when } t \leq 25 \\ b^2, & \text{when } t > 25 \end{cases} \quad (7)$$

which truncates the squared standardized residuals if the standardized residual is below -5 or above +5.

Two different methods were used to define the weights,  $w(i)$ . They were chosen to reflect the usability value of the models. In the first goodness criterion the weights  $w(i)$  were defined productwise, meaning the weight of the observations of a product could be at most as much as the weight of  $T$  observations. Let  $T_i$  be the number of observations that belong to the same product as the  $i$ th observation. Then the weight of observation  $i$  is

$$w(i) = \begin{cases} 1, & \text{when } T_i \leq T \\ T/T_i, & \text{when } T_i > T. \end{cases} \quad (8)$$

Here the value of  $T = 50$ , meaning that if there is more than 50 observation of a product, the weight is scaled down. The second goodness criterion was formed to take only rare observations into account. Thus, they were cases for which there were only less than 30 previous observation within a distance 0.9 or a distance 1.8, but the weight of the latter was dual (equation 9).

$$w(i) = \begin{cases} 1, & \text{when } \{\#x_j \mid \|x_i - x_j\| < 0.9 \ \& \ j < i\} < 30 \\ 2, & \text{when } \{\#x_j \mid \|x_i - x_j\| < 1.8 \ \& \ j < i\} < 30 \\ 0, & \text{else} \end{cases} \quad (9)$$

In the property model case, the results of the mean absolute prediction errors (MAE) using the goodness criteria are shown in Table 1. The step size indicates the length of the iteration step. The results are averages of the absolute prediction errors of the observations between each iteration step. Thus, the size of the data set used to calculate the average is the same as the step size. In addition to this, to compare the results, the prediction error averages are presented in three different cases: predicted with a newly updated model, with a model updated in the previous iteration step, and with a model that was not updated at all. The results in Table 1 show that, although the differences between the new model and the model from the previous iteration step are not in every case very big, the update improves the prediction in each of the steps. In addition, in some steps it can be seen that even small addition of training data improves the accuracy remarkably. For example, when adding the data of step size 743, the accuracy of the newly updated model is remarkably better compared with the model of the previous iteration step. Naturally, the benefit of the model update is obvious when the results of the updated model and the model without an update are compared.

Table 2 shows the results for the deviation model using the two different goodness criteria. Also in this case, the positive effect of the model update on the robustified negative log-likelihood (robL) can be seen, although the difference between new and previous models is not so high. However, also in this case the update improves the model when compared with the model without an update.

**Table 1.** Means of absolute property prediction errors with the goodness criteria.

Step size	With new model goodness 1	With previous model goodness 1	Without update goodness 1	With new model goodness 2	With previous model goodness 2	Without update goodness 2
12228	-	-	-	-	-	-
36703	11.59	11.74	11.74	14.40	14.62	14.62
18932	10.48	11.01	10.70	13.20	14.15	14.03
5626	10.93	11.07	11.25	12.94	13.11	13.10
17636	12.48	12.57	12.97	16.48	16.60	18.23
6104	11.47	12.39	13.35	12.56	13.87	14.67
743	19.94	20.01	25.72	57.18	57.48	71.18
3772	12.77	14.75	17.77	21.10	36.00	49.63
13533	12.47	12.62	13.68	16.34	17.03	22.37
43338	11.78	11.97	12.51	13.66	13.98	15.61
14831	12.78	13.26	13.77	21.05	20.98	25.55
21397	12.45	12.70	14.26	22.43	23.84	36.60
622	16.20	18.28	25.39	73.18	99.29	156.71
<b>mean</b>	<b>11.99</b>	<b>12.26</b>	<b>12.79</b>	<b>15.43</b>	<b>16.08</b>	<b>18.10</b>

**Table 2.** Robustified negative log-likelihood values with the goodness criteria.

Step size	With new model goodness 1	With previous model goodness 1	Without update goodness 1	With new model goodness 2	With previous model goodness 2	Without update goodness 2
12228	-	-	-	-	-	-
36703	6.29	6.30	6.30	6.87	6.92	6.92
18932	6.13	6.27	6.19	6.57	6.76	6.72
5626	6.22	6.23	6.29	6.52	6.52	6.56
17636	6.38	6.39	6.46	7.15	7.13	7.36
6104	6.32	6.42	6.53	6.62	6.90	6.84
743	8.02	8.06	10.66	15.84	16.11	29.56
3772	6.36	6.94	8.58	7.16	11.58	23.04
13533	6.43	6.45	6.84	7.15	7.20	10.13
43338	6.35	6.38	6.47	6.67	6.71	7.33
14831	6.55	6.65	6.94	7.47	7.52	11.13
21397	6.40	6.45	7.13	7.17	7.54	15.39
622	6.58	6.76	11.47	9.65	11.66	70.85
<b>mean</b>	<b>6.36</b>	<b>6.41</b>	<b>6.61</b>	<b>6.91</b>	<b>7.05</b>	<b>8.28</b>

With this data set, determination of the need for a model update and the actual update process proved their efficiency. The number of iteration steps seems to be quite large, but the iteration steps get longer when more data are used to train the model (the smallness of the last iteration step is due to the end of the data). Thus, the amount of updates decreases as time goes on. However, the developed approach can also adapt to changes rapidly, when needed, as when new products are introduced or the production method of an existing product is changed. Finally, the benefits of this more intelligent updating procedure are obvious in comparison with a dummy periodic update procedure (when the model is updated at one-year intervals, for example, the prediction error means of the property model for the whole data sets are 12.24 using criterion 1 and 16.34 using criterion 2, notably worse than the results achieved with our method, 11.99 and 15.43). The periodic procedure could not react to changes quickly or accurately enough, and in some cases it would react unnecessarily.



## 6 Conclusions

This paper described the development of a method for detecting the need for model updates in the steel industry. The prediction accuracy of regression models may decrease in the long run, and a model update at periodic intervals may not react to changes rapidly and accurately enough. Thus, there was a need for a reliable method for determining suitable times for updating the model. Two limits were used to detect these update times, and the results appear promising. In addition, it is possible to rework the actual values of the limits to optimize the updating steps and improve the results before implementing the method in an actual application environment. Although the procedure was tested using a single data set, the extent of the data set clearly proves the usability of the procedure. Nevertheless, the procedure will be validated when it is adapted also to regression models developed to model the tensile strength and elongation of metal plates.

In this study the model update was performed by using all the previously gathered data to define the regression model parameters. However, in the future the amount of data will increase, making it hard to use all the gathered data in an update. Thus, new methods for intelligent data selection are needed to form suitable training data.

## References

1. Khattree, R., Rao, C., eds.: *Statistics in industry - Handbook of statistics 22*. Elsevier (2003)
2. Juutilainen, I., Röning, J.: Planning of strength margins using joint modelling of mean and dispersion. *Materials and Manufacturing Processes* **21** (2006) 367–373
3. Koskimäki, H., Juutilainen, I., Laurinen, P., Röning, J.: Detection of the need for a model update in steel manufacturing. *Proceedings of International Conference on Informatics in Control, Automation and Robotics* (2007) 55–59
4. Haykin, S.: *Neural Networks, A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, New Jersey (1999)
5. Yang, M., Zhang, H., Fu, J., Yan, F.: A framework for adaptive anomaly detection based on support vector data description. *Lecture Notes in Computer Science, Network and Parallel Computing* (2004) 443–450
6. Gabrys, B., Leiviskä, K., Strackeljan, J.: *Do Smart Adaptive Systems Exist, A Best-Practice for Selection and Combination of Intelligent Methods*. Springer-Verlag, Berlin, Heidelberg (2005)
7. Juutilainen, I., Röning, J., Myllykoski, L.: Modelling the strength of steel plates using regression analysis and neural networks. *Proceedings of International Conference on Computational Intelligence for Modelling, Control and Automation* (2003) 681–691
8. Juutilainen, I., Röning, J.: A method for measuring distance from a training data set. *Communications in Statistics- Theory and Methods* **36** (2007) 2625–2639

# Robust Optimizers for Nonlinear Programming in Approximate Dynamic Programming

Olivier Teytaud and Sylvain Gelly

TAO (Inria), LRI, UMR 8623(CNRS - Univ. Paris-Sud)  
bat 490 Univ. Paris-Sud 91405 Orsay, France  
teytaud@lri.fr

**Abstract.** Many stochastic dynamic programming tasks in continuous action-spaces are tackled through discretization. We here avoid discretization; then, approximate dynamic programming (ADP) involves (i) many learning tasks, performed here by Support Vector Machines, for Bellman-function-regression (ii) many non-linear-optimization tasks for action-selection, for which we compare many algorithms. We include discretizations of the domain as well as other non-linear-programming tools in our experiments, so that by the way we compare optimization approaches and discretization methods. We conclude that robustness is strongly required in the non-linear optimizations in ADP, and experimental results show that (i) discretization is sometimes inefficient, but some specific discretization is very efficient for "bang-bang" problems (ii) simple evolutionary tools outperform quasi-random in a stable manner (iii) gradient-based techniques are much less stable (iv) for most high-dimensional "less unsmooth" problems Covariance-Matrix-Adaptation is first ranked.

## 1 Non-linear Optimization in Stochastic Dynamic Programming (SDP)

Some of the most traditional fields of stochastic dynamic programming, e.g. energy stock-management, which have a strong economic impact, have not been studied thoroughly in the reinforcement learning or approximate dynamic programming (ADP) community. This is damageable to reinforcement learning as it has been pointed out that there are not yet many industrial realizations of reinforcement learning. Energy stock-management leads to continuous problems that are usually handled by traditional linear approaches in which (i) convex value-functions are approximated by linear cuts (leading to piecewise linear approximations (PWLA)) (ii) decisions are solutions of a linear-problem. However, this approach does not work in large dimension, due to the curse of dimensionality which strongly affects PWLA. These problems should be handled by other learning tools. However, in this case, the action-selection, minimizing the expected cost-to-go, can't be anymore done using linear programming, as the Bellman function is no more a convex PWLA.

The action selection is therefore a nonlinear programming problem. There are not a lot of works dealing with continuous actions, and they often do not study the non-linear optimization step involved in action selection. In this paper, we focus on this

part: we compare many non-linear optimization-tools, and we also compare these tools to discretization techniques to quantify the importance of the action-selection step.

We here roughly introduce stochastic dynamic programming. The interested reader is referred to [1] for more details.

Consider a dynamical system that stochastically evolves in time depending upon your decisions. Assume that time is discrete and has finitely many time steps. Assume that the total cost of your decisions is the sum of instantaneous costs. Precisely:

$$\begin{aligned} cost &= c_1 + c_2 + \dots + c_T \\ c_i &= c(x_i, d_i), \quad x_i = f(x_{i-1}, d_{i-1}, \omega_i) \\ d_{i-1} &= strategy(x_{i-1}, \omega_i) \end{aligned}$$

where  $x_i$  is the state at time step  $i$ , the  $\omega_i$  are a random process,  $cost$  is to be minimized, and  $strategy$  is the decision function that has to be optimized. We are interested in a control problem: the element to be optimized is a function.

Stochastic dynamic programming, a tool to solve this control problem, is based on Bellman's optimality principle that can be informally stated as follows:

*"Take the decision at time step  $t$  such that the sum "cost at time step  $t$  due to your decision" plus "expected cost from time step  $t + 1$  to  $\infty$ " is minimal."*

Bellman's optimality principle states that this strategy is optimal. Unfortunately, it can only be applied if the expected cost from time step  $t + 1$  to  $\infty$  can be guessed, depending on the current state of the system and the decision. Bellman's optimality principle reduces the control problem to the computation of this function. If  $x_t$  can be computed from  $x_{t-1}$  and  $d_{t-1}$  (i.e., if  $f$  is known) then the control problem is reduced to the computation of a function

$$V(t, x_t) = E[c(x_t, d_t) + c(x_{t+1}, d_{t+1}) + \dots + c(x_T, d_T)]$$

Note that this function depends on the strategy (we omit for short dependencies on the random process). We consider this expectation for any optimal strategy (even if many strategies are optimal,  $V$  is uniquely determined as it is the same for any optimal strategy).

Stochastic dynamic programming is the computation of  $V$  backwards in time, thanks to the following equation:

$$\begin{aligned} V(t, x_t) &= \inf_{d_t} c(x_t, d_t) + EV(t + 1, x_{t+1}) \\ &\text{or equivalently} \\ V(t, x_t) &= \inf_{d_t} c(x_t, d_t) + EV(t + 1, f(x_t, d_t)) \end{aligned} \quad (1)$$

For each  $t$ ,  $V(t, x_t)$  is computed for many values of  $x_t$ , and then a learning algorithm (here by support vector machines) is applied for building  $x \mapsto V(t, x)$  from these examples. Thanks to Bellman's optimality principle, the computation of  $V$  is sufficient to define an optimal strategy. This is a well known, robust solution, applied in many areas including power supply management. A general introduction, including learning, is [2, 1]. Combined with learning, it can lead to positive results in spite of large dimensions.

Many developments, including RTDP and the field of reinforcement learning, can be found in [3].

Equation 1 is used many many times during a run of dynamic programming. For  $T$  time steps, if  $N$  points are required for efficiently approximating each  $V_t$ , then there are  $T \times N$  optimizations. Furthermore, the derivative of the function to optimize is not always available, due to the fact that complex simulators are sometimes involved in the transition  $f$ . Convexity sometimes holds, but sometimes not. Binary variables are sometimes involved, e.g. in power plants management. This suggests that evolutionary algorithms are a possible tool.

## 1.1 Robustness in Non-linear Optimization

Robustness is one of the main issue in non-linear optimization and has various meanings.

1. A first meaning is the following: robust optimization is the search of  $x$  such that in the neighborhood of  $x$  the fitness is good, and not only at  $x$ . In particular, [4] has introduced the idea that evolutionary algorithms are not function-optimizers, but rather tools for finding wide areas of good fitness.

2. A second meaning is that robust optimization is the avoidance of local minima. It is known that iterative deterministic methods are often more subject to local minima than evolutionary methods; however, various forms of restarts (relaunch the optimization from a different initial point) can also be efficient for avoiding local minima.

3. A third possible meaning is the robustness with respect to fitness noise. Various models of noise and conclusions can be found in [5–9].

4. A fourth possible meaning is the robustness with respect to unsmooth fitness functions, even in cases in which there's no local minima. Evolutionary algorithms are usually rank-based (the next iterate point depends only on the ranks of previously visited points), therefore do not depend on increasing transformations of the fitness-function. It is known that they have optimality properties w.r.t this kind of transformations [10]. For example,  $\sqrt{\|x\|}$  (or some  $C^\infty$  functions close to this one) lead to a very bad behavior of standard Newton-based methods like BFGS [11–14] whereas a rank-based evolutionary algorithm behaves the same for  $\|x\|^2$  and  $\sqrt{\|x\|}$ .

5. The fifth possible meaning is the robustness with respect to the non-deterministic choices made by the algorithm. Even algorithms that are considered as deterministic often have a random part<sup>1</sup>: the choice of the initial point. Population-based methods are more robust in this sense, even if they use more randomness for the initial step (full random initial population compared to only one initial point): a bad initialization which would lead to a disaster is much more unlikely.

The first sense of robustness given above, i.e. avoiding too narrow areas of good fitness, fully applies here. Consider for example a robot navigating in an environment in order to find a target. The robot has to avoid obstacles. The strict optimization of the cost-to-go leads to choices just tangent to obstacles. As at each step the learning is far

---

<sup>1</sup> Or, if not random, a deterministic but arbitrary part, such as the initial point or the initial step-size.

from perfect, then being tangent to obstacles leads to hit the obstacles in 50 % of cases. We see that some local averaging of the fitness is suitable.

The second sense, robustness in front of non-convexity, of course also holds here. Convex and non-convex problems both exist. The law of increasing marginal costs implies the convexity of many stock management problems, but approximations of  $V$  are usually not convex, even if  $V$  is theoretically convex. Almost all problems of robotics are not convex.

The third sense, fitness (or gradient) noise, also applies. The fitness functions are based on learning from finitely many examples. Furthermore, the gradient, when it can be computed, can be pointless even if the learning is somewhat successful; even if  $\hat{f}$  approximates  $f$  in the sense that  $\|f - \hat{f}\|_p$  is small,  $\nabla \hat{f}$  can be very far from  $\nabla f$ .

The fourth sense is also important. Strongly discontinuous fitnesses can exist: obstacle avoidance is a binary reward, as well as target reaching. Also, a production-unit can be switched on or not, depending on the difference between demand and stock-management, and that leads to large binary-costs.

The fifth sense is perhaps the most important. SDP can lead to thousands of optimizations, similar to each other. Being able of solving very precisely 95 % of families of optimization problems is not the goal; here it's better to solve 95 % of any family of optimization problems, possibly in a suboptimal manner. We do think that this requirement is a main explanation of results below.

Many papers have been devoted to ADP, but comparisons are usually far from being extensive. Many papers present an application of one algorithm to one problem, but do not compare two techniques. Problems are often adapted to the algorithm, and therefore comparing results is difficult. Also, the optimization part is often neglected; sometimes not discussed, and sometimes simplified to a discretization.

In this paper, we compare experimentally many non-linear optimization-tools. The list of methods used in the comparison is given in 2. Experiments are presented in section 3. Section 4 concludes.

## 2 Algorithms used in the Comparison

We include in the comparison standard tools from mathematical programming, but also evolutionary algorithms and some discretization techniques. Evolutionary algorithms can work in continuous domains [15–17]; moreover, they are compatible with mixed-integer programming (e.g. [18]). However, as there are not so many algorithms that could naturally work on mixed-integer problems and in order to have a clear comparison with existing methods, we restrict our attention to the continuous framework. We can then easily compare the method with tools from derivative free optimization [19], and limited-BFGS with finite differences [20, 21]. We also considered some very naive algorithms that are possibly interesting thanks to the particular requirement of robustness within a moderate number of iterates: random search and some quasi-random improvements. The discretization techniques are techniques that test a predefined set of actions, and choose the best one. As detailed below, we will use dispersion-based samplings or discrepancy-based samplings.

Cma-ES from Beagle [22, 23] is similar to Cma-ES from EO[24] and therefore it has been removed. We now provide details about the methods integrated in the experiments. For the sake of neutrality and objectivity, none of these source codes has been implemented for this work: they are all existing codes that have been integrated to our platform, except the baseline algorithms.

- random search: randomly draw  $N$  points in the domain of the decisions ; compute their fitness ; consider the minimum fitness.
- quasi-random search: idem, with low discrepancy sequences instead of random sequences [25]. Low discrepancy sequences are a wide area of research [25, 26], with clear improvements on Monte-Carlo methods, in particular for integration but also for learning [27], optimization [25, 28], path planning [29]. Many recent works are concentrated on high dimension [30, 31], with in particular successes when the "true" dimensionality of the underlying distribution or domain is smaller than the apparent one [32], or with scrambling-techniques [33].
- Low-dispersion optimization is similar, but uses low-dispersion sequences [25, 34, 35] instead of random i.i.d sequences ; low-dispersion is related to low-discrepancy, but easier to optimize. A dispersion-criterion is

$$Dispersion(P) = \sup_{x \in D} \inf_{p \in P} d(x, p) \quad (2)$$

where  $d$  is the euclidean distance. It is related to the following (easier to optimize) criterion (to be maximized and not minimized):

$$Dispersion_2(P) = \inf_{(x_1, x_2) \in D^2} d(x_1, x_2) \quad (3)$$

we use Equation. 3 in the sequel of this paper. We optimize dispersion in a greedy manner: each point  $x_n$  is optimal for the dispersion of  $x_1, \dots, x_n$  conditionally to  $x_1, \dots, x_{n-1}$ ; i.e.  $x_1 = (0.5, 0.5, \dots, 0.5)$ ,  $x_2$  is such that  $Dispersion_2(\{x_1, x_2\})$  is maximal, and  $x_n$  is such that

$$Dispersion_2(\{x_1, \dots, x_{n-1}, x_n\}) \text{ is minimal.}$$

This sequence has the advantage of being much faster to compute than the non-greedy one, and that one does not need a priori knowledge of the number of points. Of course, it is not optimal for Equation. 3 or Equation. 2.

- Equation 3 pushes points on the frontier, what is not the case in equation 2 ; therefore, we also considered low-dispersion sequences "far-from-frontier", where equation 3 is replaced by:

$$Dispersion_3(P) = \inf_{(x_1, x_2) \in D^2} d(x_1, \{x_2\} \cup D') \quad (4)$$

As for  $Dispersion_2$ , we indeed used the greedy and incremental counterpart of Equation. 4.

- CMA-ES (EO and openBeagle implementation): an evolution strategy with adaptive covariance matrix [22, 24, 23].

- The Hooke & Jeeves (HJ) algorithm [36–38], available at <http://www.ici.ro/camo/unconstr/hooke.htm> : a geometric local method implemented in C by M.G. Johnson.
- a genetic algorithm (GA), from the *sgLibrary* <http://opendp.sourceforge.net>. It implements a very simple genetic algorithm where the mutation is an isotropic Gaussian of standard deviation  $\frac{\sigma}{\sqrt[n]{n}}$  with  $n$  the number of individuals in the population and  $d$  the dimension of space. The crossover between two individuals  $x$  and  $y$  gives birth to two individuals  $\frac{1}{3}x + \frac{2}{3}y$  and  $\frac{2}{3}x + \frac{1}{3}y$ . Let  $\lambda_1, \lambda_2, \lambda_3, \lambda_4$  be such that  $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$  ; we define  $S_1$  the set of the  $\lambda_1.n$  best individuals,  $S_2$  the  $\lambda_2.n$  best individuals among the others. At each generation, the new offspring is (i) a copy of  $S_1$  (ii)  $n\lambda_2$  cross-overs between individuals from  $S_1$  and individuals from  $S_2$  (iii)  $n\lambda_3$  mutated copies of individuals from  $S_1$  (iv)  $n\lambda_4$  individuals randomly drawn uniformly in the domain. The parameters are  $\sigma = 0.08, \lambda_1 = 1/10, \lambda_2 = 2/10, \lambda_3 = 3/10, \lambda_4 = 4/10$ ; the population size is the square-root of the number of fitness-evaluations allowed. These parameters are standard ones from the library. We also use a "no memory" (GANM) version, that provides as solution the best point in the final offspring, instead of the best visited point. This is made in order to avoid choosing a point from a narrow area of good fitness.
- limited-BFGS with finite differences, thanks to the LBFGB library [20, 21]. Roughly speaking, LBFGB uses an approximated Hessian in order to approximate Newton-steps without the huge computational and space cost associated to the use of a full Hessian.

In our experiments with restart, any optimization that stops due to machine precision is restarted from a new random (independent, uniform) point.

For algorithms based on an initial population, the initial population is chosen randomly (uniformly, independently) in the domain. For algorithms based on an initial point, the initial point is the middle of the domain. For algorithms requiring step sizes, the step size is the distance from the middle to the frontier of the domain (for each direction). Other parameters were chosen by the authors with equal work for each method on a separate benchmark, and then plugged in our dynamic programming tool. The detailed parametrization is available in <http://opendp.sourceforge.net>, with the command-line generating tables of results.

Some other algorithms have been tested and rejected due to their huge computational cost: the DFO-algorithm from Coin [19], <http://www.coin-or.org/>; Cma-ES from Beagle [22, 23] is similar to Cma-ES from EO[24] and has also been removed.

## 3 Experiments

### 3.1 Experimental Settings

The characteristics of the problems are summarized in table 1; problems are scalable and experiments are performed with dimension (i) the baseline dimension in table 1

(ii) twice this dimensionality (iii) three times (iv) four times. Both the state space dimension and the action space are multiplied. Results are presented in tables below. The detailed experimental setup is as follows: the learning of the function value is performed by SVM with Laplacian-kernel (SVM-Torch,[39]), with hyper-parameters heuristically chosen; each optimizer is allowed to use a given number of points (specified in tables of results); 300 points for learning are sampled in a quasi-random manner for each time step, non-linear optimizers are limited to 100 function-evaluations. Each result is averaged among 66 runs. We can summarize results below as follows. Experiments are performed with:

- 2 algorithms for gradient-based methods (LBFSG and LBFSG with restart),
- 3 algorithms for evolutionary algorithms (EO-CMA, GeneticAlgorithm, GeneticAlgorithmNoMemory),
- 4 algorithms for "best of a predefined sample" (Low-Dispersion, Low-Dispersion "fff", Random, Quasi-Random),
- 2 algorithms for pattern-search methods (Hooke&Jeeves, Hooke&Jeeves with restart)

### 3.2 Results

Results varies from one benchmark to another. We have a wide variety of benchmarks, and no clear superiority of one algorithm onto others arises. E.g., CMA is the best algorithm in some cases, and the worst one in some others. One can consider that it would be better to have a clear superiority of one and only one algorithm, and therefore a clear conclusion. Yet, it is better to have plenty of benchmarks, and as a by-product of our experiments, we claim that conclusions extracted from one or two benchmarks, as done in some papers, are unstable, in particular when the benchmark has been adapted to the question under study. The significance of each comparison (for one particular benchmark) can be quantified and in most cases we have sufficiently many experiments to make results significant. But, this significance is for each benchmark independently; in spite of the fact that we have chosen a large set of benchmarks, coming from robotics or industry, we can not conclude that the results could be extended to other benchmarks. However, some (relatively) stable conclusions are:

- For "best of a predefined" set of points (design of experiments):
  - Quasi-random search is better than random search in 17/20 experiments with very good overall significance and close to random in the 3 remaining experiments.
  - But low-dispersion, that is biased in the sense that it "fills" the frontier, is better in 10 on 20 benchmarks only; this is problem-dependent, in the sense that in the "away" or "arm" problem, involving nearly bang-bang solutions (i.e. best actions are often close to the boundary for each action-variable) the Low-dispersion-approach is often the best. LD is the best with strong significance for many \*-problems (in which bang-bang solutions are reasonable).
  - And low-dispersion-fff, that is less biased, outperforms random for 14 on 20 experiments (but is far less impressive for bang-bang-problems).



- For order-2 techniques<sup>2</sup>: LBFGBS outperforms quasi-random-optimization for 9/20 experiments; Restart-LBFGBS outperforms quasi-random optimization for 10/20 experiments. We suggest that this is due to (i) the limited number of points (ii) the non-convex nature of our problems (iii) the cost of estimating a gradient by finite-differences that are not in favor of such a method. Only comparison-based tools were efficient. CMA is a particular tool in the sense that it estimates a covariance (which is directly related to the Hessian), but without computing gradients; a drawback is that CMA is much more expensive (much more computation-time per iterate) than other methods (except BFGS sometimes). However it is sometimes very efficient, as being a good compromise between a precise information (the covariance related to the Hessian) and fast gathering of information (no gradient computation). In particular, CMA was the best algorithm for all stock-management problems (involving precise choices of actions) as soon as the dimension is  $\geq 8$ , with in most cases strong statistical significance.
- The pattern-search method (the Hooke&Jeeves algorithm with Restart) outperforms quasi-random for 10 experiments on 20.
- For the evolutionary-algorithms:
  - EoCMA outperforms Quasi-Random in 5/20 experiments. These 5 experiments are all stock-management in high-dimension, and are often very significant.
  - GeneticAlgorithm outperforms Quasi-Random in 14/20 experiments and Random in 17/20 experiments (with significance in most cases). This algorithm is probably the most stable one in our experiments. GeneticAlgorithmNoMemory outperforms Quasi-Random in 14/20 experiments and Random in 15/20 experiments.

Detailed results are presented in <http://www.lri.fr/teytaud/sefordplong.pdf>; a summary is provided in table 2.

**Table 1.** Summary of the characteristics of the benchmarks. The stock management problems theoretically lead to convex Bellman-functions, but their learnt counterparts are not convex. The "arm" and "away" problem deal with robot-hand-control; these two problems can be handled approximately (but not exactly) by bang-bang solutions. Walls and Multi-Agent problems are motion-control problems with hard penalties when hitting boundaries; the loss functions are very unsmooth.

Name	Nb of time steps	State space dimension (basic case)	Nb scenarios	Action space dimension (basic case)
Stock Management	30	4	9	4
Stock Management V2	30	4	9	4
Fast obstacle avoidance	20	2	0	1
Arm	30	3	50	3
Walls	20	2	0	1
Multi-agent	20	8	0	4
Away	40	2	2	2

<sup>2</sup> We include CMA in order-2 techniques in the sense that it uses a covariance matrix which is strongly related to the Hessian.

**Table 2.** Experimental results. For the "best algorithm" column, **bold** indicates 5% significance for the comparison with all other algorithms and *italic* indicates 5% significance for the comparison with all but one other algorithms. **y** holds for 10%-significance. Detailed results show that many comparisons are significant for larger families of algorithms, e.g. if we group GA and GANM, or if we compare algorithms pairwise. Problems with a star are problems for which bang-bang solutions are intuitively appealing; LD, which over-samples the frontiers, is a natural candidate for such problems. Problems with two stars are problems for which strongly discontinuous penalties can occur; the first meaning of robustness discussed in section 1.1 is fully relevant for these problems. Conclusions: 1. GA outperforms random and often QR. 2. For \*-problems with nearly bang-bang solutions, LD is significantly better than random and QR in all but one case, and it is the best in 7 on 8 problems. It's also in some cases the worst of all the tested techniques, and it outperforms random less often than QR or GA. LD therefore appears as a natural efficient tool for generating nearly bang-bang solutions. 3. In \*\*-problems, GA and GANM are often the two best tools, with strong statistical significance; their robustness for various meanings cited in section 1.1 make them robust solutions for solving non-convex and very unsmooth problems with ADP. 4. Stock management problems (the two first problems) are very efficiently solved by CMA-ES, which is a good compromise between robustness and high-dimensional-efficiency, as soon as dimensionality increases.

Problem	Dim.	Best algo.	QR beats random	GA beats random ; QR	LBFGBSrestart beats random;QR	LD beats random;QR
Stock and Demand	4	<b>LDff</b>	y	y;n	y ; n	y ; n
	8	<b>EoCMA</b>	y	n;n	n ; n	n ; n
	12	<i>EoCMA</i>	y	n;n	n ; n	n ; n
	16	<b>EoCMA</b>	n	n;n	n ; n	n ; n
Stock and Demand2	4	<b>LD</b>	y	y;y	y; y	y; y
	8	<i>EoCMA</i>	n	y;y	n ; y	y ; y
Avoidance	1	<b>HJ</b>	y	y;n	n ; n	y; y
Walls**	1	GA	y	y;y	y ; y	y ; y
Multi-agent**	4	<b>GA</b>	n	y;y	n ;n	n ; n
	8	<b>GANM</b>	y	y;y	n ;n	n ; n
	12	<b>LDff</b>	y	y;y	n ;n	n ; n
	16	<i>GANM</i>	y	y;y	n ;n	y ; n
Arm*	3	LD	y	y;y	y ; y	y; y
	6	HJ	y	y;y	y ; y	y; y
	9	LD	y	y;n	y ; y	y; y
	12	<b>LD</b>	y	y;y	y ; y	y; y
Away*	2	LD	y	y;y	y ; n	y; y
	4	LD	y	y;y	y ; y	y; y
	6	LD	y	y;y	y ; y	y; y
	8	<b>LD</b>	y	y;y	y ; y	y; y
Total			17/20	17/20 ; 14/20	11/20 ; 10/20	14/20 ; 12/20

## 4 Conclusions

We presented an experimental comparison of non linear optimization algorithms in the context of ADP. The comparison involves evolutionary algorithms, (quasi-)random

search, discretizations, and pattern-search-optimization. ADP has strong robustness requirements, thus the use of evolutionary algorithms, known for their robustness properties, is relevant. These experiments are made in a neutral way; we did not work more on a particular algorithm than another. Of course, perhaps some algorithms require more work to become efficient on the problem. The reader can download our source code, modify the conditions, check the parametrization, and experiment himself. Therefore, our source code is freely available at <http://opendp.sourceforge.net> for further experiments.

Our main claims are:

- **High-dimensional Stock-management.** CMA-ES is an efficient evolution-strategy when dimension increases and for "less-unsmooth" problems. It is less robust than the GA, but appears as a very good compromise for the important case of high-dimensional stock-management problems. We do believe that CMA-ES, which is very famous in evolution strategies, is indeed a very good candidate for non-linear optimization as involved in high-dimensional-stock-management where there is enough smoothness for covariance-matrix-adaptation. LBFGS is not satisfactory: in ADP, convexity or derivability are not reliable assumptions, as explained in section 1.1, even if the law of increasing marginal cost applies. Experiments have been performed with dimension ranging from 4 to 16, without heuristic dimension reduction or problem-rewriting in smaller dimension, and results are statistically clearly significant. However, we point out that CMA-ES has a huge computational cost. The algorithms are compared above in the case of a given number of calls to the fitness; this is only a good criterion when the computational cost is mainly the fitness-evaluations. For very-fast fitness-evaluations, CMA-ES might be prohibitively too expensive.
- **Robustness Requirement in Highly Unsmooth Problems.** Evolutionary techniques are the only ones that outperform quasi-random-optimization in a stable manner even in the case of very unsmooth penalty-functions (see \*-problems in the Table 2). The GA is not always the best optimizer, but in most cases it is at least better than random; we do believe that the well-known robustness of evolutionary algorithms, for the five meanings of robustness pointed out in section 1.1, are fully relevant for ADP.
- **A Natural Tool for Generating Bang-bang-efficient Controllers.** In some cases (typically bang-bang problems) the LD-discretization introducing a bias towards the frontiers are (unsurprisingly) the best ones, but for other problems LD leads to the worst results of all techniques tested. This is not a trivial result, as this points out LD as a natural way of generating nearly bang-bang solutions, which depending on the number of function-evaluations allowed, samples the middle of the action space, and then the corners, and then covers the whole action space (what is probably a good "anytime" behavior). A posteriori, LD appears as a natural candidate for such problems, but this was not so obvious a priori.

## References

1. Bertsekas, D., Tsitsiklis, J.: *Neuro-dynamic Programming*. Athena Scientific (1996)
2. Bertsekas, D.: *Dynamic Programming and Optimal Control*, vols I and II. Athena Scientific (1995)
3. Sutton, R., Barto, A.: *Reinforcement learning: An introduction*. MIT Press., Cambridge, MA (1998)
4. DeJong, K.A.: Are genetic algorithms function optimizers ? In Manner, R., Manderick, B., eds.: *Proceedings of the 2<sup>nd</sup> Conference on Parallel Problems Solving from Nature*, North Holland (1992) 3–13
5. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments, a survey. *IEEE Transactions on Evolutionary Computation* 9 (2005) 303–317
6. Sendhoff, B., Beyer, H.G., Olhofer, M.: The influence of stochastic quality functions on evolutionary search. *Recent Advances in Simulated Evolution and Learning* (2004) 152–172
7. Tsutsui, S.: A comparative study on the effects of adding perturbations to phenotypic parameters in genetic algorithms with a robust solution searching scheme. In: *Proceedings of the 1999 IEEE System, Man, and Cybernetics Conference SMC 99*. Volume 3., IEEE (1999) 585–591
8. Fitzpatrick, J., Grefenstette, J.: Genetic algorithms in noisy environments. *Machine Learning: Special Issue on Genetic Algorithms* 3 (1988) 101–120
9. Beyer, H.G., Olhofer, M., Sendhoff, B.: On the impact of systematic noise on the evolutionary optimization performance - a sphere model analysis. *Genetic Programming and Evolvable Machines* 5(2004) 327–360
10. Gelly, S., Ruetten, S., Teytaud, O.: Comparison-based algorithms: worst-case optimality, optimality w.r.t a bayesian prior, the intraclass-variance minimization in eda, and implementations with billiards. In: *PPSN-BTP workshop*. (2006)
11. Broyden, C.G.: The convergence of a class of double-rank minimization algorithms 2. The New Algorithm. *J. of the Inst. for Math. and Applications* 6 (1970) 222–231
12. Fletcher, R.: A new approach to variable-metric algorithms. *Computer Journal* 13 (1970) 317–322
13. Goldfarb, D.: A family of variable-metric algorithms derived by variational means. *Mathematics of Computation* 24 (1970) 23–26
14. Shanno, D.F.: Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation* 24 (1970) 647–656
15. Bäck, T., Hoffmeister, F., Schwefel, H.P.: A survey of evolution strategies. In Belew, R.K., Booker, L.B., eds.: *Proceedings of the 4<sup>th</sup> International Conference on Genetic Algorithms*, Morgan Kaufmann (1991) 2–9
16. Bäck, T., Rudolph, G., Schwefel, H.P.: Evolutionary programming and evolution strategies: Similarities and differences. In Fogel, D.B., Atmar, W., eds.: *Proceedings of the 2<sup>nd</sup> Annual Conference on Evolutionary Programming*, Evolutionary Programming Society (1993) 11–22
17. Beyer, H.G.: *The Theory of Evolutions Strategies*. Springer, Heidelberg (2001)
18. Bäck, T., Schütz, M.: Evolution strategies for mixed-integer optimization of optical multi-layer systems. In McDonnell, J.R., Reynolds, R.G., Fogel, D.B., eds.: *Proceedings of the 4<sup>th</sup> Annual Conference on Evolutionary Programming*, MIT Press (1995)
19. Conn, A., Scheinberg, K., Toint, L.: *Recent progress in unconstrained nonlinear optimization without derivatives* (1997)
20. Zhu, C., Byrd, R., P.Lu, Nocedal, J.: L-BFGS-B: a limited memory FORTRAN code for solving bound constrained optimization problems. Technical Report, EECS Department, Northwestern University (1994)

21. Byrd, R., Lu, P., Nocedal, J., C.Zhu: A limited memory algorithm for bound constrained optimization. *SIAM J. Scientific Computing*, vol.16, no.5 (1995)
22. Hansen, N., Ostermeier, A.: Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: *Proc. of the IEEE Conference on Evolutionary Computation (CEC 1996)*, IEEE Press (1996) 312–317
23. Gagné, C.: *Openbeagle 3.1.0-alpha*. Technical report (2005)
24. Keijzer, M., Merelo, J.J., Romero, G., Schoenauer, M.: Evolving objects: A general purpose evolutionary computation library. In: *Artificial Evolution*. (2001) 231–244
25. Niederreiter, H.: *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM (1992)
26. Owen, A.B.: Quasi-Monte Carlo sampling. In Jensen, H.W., ed.: *Monte Carlo Ray Tracing: Siggraph 2003 Course 44, SIGGRAPH (2003)* 69–88
27. Cervellera, C., Muselli, M.: A deterministic learning approach based on discrepancy. In: *Proceedings of WIRN'03*, pp53-60. (2003)
28. Auger, A., Jebalia, M., Teytaud, O.: Xse: quasi-random mutations for evolution strategies. In: *Proceedings of EA'2005*. (2005) 12–21
29. Tuffin, B.: On the use of low discrepancy sequences in monte carlo methods. In: *Technical Report 1060, I.R.I.S.A.* (1996)
30. Sloan, I., Woźniakowski, H.: When are quasi-Monte Carlo algorithms efficient for high dimensional integrals? *Journal of Complexity* 14 (1998) 1–33
31. Wasilkowski, G., Wozniakowski, H.: The exponent of discrepancy is at most 1.4778. *Math. Comp* 66 (1997) 1125–1132
32. Hickernell, F.J.: A generalized discrepancy and quadrature error bound. *Mathematics of Computation* 67 (1998) 299–322
33. L'Ecuyer, P., Lemieux, C.: Recent advances in randomized quasi-monte carlo methods. In Dror, M., L'Ecuyer, P., Szidarovszkin, F., eds.: *Modeling Uncertainty: An Examination of its Theory, Methods, and Applications*, Kluwer Academic (2002) 419–474
34. Lindemann, S.R., LaValle, S.M.: Incremental low-discrepancy lattice methods for motion planning. In: *Proceedings IEEE International Conference on Robotics and Automation*. (2003) 2920–2927
35. LaValle, S.M., Branicky, M.S., Lindemann, S.R.: On the relationship between classical grid search and probabilistic roadmaps. I. *J. Robotic Res.* 23 (2004) 673–692
36. Hooke, R., Jeeves, T.A.: Direct search solution of numerical and statistical problems. *Journal of the ACM*, Vol. 8, pp. 212-229 (1961)
37. Kaue, A.F.: Algorithm 178: direct search. *Commun. ACM* 6 (1963) 313–314
38. Wright, M.: Direct search methods: Once scorned, now respectable. *Numerical Analysis* (D. F. Griffiths and G. A. Watson, eds.), Pitman Research Notes in Mathematics (1995) 191–208 <http://citeseer.ist.psu.edu/wright95direct.html>.
39. Collobert, R., Bengio, S.: Svmtorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research* 1 (2001) 143–160

# **PART II**

## **Robotics and Automation**

# Improved Positional Accuracy of Robots with High Nonlinear Friction using a Modified Impulse Controller

Stephen van Duin, Christopher D. Cook, Zheng Li and Gursel Alici

University of Wollongong, Faculty of Engineering  
Northfields Avenue, NSW, Wollongong, Australia  
{svanduin, ccook, zheng, gursel}@uow.edu.au

**Abstract.** This paper presents a modified impulse controller to improve the steady state positioning of a SCARA robot having characteristics of high nonlinear friction. A hybrid control scheme consisting of a conventional PID part and an impulsive part is used as a basis to the modified controller. The impulsive part uses short width torque pulses to provide small impacts of force to overcome static friction and move a robot manipulator towards its reference position. It has been shown that this controller can greatly improve a robot's accuracy in position tracking. However, the system in attempting to reach steady state will inevitably enter into a small limit cycle whose amplitude of oscillation is related to the smallest usable impulse. It is shown in this paper that by modifying the impulse controller to adjust the width of successive pulses, the limit cycle can be shifted up or down in position so that the final steady state error can be even further reduced.

**Keywords.** Impulse control, static friction, limit cycle, stick-slip, impulse shape, friction model, accuracy.

## 1 Introduction

In order to remain competitive, precision robot manufacturers continually strive to increase the accuracy of their machinery. The ability of a robot manipulator to position its tool centre point to within a very high accuracy, allows the robot to be used for more precise tasks. For positioning of a tool centre point, the mechanical axes of a robot will be required to be precisely controlled around zero velocity where friction is highly nonlinear and difficult to control.

Nonlinear friction is naturally present in all mechanisms and can cause stick-slip during precise positioning. In many instances, stick-slip has been reduced or avoided by modifying the mechanical properties of the system; however this approach may not always be practical or cost effective. Alternatively, advances in digital technology have made it possible for the power electronics of servomechanisms to be controlled with much greater flexibility. By developing better controllers, the unfavourable effects of nonlinear friction may be reduced or eliminated completely.

Impulse control has been successfully used for accurate positioning of servomechanisms with high friction where conventional control schemes alone have difficulty in approaching zero steady state error. Static and Coulomb friction can cause a conventional PID controller having integral action (I), to overshoot and limit

cycle around the reference position. This is a particular problem near zero velocities where friction is highly nonlinear and the servomechanism is most likely to stick-slip.

Stick-slip can be reduced or eliminated by using impulsive control near or at zero velocities. The impulsive controller is used to overcome static friction by impacting the mechanism and moving it by microscopic amounts. By combining the impulsive controller and conventional controller together, the PID part can be used to provide large scale movement and stability when moving towards the reference position, while the impulse controller is used to improve accuracy for the final positioning where the error signal is small.

By applying a short impulse of sufficient force, plastic deformation occurs between the asperities of mating surfaces resulting in permanent controlled movement. If the initial pulse causes insufficient movement, the impulsive controller produces additional pulses until the position error is reduced to a minimum.

A number of investigators have devised impulsive controllers which achieve precise motion in the presence of friction by controlling the height or width of a pulse. Yang and Tomizuka [17] applied a standard rectangular shaped pulse whereby the height of the pulse is a force about 3 to 4 times greater than the static friction to guarantee movement. The width of the pulse is adaptively adjusted proportional to the error and is used to control the amount of energy required to move the mechanism towards the reference positioning. Alternatively, Popovic [12] described a fuzzy logic pulse controller that determines both the optimum pulse amplitude and pulse width simultaneously using a set of membership functions. Hojjat and Higuchi [6] limited the pulse width to a fixed duration of 1ms and vary the amplitude by applying a force about 10 times the static friction. Rathbun et al [14] identify that a flexible-body plant can result in a position error limit cycle and that this limit cycle can be eliminated by reducing the gain using a piecewise-linear-gain pulse width control law.

In a survey of friction controllers by Armstrong-Hélouvy [2], it is commented that underlying the functioning of these impulsive controllers is the requirement for the mechanism to be in the stuck or stationary position before subsequent impulses are applied. Thus, previous impulse controllers required each small impacting pulse to be followed by an open loop slide ending in a complete stop.

In this paper, a hybrid PID + impulsive controller is used to improve the precision of a servomechanism under the presence of static and Coulomb friction. The design and functioning of the controller does not require the mechanism to come to rest between subsequent pulses, making it suitable for both point to point positioning and speed regulation. The experimental results of this paper show that the shape of the impulse can be optimised to increase the overall precision of the controller. It is shown that the smallest available movement of the servomechanism can be significantly reduced without modification to the mechanical plant.

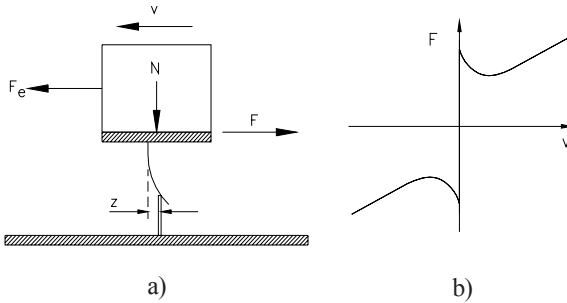
## 2 Modeling and Experimental System

### 2.1 Friction Model

On a broad scale, the properties of friction are both well understood and documented. Armstrong-Hélouvy [2] have surveyed some of the collective understandings of how friction can be modelled to include the complexities of mating surfaces at a



microscopic level. Canudas de Wit [3] add to this contribution by presenting a new model that more accurately captures the dynamic phenomena of rising static friction [13], frictional lag [13], varying break away force [7], [15] dwell time [9], pre-sliding displacement [4], [5], [8] and Stribeck effect [11].



**Fig. 1.** Bristle model; Figure a) shows the deflection of a single bristle. Figure b) shows the resulting static friction model for a single instance in time.

The friction interface is thought of as a contact between elastic bristles. When a tangential force is applied, the bristles deflect like springs which give rise to the friction force [3]; see Fig. 1(a). If the effective applied force  $F_e$  exceeds the bristles force, some of the bristles will be made to slip and permanent plastic movement occurs between each of the mating surfaces. The set of equations governing the dynamics of the bristles are given by [11]:

$$\frac{dz}{dt} = v - \frac{|v|}{g(v)} z \quad (1)$$

$$g(v) = \frac{1}{\sigma_0} \left( F_C + (F_s - F_C) e^{-(v/v_s)^2} \right) \quad (2)$$

$$F = \sigma_0 z + \sigma_1(v) \frac{dz}{dt} + F_v v \quad (3)$$

$$\sigma_1(v) = \sigma_1 e^{-(v/v_d)^2} \quad (4)$$

where  $v$  is the relative velocity between the two surfaces and  $z$  is the average deflection of the bristles.  $\sigma_0$  is the bristle stiffness and  $\sigma_1$  is the bristle damping. The term  $v_s$  is used to introduce the velocity at which the Stribeck effect begins while the parameter  $v_d$  determines the velocity interval around zero for which the velocity damping is active. Fig. 1(b) shows the friction force as a function of velocity.  $F_s$  is the average static friction while  $F_C$  is the average Coulomb friction. For very low velocities, the viscous friction  $F_v$  is negligible but is included for model completeness.  $F_s$ ,  $F_C$ , and  $F_v$  are all estimated experimentally by subjecting a real mechanical system to a series of steady state torque responses. The parameters  $\sigma_0$ ,  $\sigma_1$ ,  $v_s$  and  $v_d$  are also

determined by measuring the steady state friction force when the velocity is held constant [3].

## 2.2 Experimental System

For these experiments, a Hirata ARi350 SCARA (Selective Compliance Assembly Robot Arm) robot was used. The Hirata robot has four axes named A, B, Z and W. The main rotational axes are A-axis (radius 350mm) and B-axis (radius 300mm) and they control the end-effector motion in the horizontal plane. The Z-axis moves the end-effector in the vertical plane with a linear motion, while the W-axis is a revolute joint and rotates the end-effector about the Z-axis. A photograph of the robot is shown in Fig. 2.



**Fig. 2.** The Hirata SCARA robot.

For these experiments, only the A and B axis of the Hirata robot are controlled. Both the A and B axes have a harmonic gearbox between the motor and robot arm. Their gear ratios are respectively 100:1 and 80:1. All of the servomotors on the Hirata robot are permanent magnet DC type and the A and B axis motors are driven with Baldor® TSD series DC servo drives. Each axis has characteristics of high nonlinear friction whose parameters are obtained by direct measurement. For both axes, the static friction is approximately 1.4 times the Coulomb friction.

MATLAB's xPC target oriented server was used to provide control to each of the servomotor drives. For these experiments, each digital drive was used in current control mode which in effect means the output voltage from the 12-bit D/A converter gives a torque command to the actuator's power electronics. The system controller was compiled and run using Matlab's real time xPC Simulink® block code. A 12-bit A/D converter was used to read the actuator's shaft position signal.

## 2.3 PID + Impulse Hybrid Controller

Fig. 3 shows the block diagram of a PID linear controller + impulsive controller. This hybrid controller has been suggested by Li [10] whereby the PID driving torque and impulsive controller driving torque are summed together. It is unnecessary to stop at the end of each sampling period and so the controller can be used for both position and speed control.

The controller can be divided into two parts; the upper part is the continuous driving force for large scale movement and control of external force disturbances. The lower part is an additional proportional controller  $k_{pwm}$  with a pulse width modulated sampled-data hold (PWMH), and is the basis of the impulsive controller for the control of stick-slip.

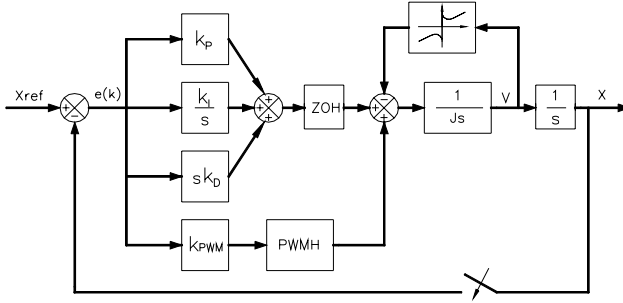


Fig. 3. Block diagram of the experimental system controller.

The system controller is sampled at 2 kHz. The impulse itself is sampled and applied at one twentieth of the overall sampling period (i.e. 100 Hz) to match the mechanical system dynamics. Fig. 4 shows a typical output of the hybrid controller for one impulse sampling period  $\tau_s$ . The pulse with height  $f_p$  is added to the PID output. Because the PID controller is constantly active, the system has the ability to counteract random disturbances applied to the servomechanism. The continuous part of the controller is tuned to react to large errors and high velocity, while the impulse part is optimized for final positioning where stick-slip is most prevalent.

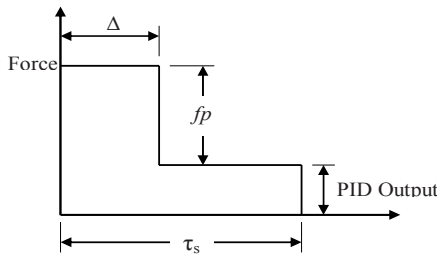


Fig. 4. Friction controller output.

For large errors, the impulse width approaches the full sample period  $\tau_s$ , and for very large errors, it transforms into a continuous driving torque. When this occurs, the combined control action of the PID controller and the impulsive controller will be continuous. Conversely, for small errors, the PID output is too small to have any substantial effect on the servomechanism dynamics.

The high impulse sampling rate, combined with a small error, ensures that the integral (I) part of the PID controller output has insufficient time to rise and produce limit cycling. To counteract this loss of driving torque, when the error is below a threshold, the impulsive controller begins to segment into individual pulses of varying

width and becomes the primary driving force. One way of achieving this is to make the pulse width determined by:

$$\Delta = \frac{k_{pwm} \cdot e(k) \tau_s}{f_p} \quad \text{if } k_{pwm} \cdot |e(k)| \leq |f_p|$$

$$\Delta = \tau_s \quad \text{otherwise} \quad (6)$$

In (6)

$$f_p = |f_p| \cdot \text{sign}(e(k)) \quad (7)$$

where  $e(k)$  is the error input to the controller,  $|f_p|$  is a fixed pulse height greater than the highest static friction and  $\tau_s$  is the overall sampling period. For the experimental results of this paper, the impulsive sampling period  $\tau_s$  was 10ms and the pulse width could be incrementally varied by 1ms intervals. The pulse width gain  $k_{pwm}$ , is experimentally determined by matching the mechanism's observed displacement  $d$  to the calculated pulse width  $t_p$  using the equation of motion:

$$d = \frac{f_p (f_p - f_C)}{2mf_C} t_p^2, \quad f_p > 0 \quad (8)$$

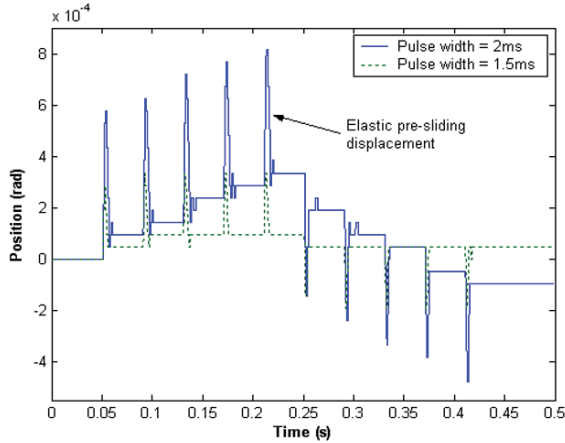
The gain is iteratively adjusted until the net displacement for each incremental pulse width is as small as practical.

## 2.4 Minimum Pulse Width for Position Pointing

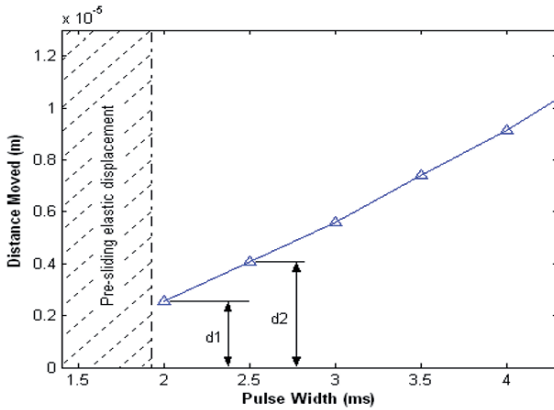
The point to point steady state precision of the system is governed by the smallest incremental movement which will be produced from the smallest usable width pulse. Because the shape of the pulse is affected by the system's electrical circuit response, a practical limit is placed on the amplitude of the pulse over very short durations and this restricts the amount of energy that can be contained within a very thin pulse. Consequently, there exists a minimum pulse width that is necessary to overcome the static friction and guarantee plastic movement.

For the Hirata robot, the minimum pulse width guaranteeing plastic displacement was determined to be 2ms and therefore the pulse width is adjusted between 2 and 10ms. Any pulse smaller than 2ms results in elastic movement of the mating surfaces in the form of pre-sliding displacement. In this regime, short impulses can produce unpredictable displacement or even no displacement at all. In some cases, the mechanism will spring back greater than the forward displacement resulting in a larger error. Fig. 5 shows the displacement of the experimental system of five consecutive positive impulses followed by five negative impulses. The experiment compares impulses of width 2ms and 1.5ms. For impulses of 2ms, the displacement is represented by the consistent staircase movement. For a lesser width of 1.5ms, the

displacement is unpredictable with mostly elastic pre-sliding movement which results in zero net displacement.



**Fig. 5.** Experimentally measured displacement for both positive and negative impulses using successive pulse widths 1.5ms and 2ms.



**Fig. 6.** Simulated displacements as a function of pulse width.

Wu et al [16] use the pre-sliding displacement as a means to increase the precision of the controller by switching the impulse controller off and using a continuous ramped driving torque to hold the system in the desired position. The torque is maintained even after the machine is at rest. This is difficult in practice as pre-sliding movement must be carefully controlled in the presence of varying static friction so that inadvertent breakaway followed by limit cycling is avoided.

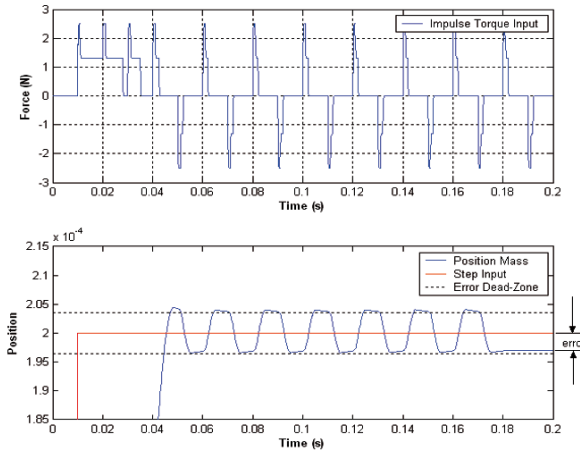
### 3 Limit Cycle Offset

#### 3.1 Motivation

Fig. 6 shows the simulated displacements of varying pulse widths which have been labelled  $d1, d2, d3...dn$  respectively, where  $d1$  is the minimum pulse width which will generate non-elastic movement and defines the system’s resolution.

Using the variable pulse width PID + impulse controller for a position pointing task, the torque will incrementally move the mechanism towards the reference set point in an attempt to reach steady state. Around the set point, the system will inevitably begin to limit cycle when the error  $e(k)$  is approximately the same magnitude as the system resolution (the displacement for the minimum pulse width  $d1$ ).

For the limit cycle to be extinguished, the controller must be disabled. As an example, the limit cycle in Fig. 7 is extinguished by disabling the impulse controller at  $t=0.18s$ , and in this case, the resulting error is approximately half the displacement of the minimum pulse width  $d1$ .



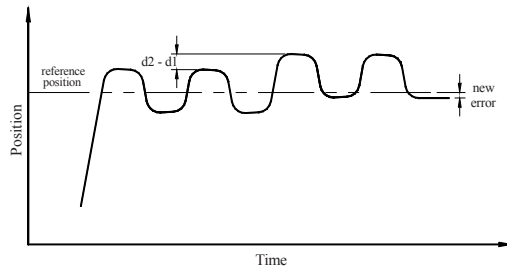
Model Parameter	$F_s$	$F_c$	$\theta$	$I$	$F_v$	$v_s$	$v_d$
Value	2	1	$4.5 \cdot 10^5$	12,000	0.4	0.001	0.0004

**Fig. 7.** Simulation of the impulse controller limit cycling around a position reference set-point where the final torque output is a pulse with a minimum width and the mean peak to peak oscillation is  $d1$ . The friction parameters used for the simulation are also given in the accompanying table.

Limit cycling will occur for all general servomechanisms using a torque pulse because every practical system inherently has a minimum pulse width that defines the system’s resolution. Fig. 7 simulates a typical limit cycle with a peak to peak oscillation equal to the displacement of the minimum pulse width  $d1$ .

One way to automatically extinguish the limit cycle is to include a dead-zone that disables the controller output when the error is between an upper and lower bound of

the reference point (see Fig. 7). The final error is then dependent on the amount of offset the limit cycle has in relation to the reference point. Fig. 7 shows a unique case.



**Fig. 8.** Conceptual example of reducing the steady state error using ‘Limit Cycle Offset’ with the limit cycle shifted up by  $d2-d1$  and the new error that is guaranteed to fall within the dead-zone.

where the  $\pm$  amplitude of the limit cycle is almost evenly distributed either side of the reference set point; i.e. the centre line of the oscillation lies along the reference set point. In this instance, disabling the controller would create an error  $e(k)$  equal to approximately  $\left| \frac{d1}{2} \right|$ . This however, would vary in practice and the centreline is likely to be offset by some arbitrary amount. The maximum precision of the system will therefore be between  $d1$  and zero.

### 3.2 Limit Cycle Offset

By controlling the offset of the limit cycle centreline, it is possible to guarantee that the final error lies within the dead-zone, and therefore to increase the precision of the system. As a conceptual example, Fig. 8 shows a system limit cycling either side of the reference point by the minimum displacement  $d1$ . By applying the next smallest pulse  $d2$ , then followed by the smallest pulse  $d1$ , the limit cycle can be shifted by  $d2 - d1$ . The effect is that the peak to peak centreline of the oscillation has now been shifted away from the reference point.

However, at least one of the peaks of the oscillation has been shifted closer to the set point. If the controller is disabled when the mechanism is closest to the reference set point, a new reduced error is created. For this to be realised, the incremental difference in displacement between successively increasing pulses must be less than the displacement from the minimum pulse width; i.e.  $d2 - d1 < d1$ .

### 3.3 Modified Controller Design

For the limit cycle to be offset at the correct time, the impulse controller must have a set of additional control conditions which identify that a limit cycle has been initiated with the minimum width pulse. The controller then readjusts itself accordingly using a ‘switching bound’ and finally disables itself when within a new specified error

‘dead-zone’. One way to achieve this is to adjust the pulse width so that it is increased by one Pulse Width Increment (PWI) when satisfying the following conditions:

$$\begin{aligned}
 &\text{if} && \text{switching bound} > |e(k)| \geq \text{dead-zone} \\
 &\text{then} && \Delta = \frac{k_{pwm} \cdot e(k) \tau_s}{f_p} + PWI \\
 &\text{otherwise} && \Delta = \frac{k_{pwm} \cdot e(k) \tau_s}{f_p}
 \end{aligned} \tag{9}$$

where the switching bound is given by:

$$|\text{switching bound}| < \frac{d1}{2} \tag{10}$$

and the dead-zone is given by:

$$|\text{dead-zone}| = \frac{(d2 - d1)}{2} \tag{11}$$

The steady state error  $e(k)$  becomes:

$$|e(k)_{\text{steady state}}| \leq \frac{\text{deadzone}}{2} \tag{12}$$

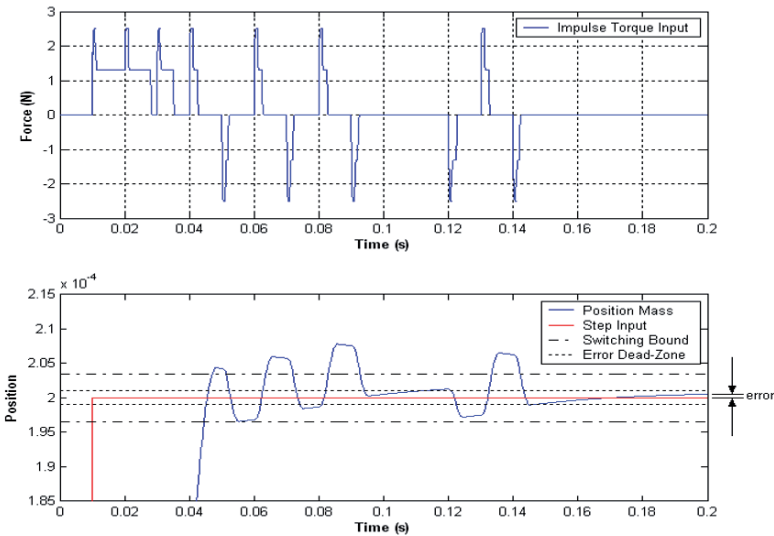


Fig. 9. Simulation of the limit cycle offset function used with the PID + impulse controller.

### 3.4 Simulation of the Limit Cycle Offset Function

To demonstrate the limit cycle offset function, the modified controller is simulated using a simple unit mass with the LeGre friction model [11] using Eqs. 1 to 4.



A simulated step response is shown in Fig. 9 to demonstrate how the modified controller works. Here the mechanism moves towards the reference set point and begins limit cycling. Because at least one of the peaks of the limit cycle immediately lies within the switching bound, the controller shifts the peak to peak oscillation by  $d2 - d1$  by applying the next smallest pulse, and then followed by the smallest pulse. In this example, the first shift is insufficient to move either peak into the set dead-zone so the controller follows with a second shift. At time 0.1 seconds, the controller is disabled; however, the elastic nature of the friction model causes the mechanism's position to move out of the dead-zone. As a result, the controller is reactivated (time 0.12s) and the controller follows with a third shift. In this instance, the mechanism reaches steady state at  $t=0.2s$ , and the final error is  $|e(k)| \leq \frac{1}{2} \cdot (\text{dead zone})$  which in this case is  $\pm 1e-6$  radians.

A final analysis of the result shows that the new controller has reduced the error by an amount significantly more than a standard impulse controller. This reduction correlates directly to the improvement in the system's accuracy by a factor of 4.

## 4 Experimental

### 4.1 Coordinated Motion in Two Axes for Speed Regulation

For continuous coordinated motion, a 100mm diameter circle was drawn using the A and B axes of the robot to compare a conventional PID controller to the PID + impulse control. Fig. 10 shows the relative motion of each axis from the control reference inputs. Velocity reversals occur at  $t=40s$  and  $t=120s$  for the A axis and for the B axis occur at  $t=0$  and  $100s$ . The robot's tool tip angular velocity  $\omega = 31.4$  mrad/s.

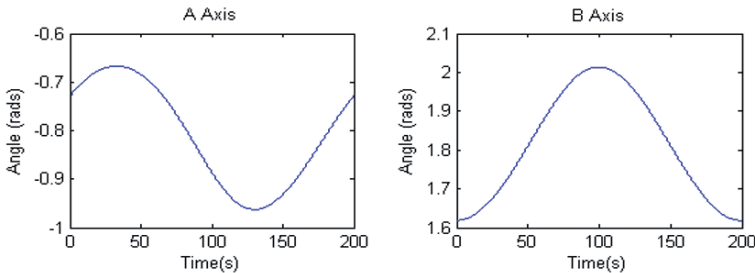


Fig. 10. Reference control signals for the A and B axes ( $\omega=31.4$  mrad/s).

The experimental results are shown in Fig. 11 (a) and Fig. 11 (b) respectively whereby the classic staircase stick-slip motion is extinguished when using the PID + impulse controller. The deviation of the desired 100mm diameter circle is shown in Fig. 12. This is a polar plot where each of the reference position errors for each controller is compared. The maximum deviation from the circle using the PID only controller is  $\pm 3.5mm$ . The maximum deviation using the PID + impulse controller is significantly less with an error of  $\pm 0.1mm$ .

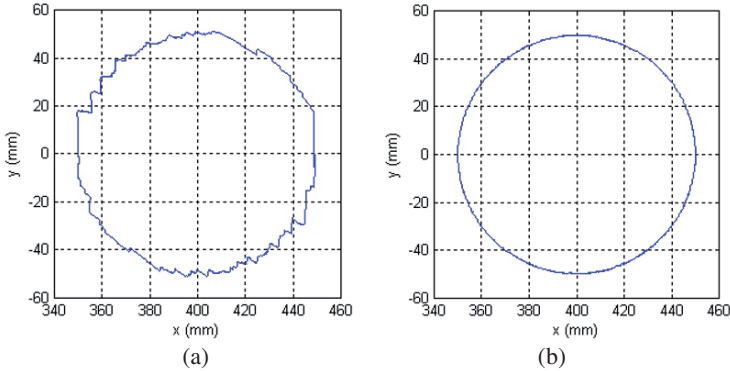


Fig. 11. Circle using PID only (a) and PID + impulse control (b).

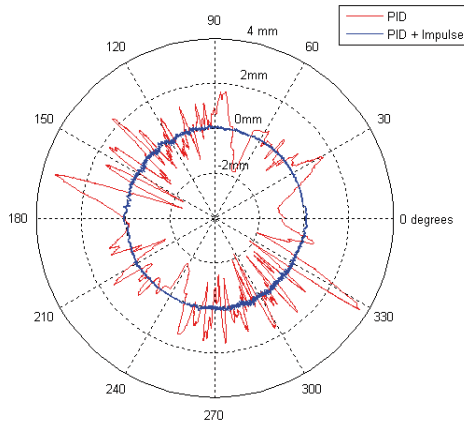


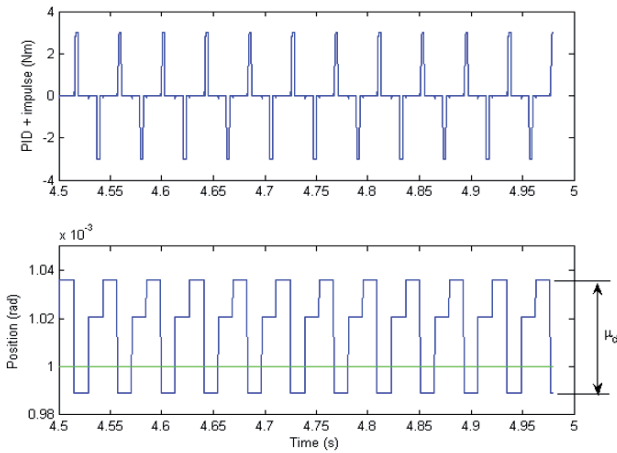
Fig. 12. Circle tracking errors.

## 4.2 Position Pointing

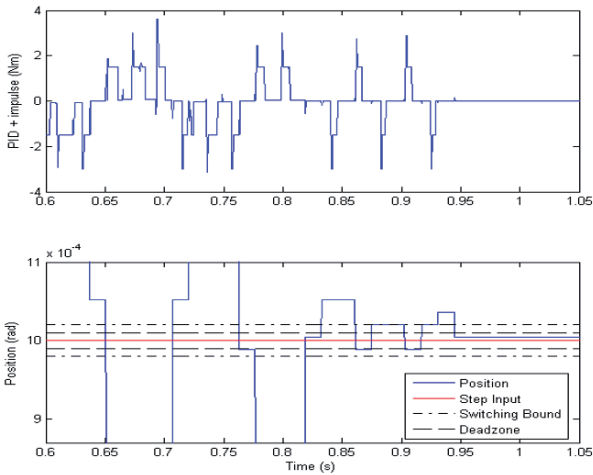
This section evaluates the limit cycle offset function using the experimental Hirata robot having position dependent variables. Fig. 13 shows a steady state limit cycle for a position pointing step response of 0.001 radians using a PID + impulse hybrid controller. The mean peak to peak displacement of the smallest non-elastic part of the limit cycle is  $\mu_r$ .

The experiment was repeated using the limit cycle offset function with the same position step reference of 0.001 radians. Fig. 14 shows a sample experiment and in this example, the limit cycle offset function is activated at  $t=0.9s$ . At this time, the amplitude of the non-elastic part of the limit cycle is identified as lying between the switching bounds. The switching bounds and dead-zone are set according to the methodology given earlier. Once the offset function is activated, the controller adjusts itself by forcing the proceeding pulse to be one increment wider before returning to the smallest pulse width. This results in the limit cycle being shifted down into the

dead-zone region where the impulse controller is automatically disabled at  $t=0.95$ s. At this time, the final error is guaranteed to fall within the error dead zone which can be seen from Fig 14 to be in the vicinity of  $\pm 1e-4$  radians.



**Fig. 13.** Steady state limit cycle for the PID + impulse hybrid controller when applying a unit step input to the Hirata robot. The mean peak to peak displacement  $\mu_d$  is the non-elastic part of limit cycle.



**Fig. 14.** Using the 'Limit Cycle Offset' function to reduce the final steady state error of the Hirata robot.

### 4.3 Discussion of Results

This set of results demonstrates the Limit Cycle Offset function can be successfully applied to a commercial robot manipulator having characteristics of high nonlinear friction. The results show that the unmodified controller will cause the robot to limit

cycle near steady state position and that the peak to peak displacement is equal to the displacement of the smallest usable width pulse.

By using the Limit Cycle Offset function, the limit cycle can be detected and the pulse width adjusted so that at least one of the peaks of the limit cycle is moved towards the reference set point. Finally, the results show that the controller recognises the limit cycle as being shifted into a defined error dead-zone whereby the controller is disabled. The steady state error is therefore guaranteed to fall within a defined region so that the steady state error is reduced. For the SCARA robot, the improvement in accuracy demonstrated was  $1.1 \times 10^{-4}$  radians in comparison to  $4.5 \times 10^{-4}$  radians achieved without the limit cycle offset.

## 5 Conclusions

Advances in digital control have allowed the power electronics of servo amplifiers to be manipulated in a way that will improve a servomechanism precision without modification to the mechanical plant. This is particularly useful for systems having highly nonlinear friction where conventional control schemes alone under perform. A previously developed hybrid PID + impulse controller which does not require the mechanism to come to a complete stop between pulses has been modified to further improve accuracy. This modification shifts the limit cycling into a different position to provide substantial additional improvement in the mechanism's position accuracy. This improvement has been demonstrated both in simulations and in experimental results on a SCARA robot arm. The mechanism does not have to come to a complete stop between pulses, and no mechanical modification has to be made to the robot.

## References

1. Armstrong-Hélouvy, B., 1991, "*Control of Machines with Friction*" Kluwer Academic Publishers, 1991, Norwell MA.
2. Armstrong-Hélouvy, B., Dupont, P., and Canudas de Wit, C., 1994, "*A survey of models, analysis tools and compensation methods for the control of machines with friction*" *Automatica*, vol. 30(7), pp. 1083-1138.
3. Canudas de Wit, C., Olsson, H., Åström, K. J., 1995 "*A new model for control of systems with friction*" *IEEE Transactions on Automatic Control*, vol. 40 (3), pp. 419-425.
4. Dahl, P., 1968, "*A solid friction model*" Aerospace Corp., El Segundo, CA, Tech. Rep. TOR-0158(3107-18)-1.
5. Dahl, P, 1977, "*Measurement of solid friction parameters of ball bearings*" Proc. of 6<sup>th</sup> annual Symp. on Incremental Motion, *Control Systems and Devices*, University of Illinois, ILO.
6. Hojjat, Y., and Higuchi, T., 1991 "*Application of electromagnetic impulsive force to precise positioning*" *Int J. Japan Soc. Precision Engineering*, vol. 25 (1), pp. 39-44.
7. Johannes, V. I., Green, M.A., and Brockley, C.A., 1973, "*The role of the rate of application of the tangential force in determining the static friction coefficient*", *Wear*, vol. 24, pp. 381-385.
8. Johnson, K.L., 1987, "*Contact Mechanics*" Cambridge University Press, Cambridge.

9. Kato, S., Yamaguchi, K. and Matsubayashi, T., 1972, "Some considerations of characteristics of static friction of machine tool slideway" *J. o Lubrication Technology*, vol. 94 (3), pp. 234-247.
10. Li, Z, and Cook, C.D., 1998, "A PID controller for Machines with Friction" Proc. Pacific Conference on Manufacturing, Brisbane, Australia, 18-20 August, 1998, pp. 401-406.
11. Olsson, H., 1996, "Control Systems with Friction" Department of Automatic Control, Lund University, pp.46-48.
12. Popovic, M.R., Gorinevsky, D.M., Goldenberg, A.A., 2000, "High precision positioning of a mechanism with nonlinear friction using a fuzzy logic pulse controller" *IEEE Transactions on Control Systems Technology*, vol. 8 (1) pp. 151-158.
13. Rabinowicz, E., 1958, "The intrinsic variables affecting the stick-slip process," *Proc. Physical Society of London*, vol. 71 (4), pp.668-675.
14. Rathbun, D., Berg, M. C., Buffinton, K. W., 2004, "Piecewise-Linear-Gain Pulse Width Control for Precise Positioning of Structurally Flexible Systems Subject to Stiction and Coulomb Friction", *ASME J. of Dynamic Systems, Measurement and Control*, vol. 126, pp. 139-126.
15. Richardson, R. S. H., and Nolle, H., 1976, "Surface friction under time dependant loads" *Wear*, vol. 37 (1), pp.87-101.
16. Wu, R., Tung, P., 2004, "Fast Positioning Control for Systems with Stick-Slip Friction", *ASME J. of Dynamic Systems, Measurement and Control*, vol. 126, pp. 614-627.
17. Yang, S., Tomizuka, M., 1988, "Adaptive pulse width control for precise positioning under the influence of stiction and Coulomb friction" *ASME J .of Dynamic Systems, Measurement and Control*, vol. 110 (3), pp. 221-227.

# An Estimation Process for Tire-Road Forces and Sideslip Angle for Automotive Safety Systems

Guillaume Baffet, Ali Charara, Daniel Lechner and Damien Thomas

HEUDIASYC Laboratory (UMR CNRS 6599)

Université de Technologie de Compiègne, Centre de recherche Royallieu

BP20529 - 60205 Compiègne, France

INRETS-MA Laboratory (Department of Accident Mechanism Analysis)

Chemin de la Croix Blanche, 13300 Salon de Provence, France

guillaume.baffet@emn.fr, acharara@hds.utc.fr

daniel.lechner@inrets.fr, damien.thomas@sunset.salon.inrets.fr

**Abstract.** This study focuses on the estimation of car dynamic variables for the improvement of vehicle safety, handling characteristics and comfort. More specifically, a new estimation process is proposed to estimate longitudinal/lateral tire-road forces, velocity, sideslip angle and wheel cornering stiffness. This method uses measurements from currently available standard sensors (yaw rate, longitudinal/lateral accelerations, steering angle and angular wheel velocities). The estimation process is separated into two blocks: the first block contains an observer whose principal role is to calculate tire-road forces without a descriptive force model, while in the second block an observer estimates sideslip angle and cornering stiffness with an adaptive tire-force model. The different observers are based on an Extended Kalman Filter (EKF). The estimation process is applied and compared to real experimental data, notably sideslip angle and wheel force measurements. Experimental results show the accuracy and potential of the estimation process.

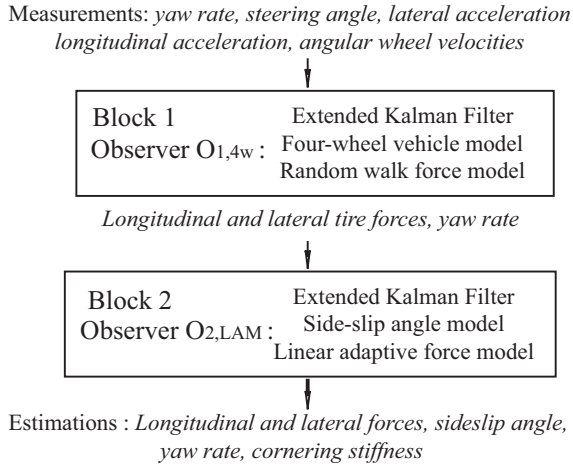
**Keywords.** State observers, vehicle dynamic, sideslip angle estimation, tire-force estimation, wheel cornering stiffness estimation, linear adaptive force model.

## 1 Introduction

The last few years have seen the emergence in cars of active security systems to reduce dangerous situations for drivers. Among these active security systems, Anti-lock Braking Systems (ABS) and Electronic Stability Programs (ESP) significantly reduce the number of road accidents. However, these systems may improved if the dynamic potential of a car is well known. For example, information on tire-road friction means a better definition of potential trajectories, and therefore a better management of vehicle controls. Nowadays, certain fundamental data relating to vehicle-dynamics are not measurable in a standard car for both technical and economic reasons. As a consequence, dynamic variables such as tire forces and sideslip angle must be observed or estimated.

Vehicle-dynamic estimation has been widely discussed in the literature, e.g. ([6], [16], [8], [15], [2]). The vehicle-road system is usually modeled by combining a vehicle model with a tire-force model in one block. One particularity of this study is that it

separates the estimation modeling into two blocks (shown in figure 1), where the first block concerns the car body dynamic while the second is devoted to the tire-road interface dynamic. The first block contains an Extended Kalman Filter (denoted as  $O_{1,4w}$ )



**Fig. 1.** Estimation process. Observers  $O_{1,4w}$  and  $O_{2,LAM}$ .

constructed with a four-wheel vehicle model and a random walk force model. The first observer  $O_{1,4w}$  estimates longitudinal/lateral tire forces, velocity and yaw rate, which are inputs to the observer in the second block (denoted as  $O_{2,LAM}$ ). This second observer is developed from a sideslip angle model and a linear adaptive force model.

Some studies have described observers which take road friction variations into account ([7], [12], [13]). In the works of [7] road friction is considered as a disturbance. Alternatively, as in [12], the tire-force parameters are identified with an observer, while in [13] tire forces are modeled with an integrated random walk model. In this study a linear adaptive tire force model is proposed (in block 2) with an eye to studying road friction variations.

The rest of the paper is organized as follows. The second section describes the vehicle model and the observer  $O_{1,4w}$  (block 1). The third section presents the sideslip angle and cornering stiffness observer ( $O_{2,LAM}$  in block 2). In the fourth section an observability analysis is performed. The fifth section provides experimental results: the two observers are evaluated with respect to sideslip angle and tire-force measurements. Finally, concluding remarks are given in section 6.

## 2 Block 1: Observer For Tire-road Force

This section describes the first observer  $O_{1,4w}$  constructed from a four-wheel vehicle model (figure 2),

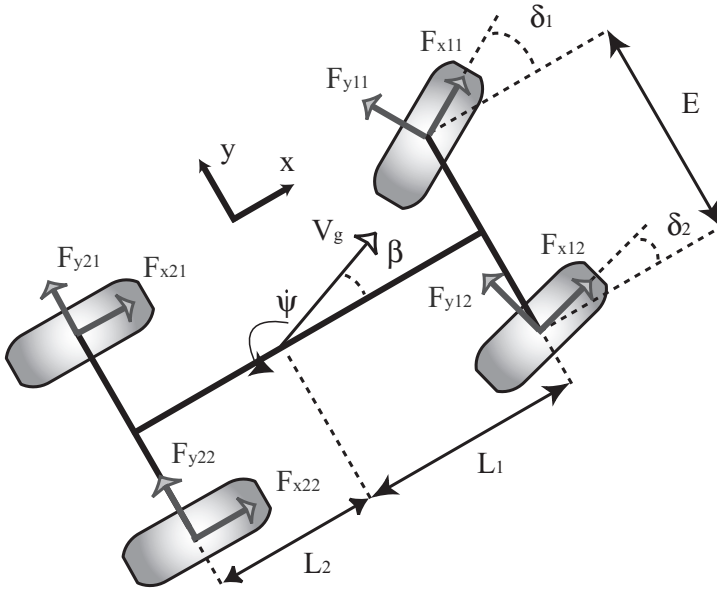


Fig. 2. Four-wheel vehicle model.

where  $\dot{\psi}$  is the yaw rate,  $\beta$  the center of gravity sideslip angle,  $V_g$  the center of gravity velocity, and  $L_1$  and  $L_2$  the distance from the vehicle center of gravity to the front and rear axles respectively.  $F_{x,y,i,j}$  are the longitudinal and lateral tire-road forces,  $\delta_{1,2}$  are the front left and right steering angles respectively, and  $E$  is the vehicle track (lateral distance from wheel to wheel).

In order to develop an observable system (notably in the case of null steering angles), rear longitudinal forces are neglected relative to the front longitudinal forces. The simplified equation for yaw acceleration (four-wheel vehicle model) can be formulated as the following dynamic relationship ( $O_{1,4w}$  model):

$$\ddot{\psi} = \frac{1}{I_z} \begin{bmatrix} L_1[F_{y11} \cos(\delta_1) + F_{y12} \cos(\delta_2)] \\ + F_{x11} \sin(\delta_1) + F_{x12} \sin(\delta_2) \\ -L_2[F_{y21} + F_{y22}] \\ + \frac{E}{2}[F_{y11} \sin(\delta_1) - F_{y12} \sin(\delta_2)] \\ + F_{x12} \cos(\delta_2) - F_{x11} \cos(\delta_1) \end{bmatrix}, \quad (1)$$

where  $m$  the vehicle mass and  $I_z$  the yaw moment of inertia. The different force evolutions are modeled with a random walk model:

$$[F_{xij}^{\cdot}, F_{yij}^{\cdot}] = [0, 0], \quad i = 1, 2 \quad j = 1, 2. \quad (2)$$



The measurement vector  $Y$  and the measurement model are:

$$\begin{aligned}
Y &= [\dot{\psi}, \gamma_y, \gamma_x] = [Y_1, Y_2, Y_3] \\
Y_1 &= \dot{\psi}, \\
Y_2 &= \frac{1}{m}[F_{y11} \cos(\delta_1) + F_{y12} \cos(\delta_2) \\
&\quad + (F_{y21} + F_{y22}) + F_{x11} \sin(\delta_1) + F_{x12} \sin(\delta_2)], \\
Y_3 &= \frac{1}{m}[-F_{y11} \sin(\delta_1) - F_{y12} \sin(\delta_2) \\
&\quad + F_{x11} \cos(\delta_1) + F_{x12} \cos(\delta_2)],
\end{aligned} \tag{3}$$

where  $\gamma_x$  and  $\gamma_y$  are the longitudinal and lateral accelerations respectively.

The  $O_{1,4w}$  system (association between equations (1), random walk force equation (2), and the measurement equations (3)) is not observable in the case where  $F_{y21}$  and  $F_{y22}$  are state vector components. For example, in equation (1, 2, 3) there is no relation allowing the rear lateral forces  $F_{y21}$  and  $F_{y22}$  to be differentiated in the sum  $(F_{y21} + F_{y22})$ : as a consequence only the sum  $(F_{y2} = F_{y21} + F_{y22})$  is observable. Moreover, when driving in a straight line, yaw rate is small,  $\delta_1$  and  $\delta_2$  are approximately null, and hence there is no significant knowledge in equation (1, 2, 3) differentiating  $F_{y11}$  and  $F_{y12}$  in the sum  $(F_{y11} + F_{y12})$ , so only the sum  $(F_{y1} = F_{y11} + F_{y12})$  is observable. These observations lead us to develop the  $O_{1,4w}$  system with a state vector composed of force sums:

$$X = [\dot{\psi}, F_{y1}, F_{y2}, F_{x1}], \tag{4}$$

where  $F_{x1}$  is the sum of front longitudinal forces ( $F_{x1} = F_{x11} + F_{x12}$ ). Tire forces and force sums are associated according to the dispersion of vertical forces:

$$F_{x11} = \frac{F_{z11}F_{x1}}{F_{z12} + F_{z11}}, \quad F_{x12} = \frac{F_{z12}F_{x1}}{F_{z12} + F_{z11}}, \tag{5}$$

$$F_{y11} = \frac{F_{z11}F_{y1}}{F_{z12} + F_{z11}}, \quad F_{y12} = \frac{F_{z12}F_{y1}}{F_{z12} + F_{z11}}, \tag{6}$$

$$F_{y21} = \frac{F_{z21}F_{y2}}{F_{z22} + F_{z21}}, \quad F_{y22} = \frac{F_{z22}F_{y2}}{F_{z22} + F_{z21}}, \tag{7}$$

where  $F_{zij}$  are the vertical forces. These are calculated, neglecting roll and suspension movements, with the following load transfer model:

$$F_{z11} = \frac{L_2mg - h_{cog}m\gamma_x}{2(L_1 + L_2)} - \frac{L_2h_{cog}m\gamma_y}{(L_1 + L_2)E}, \tag{8}$$

$$F_{z12} = \frac{L_2mg - h_{cog}m\gamma_x}{2(L_1 + L_2)} + \frac{L_2h_{cog}m\gamma_y}{(L_1 + L_2)E}, \tag{9}$$

$$F_{z21} = \frac{L_1mg + h_{cog}m\gamma_x}{2(L_1 + L_2)} - \frac{L_2h_{cog}m\gamma_y}{(L_1 + L_2)E}, \tag{10}$$

$$F_{z22} = \frac{L_1mg + h_{cog}m\gamma_x}{2(L_1 + L_2)} + \frac{L_2h_{cog}m\gamma_y}{(L_1 + L_2)E}, \tag{11}$$

$h_{cog}$  being the center of gravity height and  $g$  the gravitational constant. The superposition principle means that the load transfer model assumes the assumption of independent longitudinal and lateral acceleration contributions [8]. The input vectors  $U$  of the

$O_{1,4w}$  observer corresponds to:

$$U = [\delta_1, \delta_2, \beta, F_{z11}, F_{z12}, F_{z21}, F_{z22}]. \quad (12)$$

As regards the vertical force inputs, these are calculated from lateral and longitudinal accelerations with the load transfer model.

### 3 Block 2: Observer for Sideslip Angle and Cornering Stiffness

This section presents the observer  $O_{2,LAM}$  constructed from a sideslip angle model and a tire-force model. The sideslip angle model is based on the single-track model [14], with neglected rear longitudinal force:

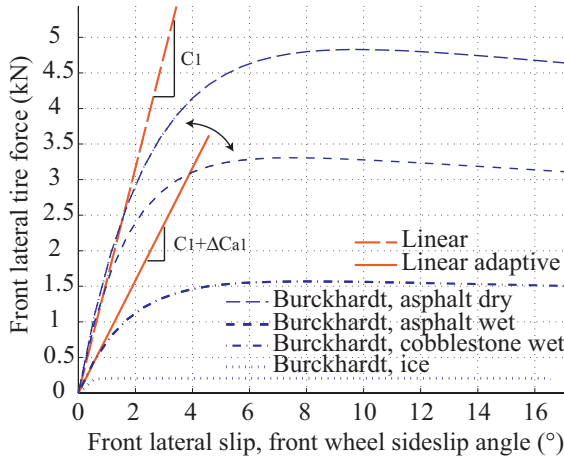
$$\dot{\beta} = \frac{F_{x1} \sin(\delta - \beta) + F_{y1} \cos(\delta - \beta) + F_{y2} \cos(\beta)}{mV_g} - \dot{\psi}. \quad (13)$$

Rear and front sideslip angles are calculated as:

$$\begin{aligned} \beta_1 &= \delta - \beta - L_1 \dot{\psi} / V_g, \\ \beta_2 &= -\beta + L_2 \dot{\psi} / V_g, \end{aligned} \quad (14)$$

where  $\delta$  is the mean of front steering angles.

The dynamic of the tire-road contact is usually formulated by modeling the tire-force as a function of the slip between tire and road ([11], [6], [4]). Figure 3 illustrates different lateral tire-force models (linear, linear adaptive and Burckhardt for various road surfaces [6]). In this study lateral wheel slips are assumed to be equal to the wheel sideslip angles. In current driving situations, lateral tire forces may be considered linear



**Fig. 3.** Lateral tire force models: linear, linear adaptive, Burckhardt for various road surfaces.

with respect to sideslip angle (linear model):

$$F_{yi}(\beta_i) = C_i \beta_i, \quad i = 1, 2, \quad (15)$$

where  $C_i$  is the wheel cornering stiffness, a parameter closely related to tire-road friction.

When road friction changes or when the nonlinear tire domain is reached, "real" wheel cornering stiffness varies. In order to take the wheel cornering stiffness variations into account, we propose an adaptive tire-force model (known as the linear adaptive tire-force model, illustrated in figure 3). This model is based on the linear model at which a readjustment variable  $\Delta C_{ai}$  is added to correct wheel cornering stiffness errors:

$$F_{yi}(\beta_i) = (C_i + \Delta C_{ai})\beta_i. \quad (16)$$

The variable  $\Delta C_{ai}$  is included in the state vector of the  $O_{2,LAM}$  observer and its evolution equation is formulated according to a random walk model ( $\Delta \dot{C}_{ai} = 0$ ). State  $X' \in R^3$ , input  $U' \in R^4$  and measurement  $Y' \in R^3$  are chosen as:

$$\begin{aligned} X' &= [x'_1, x'_2, x'_3] = [\beta, \Delta C_{a1}, \Delta C_{a2}], \\ U' &= [u'_1, u'_2, u'_3, u'_4] = [\delta, \dot{\psi}, V_g, F_{x1}], \\ Y' &= [y'_1, y'_2, y'_3] = [F_{y1}, F_{y2}, \gamma_y]. \end{aligned} \quad (17)$$

The measurement model is

$$\begin{aligned} y'_1 &= (C_1 + x'_2)\beta_1, \\ y'_2 &= (C_2 + x'_3)\beta_2, \\ y'_3 &= \frac{1}{m}[(C_1 + x'_2)\beta_1 \cos(u'_1) + (C_2 + x'_3)\beta_2 \\ &\quad + u'_4 \sin(u'_1)]. \end{aligned} \quad (18)$$

where

$$\begin{aligned} \beta_1 &= u'_1 - x'_1 - L_1 u'_2 / u'_3, \\ \beta_2 &= -x'_1 + L_2 u'_2 / u'_3. \end{aligned} \quad (19)$$

Given the state estimation denoted as  $\widehat{X}' = [\widehat{x}'_1, \widehat{x}'_2, \widehat{x}'_3]$ , the state evolution model of  $O_{2,LAM}$  is:

$$\begin{aligned} \dot{\widehat{x}}'_1 &= \frac{1}{m u_3} [u'_4 \sin(u'_1 - \widehat{x}'_1) + F_{yw1,aux} \cos(u'_1 - \widehat{x}'_1) \\ &\quad + F_{yw2,aux} \cos(\widehat{x}'_1)] - u'_2, \\ \dot{\widehat{x}}'_2 &= 0, \\ \dot{\widehat{x}}'_3 &= 0, \end{aligned} \quad (20)$$

where the auxiliary variables  $F_{yw1,aux}$  and  $F_{yw2,aux}$  are calculated as:

$$\begin{aligned} F_{yw1,aux} &= (C_1 + \widehat{x}'_2)(u'_1 - \widehat{x}'_1 - L_1 u'_2 / u'_3), \\ F_{yw2,aux} &= (C_2 + \widehat{x}'_3)(-\widehat{x}'_1 + L_2 u'_2 / u'_3). \end{aligned} \quad (21)$$

## 4 Estimation Method

The different observers ( $O_{1,4w}$ ,  $O_{2,LAM}$ ) were developed according to an extended Kalman filter method. In 1960 R. E. Kalman published a paper describing a recursive solution to the discrete-data linear filtering problem [5]. Since this publication,

Kalman's method, usually known as the "Extended Kalman Filter", has been the object of extensive search and numerous applications. For example, in [9], Mohinder and Angus present a broad overview of Kalman filtering.

This paragraph describes an EKF algorithm.  $b_{s,k}$ ,  $b_{e,k}$  and  $b_{m,k}$  represent measurement noise at time  $t_k$  for the input and models respectively. This noise is assumed to be Gaussian, white and centered.  $Q_s$ ,  $Q_e$  and  $Q_m$  are the noise variance-covariance matrices for  $b_{s,k}$ ,  $b_{e,k}$  and  $b_{m,k}$ , respectively. The discrete form of models is:

$$\begin{aligned} F(X_k, U_k^*) &= X_k + \int_{t_k}^{t_{k+1}} f(X_k, U_k^*) dt, \\ X_{k+1} &= F(X_k, U_k^*) + b_{m,k}, \\ Y_k &= h(X_k, U_k^*) + b_{s,k}, \\ U_k^* &= U_k + b_{e,k}. \end{aligned} \quad (22)$$

$\hat{X}_k^-$  and  $\hat{X}_k^+$  are state prediction and estimation vectors, respectively, at time  $t_k$ .  $f$  and  $h$  are the evolution and measurement functions. The first step of the EKF is to linearize the evolution equation around the estimated state and input:

$$\begin{aligned} A_k &= \frac{\partial F}{\partial X}(\hat{X}_k^+, U_k^*), \\ B_k &= \frac{\partial F}{\partial U}(\hat{X}_k^+, U_k^*). \end{aligned} \quad (23)$$

The second step is the prediction of the next state, from the previous state and measured input:

$$\hat{X}_{k+1}^- = F(\hat{X}_k^+, U_k^*) \quad (24)$$

The covariance matrix of state estimation uncertainty is then:

$$P_{k+1}^- = A_k P_k^+ A_k^\top + B_k Q_e B_k^\top + Q_m \quad (25)$$

The third step is to calculate the Kalman gain matrix from the linearization of the measurement matrix:

$$\begin{aligned} C_k &= \frac{\partial h}{\partial X}(\hat{X}_{k+1}^-, U_k^*), \\ B_k &= \frac{\partial F}{\partial U}(\hat{X}_{k+1}^-, U_k^*), \\ D_k &= \frac{\partial h}{\partial U}(\hat{X}_{k+1}^-, U_k^*). \end{aligned} \quad (26)$$

The following intermediate variables are used:

$$\begin{aligned} R_k &= C_k P_{k+1}^- C_k^\top + D_k Q_e D_k^\top, \\ S_k &= B_k Q_e D_k^\top, \\ T_k &= P_{k+1}^- C_k^\top + S_k, \end{aligned} \quad (27)$$

and the Kalman gain matrix is:

$$K_k = T_k (R_k + Q_s + C_k S_k + S_k^\top C_k^\top)^{-1} \quad (28)$$

The estimation step is to correct the state vector in line with measurement errors:

$$\hat{X}_{k+1}^+ = \hat{X}_{k+1}^- + K_k (Y_{k+1} - h(\hat{X}_{k+1}^-, U_{k+1}^*)) \quad (29)$$

Finally, the covariance matrix of state estimation uncertainty becomes:

$$P_{k+1}^+ = P_{k+1}^- - K_k (C_k P_{k+1}^- + S_k^\top) \quad (30)$$

## 5 Observability

From the two vehicle-road systems ( $O_{1,4w}$ ,  $O_{2,LAM}$ ), two observability functions were calculated. The two systems are nonlinear, so the observability definition is local and uses the Lie derivative [10].

The Lie derivative of  $h_i$  function, at  $p + 1$  order, is defined as:

$$L_f^{p+1}h_i(\hat{X}) = \frac{\partial L_f^p h_i(\hat{X})}{\partial \hat{X}} f(\hat{X}, U) \quad (31)$$

with

$$L_f^1 h_i(\hat{X}) = \frac{\partial h_i(\hat{X})}{\partial \hat{X}} f(\hat{X}, U) \quad (32)$$

The observability function  $o_i$  corresponding to the measurement function  $h_i$  is defined as:

$$o_i = \begin{pmatrix} dh_i(\hat{X}) \\ dL_f^1 h_i(\hat{X}) \\ \dots \\ dL_f^{(n-1)} h_i(\hat{X}) \end{pmatrix}. \quad (33)$$

where  $n$  is the dimension of  $X$  vector and  $d$  is the operator :

$$dh_i = \left( \frac{\partial h_i}{\partial x_1}, \dots, \frac{\partial h_i}{\partial x_6} \right). \quad (34)$$

The observability function of the system is calculated as:

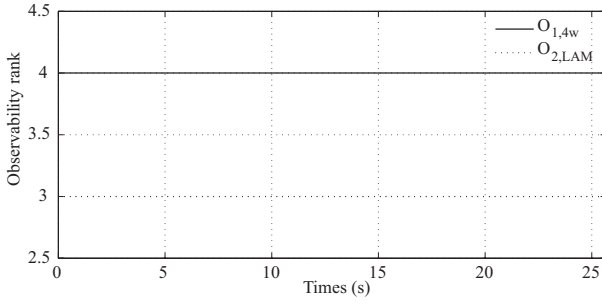
$$O = (o_1, \dots, o_p)^\top \quad (35)$$

where  $p$  is the dimension of the  $Y$  vector. Fig. 4 illustrates observability analysis of the two systems for an experimental test, presented in section 6. Ranks of the two observability functions were 4 (for  $O_{1,4w}$ ) and 3 (for  $O_{2,LAM}$ ) (state dimensions) throughout the test, and consequently the state of the two systems were locally observable.

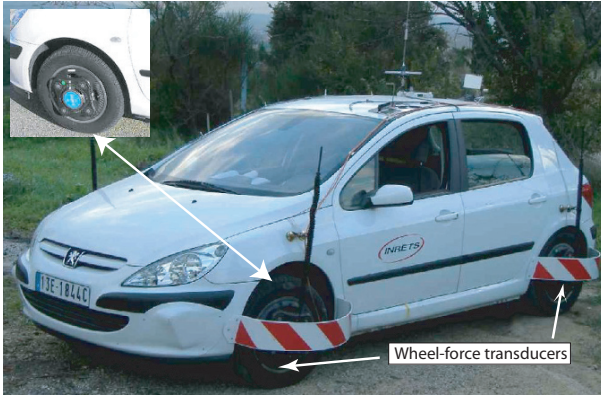
## 6 Experimental Results

The experimental vehicle (see figure 5) is a Peugeot 307 equipped with a number of sensors including GPS, accelerometer, odometer, gyrometer, steering angle, correvit and dynamometric hubs. Among these sensors, the correvit (a non-contact optical sensor) gives measurements of rear sideslip angle and vehicle velocity, while the dynamometric hubs are wheel-force transducers.

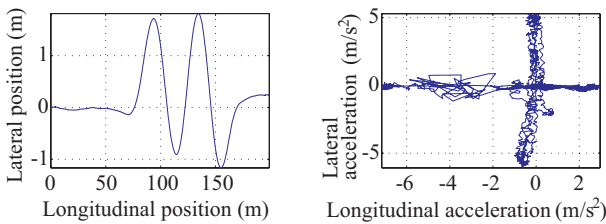
This study uses an experimental test representative of both longitudinal and lateral dynamic behaviors. The vehicle trajectory and the acceleration diagram are shown in figure 6. During the test, the vehicle first accelerated up to  $\gamma_x \approx 0.3g$ , then negotiated



**Fig. 4.** Ranks of the two observability functions for systems  $O_{1,4w}$  and  $O_{2,LAM}$ , during an experimental test (slalom).



**Fig. 5.** Laboratory's experimental vehicle.



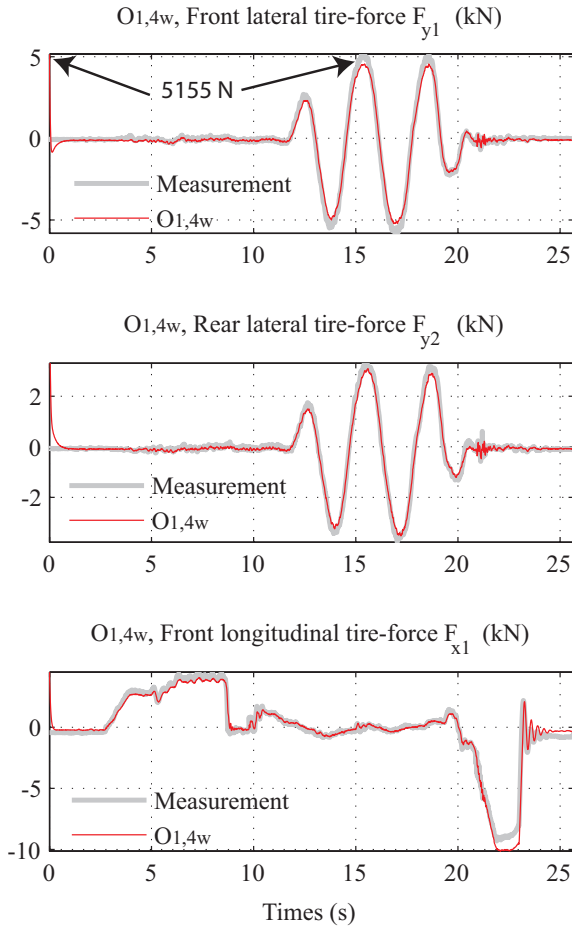
**Fig. 6.** Experimental test, vehicle positions, acceleration diagram.

a slalom at an approximate velocity of  $12m/s$  ( $-0.6g < \gamma_y < 0.6g$ ), before finally decelerating to  $\gamma_x \approx -0.7g$ . The results are presented in two forms: figures of estimations/measurements and tables of normalized errors. The normalized error  $\varepsilon_z$  for an estimation  $z$  is defined in [15] as

$$\varepsilon_z = 100(\|z - z_{measurement}\|)/(\max \|z_{measurement}\|). \tag{36}$$

### 6.1 Block 1: Observer $O_{1,4w}$ Results

During the test, the sideslip angle input of  $O_{1,4w}$  is estimated from the  $O_{2,LAM}$  observer. Figure 7 and table 1 present  $O_{1,4w}$  observer results.



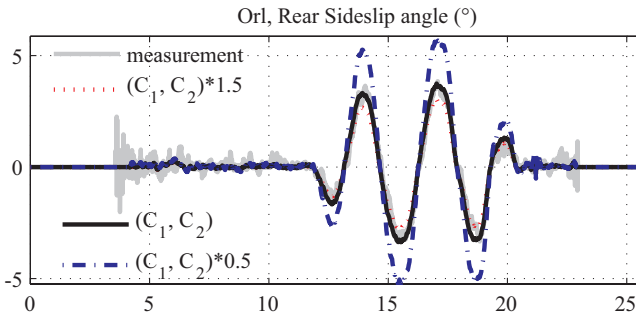
**Fig. 7.** Experimental test.  $O_{1,4w}$  results in comparison with measurements.

The state estimations were initialized using the maximum values for the measurements during the test (for instance, the estimation of the front lateral force  $F_{y1}$  was set to 5155 N). In spite of these false initializations the estimations converge quickly to the measured values, showing the good convergence properties of the observer. Moreover, the  $O_{1,4w}$  observer produces satisfactory estimations close to measurements (normalized mean and standard deviations errors are less than 7%). These good experimental results confirm that the observer approach may be appropriate for the estimation of tire-forces.

### 6.2 Block 2: Observer $O_{2,LAM}$ Results

During the test,  $(F_{x1}, F_{y1}, F_{y2}, V_g)$  inputs of  $O_{2,LAM}$  were originally those from the  $O_{1,4w}$  observer. In order to demonstrate the improvement provided by the observer using the *linear adaptive force model* ( $O_{2,LAM}$ , equation 16), another observer constructed with a *linear fixed force model* is used in comparison (denoted  $O_{rl}$ , equation 15, described in [1]). The robustness of the two observers is tested with respect to tire-road friction variations by performing the tests with different cornering stiffness parameters  $([C_1, C_2] * 0.5, 1, 1.5)$ . The observers were evaluated for the same test presented in section 6.

Figure 8 shows the estimation results of observer  $O_{rl}$  for rear sideslip angle. Observer  $O_{rl}$  gives good results when cornering stiffnesses are approximately known  $([C_1, C_2] * 1)$ . However, this observer is not robust when cornering stiffnesses change  $([C_1, C_2] * 0.5, 2)$ .



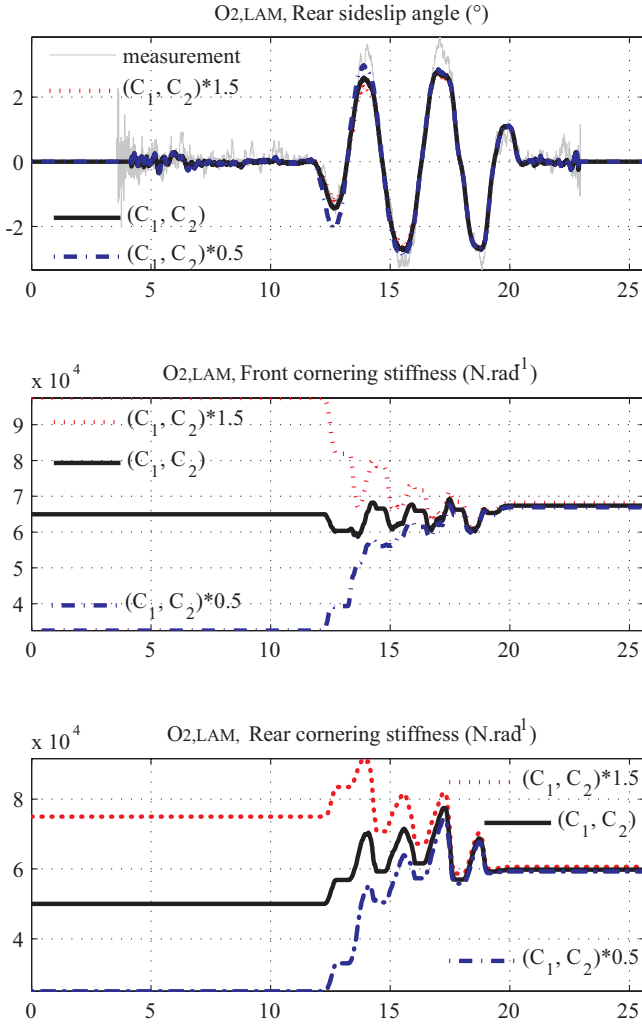
**Fig. 8.** Observer  $O_{rl}$  using a fixed linear force model, rear sideslip angle estimations with different cornering stiffness settings.

Figure 9 and table 2 show estimation results for the adaptive observer  $O_{2,LAM}$ . The performance robustness of  $O_{2,LAM}$  is very good, since sideslip angle is well estimated irrespective of cornering stiffness settings. This result is confirmed by the normalized mean errors (Table 2) which are approximately constant (about 7 %). The front and rear cornering stiffness estimations  $(C_i + \Delta C_i)$  converge quickly to the same values after the beginning of the slalom at 12 s.

**Table 1.** Maximum absolute values,  $O_{1,4w}$  normalized mean errors and normalized standard deviation (Std).

	max	Mean	Std
$F_{y1}$	5816 N	3.1%	4.0%
$F_{y2}$	3782 N	2.9%	5.4%
$F_{x1}$	9305 N	3.1%	4.1%
$\dot{\psi}$	24.6 °/s	0.4%	2.6%





**Fig. 9.**  $O_{2,LAM}$  adaptive observer, Sideslip angle estimation results, Front and rear cornering stiffness estimations  $C_i + \Delta C_i$ , with different cornering stiffness settings.

## 7 Conclusions and Future Work

This study deals with two vehicle-dynamic observers constructed for use in a two-block estimation process. Block 1 mainly estimates tire-forces (without an explicit tire-force model), while block 2 calculates sideslip angle and corrects cornering stiffnesses (with an adaptive tire-force model).

The first observer  $O_{1,4w}$  (block 1), an extended Kalman Filter, is constructed with a random walk force model. The experimental evaluations of  $O_{1,4w}$  are satisfactory, showing

**Table 2.** Observer  $O_{L\text{AM}}$ , rear sideslip angle estimation results, maximum absolute value, normalized mean errors.

$O_{2,L\text{AM}}$	$0.5(C_1, C_2)$	$(C_1, C_2)$	$1.5(C_1, C_2)$
$\max \ \beta_2\ $	$3.0^\circ$	$3.0^\circ$	$3.0^\circ$
Mean	7.4%	7.0%	7.2%

excellent estimations close to the measurements and good convergence properties.

The second observer  $O_{2,L\text{AM}}$  (block 2), developed with an adaptive tire-force model, was evaluated for different cornering stiffness settings and was compared with an observer constructed with a fixed tire-force model ( $O_{rl}$ ). Results show that  $O_{rl}$  is not robust when cornering stiffness parameters change, whereas  $O_{2,L\text{AM}}$  gives excellent estimations of the sideslip angle. This result justifies the use of an adaptive tire-force model to take into account road friction changes.

The different results show the potential of the two-block estimation process. The first block has the advantage of providing satisfactory force estimations without a tire-force model, whereas the second block provides robust sideslip angle estimations with respect to cornering stiffness changes (or tire-road friction variations).

Future studies will improve vehicle-road models, notably for the calculation of the front/rear sideslip angles, in order to widen validity domains for observers. Subsequent vehicle-road models will take into account roll, vertical dynamics and vehicle-tire elasto-kinematics. Moreover, experimental tests will be performed, notably on different road surfaces and in critical driving situations (strong understeering and oversteering).

**Acknowledgements.** This work was supported by the PREDIT/SARI/RADARR program.

## References

1. Baffet, G., Stephant, J., Charara, A.: Vehicle Sideslip Angle and Lateral Tire-Force Estimations in Standard and Critical Driving Situations: Simulations and Experiments. Proceedings of the 8th International Symposium on Advanced Vehicle Control AVEC2006, Taipei Taiwan, (2006)
2. Baffet, G., Stephant, J., Charara, A.: Sideslip angle lateral tire force and road friction estimation in simulations and experiments. Proceedings of the IEEE conference on control application CCA, Munich, Germany, (2006)
3. Bolzern, P., Cheli, F., Falciola, G., Resta, F.: Estimation of the nonlinear suspension tyre cornering forces from experimental road test data. Vehicle system dynamics. Vol. 31 (1999) 23–34
4. Canudas-De-Wit, C., Tsiotras, P., Velenis, E., Basset, M., Gissingner, G.: Dynamic friction models for road/tire longitudinal interaction. Vehicle System Dynamics, Vol. 39 (2003) 189–226
5. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME - PUBLISHER of Basic Engineering, Vol. 82 (1960) 35–45
6. Kiencke U., Nielsen, L.: Automotive control system. Springer, (2000)
7. Lakehal-ayat, M., Tseng, H.E., Mao, Y., Karidas, J.: Disturbance Observer for Lateral Velocity Estimation. Proceedings of the 8th International Symposium on Advanced Vehicle Control AVEC2006, Taipei Taiwan (2006)

8. Lechner, D.: Analyse du comportement dynamique des vehicules routiers legers: developpement d'une methodologie appliquee a la securite primaire. Ph.D. dissertation Ecole Centrale de Lyon, France (2002)
9. Mohinder, S.G., Angus, P.A.: Kalman filtering theory and practice. Prentice hall, (1993)
10. Nijmeijer, H., Van der Schaft, A.J.: Nonlinear Dynamical Control Systems. Springer-Verlag, (1990)
11. Pacejka, H.B., Bakker, E.: The magic formula tyre model. Int. colloq. on tyre models for vehicle dynamics analysis, (1991) 1–18
12. Rabhi, A., M'Sirdi, N.K., Zbiri, N., Delanne, Y: Vehicle-road interaction modelling for estimation of contact forces. Vehicle System Dynamics, Vol. 43 (2005) 403–411
13. Ray, L.: Nonlinear Tire Force Estimation and Road Friction Identification : Simulation and Experiments. Automatica, Vol. 33 (1997) 1819–1833
14. Segel, M.L.: Theoretical prediction and experimental substantiation of the response of the automobile to steering control. automobile division of the institut of mechanical engineers, Vol. 7 (1956) 310–330
15. Stephant, J., Charara, A., Meizel, D.: Evaluation of a sliding mode observer for vehicle sideslip angle. Control Engineering Practice, Available online 5 June 2006
16. Ungoren, A.Y., Peng, H., Tseng, H.E.: A study on lateral speed estimation methods. Int. J. Vehicle Autonomous Systems, Vol. 2 (2004) 126–144

# SMARTMOBILE and its Applications to Guaranteed Modeling and Simulation of Mechanical Systems

Ekaterina Auer and Wolfram Luther

IIIS, University of Duisburg-Essen, Lotharstr. 63, Duisburg, Germany  
{auer, luther}@inf.uni-due.de

**Abstract.** To automatize modeling and simulation of mechanical systems for industry and research, a number of tools were developed, among which the program MOBILE plays a considerable role. However, such tools cannot guarantee the correctness of results, for example, because of possible errors in the underlying finite precision arithmetic. To avoid such errors and prove the correctness of results, a number of so called validated methods were developed, which include interval and Taylor form based arithmetics. In this paper, we present the multibody modeling and simulation tool SMARTMOBILE based on MOBILE, which provides results guaranteed to be correct. The use of validated methods there allows us additionally to take into account the uncertainty in measurements and study its influence on simulation. We demonstrate the main concepts and usage of SMARTMOBILE with the help of several applications.

**Keywords.** Validated method, interval, Taylor model, initial value problem, guaranteed multibody modeling and simulation.

## 1 Introduction

Modeling and simulation of kinematics and dynamics of mechanical systems is employed in many branches of modern industry and applied science. This fact contributed to the appearance of various tools for automatic generation and simulation of models of multibody systems, for example, MOBILE [1]. Such tools produce a model (mostly a system of differential or algebraic equations or both) from a formalized description of the goal mechanical system. The system is then solved using a corresponding numerical algorithm. However, the usual implementations are based on finite precision arithmetic, which might lead to unexpected errors due to round off and similar effects. For example, unreliable numerics might ruin an election (German Green Party Convention in 2002) or even cost people lives (Patriot Missile failure during the Golf War), see [2], [3].

Aside from finite precision errors, possible measurement uncertainties in model parameters and errors induced by model idealization encourage the employment of a technique called interval arithmetic and its extensions in multibody modeling and simulation tools. Essential ideas of interval arithmetic were developed simultaneously and independently by several people whereas the most influential theory was formulated by R. E. Moore [4]. Instead of providing a point on the real number axis as an (inexact) answer, intervals supply the lower and upper bounds that are guaranteed to contain the

true result. These two numbers can be chosen so as to be exactly representable in a given finite precision arithmetic, which cannot be always ensured in the usual finite precision case. The ability to provide a guaranteed result supplied a name for such techniques – “validated arithmetics”. Their major drawback is that the output might be too uncertain (e.g.  $[-\infty; +\infty]$ ) to provide a meaningful answer. Usually, this is an indication that the problem might be ill conditioned or inappropriately formulated, and so the finite precision result wrong.

To minimize the possible influence of overestimation on the interval result, this technique was extended with the help of such notions as affine [5] or Taylor forms/models [6]. Besides, strategies and algorithms much less vulnerable to overestimation were developed. They include rearranging expression evaluation, coordinate transformations, or zonotopes [7].

SMARTMOBILE<sup>1</sup> enhances the usual, floating point based MOBILE with validated arithmetics and initial value problem (IVP) solvers [8]. In this way, it can model and perform validated simulation of the behavior of various classes of mechanical systems including non-autonomous and closed-loop ones as well as provide more realistic models by taking into account the uncertainty in parameters.

In this paper, we give an overview of the structure and the abilities of SMARTMOBILE. The main validated techniques and software are referenced briefly in Section 2. In Section 3, we describe in short the main features of MOBILE and focus on the implementation of SMARTMOBILE. Finally, a number of applications of this tool are provided in Section 4. We summarize the paper in Section 5. On the whole, this paper contains an overview of the potential of validated methods in mechanical modeling, and, in particular, the potential of SMARTMOBILE.

## 2 Validated Methods and Software

To guarantee the correctness of MOBILE results, it is necessary to enhance this tool with validated concepts. Fortunately, we do not need to implement these concepts from scratch. In the last decades, various libraries were implemented that supported different aspects of validated calculus. In the first Subsection, we name several of these tools. After that, we give a very brief overview of interval arithmetic and an algorithm for solving IVPs to provide a reference about the difference of validated methods to the usual floating point ones.

### 2.1 Validating Multibody Tools of the Numerical Type

To validate the results of multibody modeling and simulation software of the numerical type (cf. Section 3.1), the following components are necessary. First, the means are required to work with arithmetic operations and standard functions such as sine or cosine in a guaranteed way. Here, the basic principles of interval calculus and its extensions are used. Interval arithmetic is implemented in such libraries as PROFIL/BIAS [9], FILIB++ [10], C-XSC [11]. LIBAFFA [12] is a library for affine arithmetic, whereas COSY [13] implements Taylor models.

---

<sup>1</sup> Simulation and Modeling of dynAmics in MOBILE: Reliable and Template based

Second, validated algorithms for solving systems of algebraic, differential or algebraic-differential equations are necessary. C-XSC TOOLBOX [14] offers a general means of solving different classes of systems as well as an implementation in C-XSC. For IVP solving in interval arithmetic, there exist such packages as AWA [15], VN-ODE [16], and recently developed VALENCIA-IVP [8]. In the framework of Taylor models, the solver COSY VI [17] was developed.

Finally, almost all of the above mentioned solvers need means of computing (high order) derivatives automatically. Some of them, for example, COSY VI, use the facilities provided by the basis arithmetic in COSY. Interval implementations do not possess this facility in general; external tools are necessary in this case. The symbolic form of the mathematical model, which is necessary to be able to obtain derivatives automatically, is not available in case of software of the numerical type. However, a method called algorithmic differentiation [18] offers a possibility to obtain the derivatives using the code of the program itself.

There are two main techniques to implement algorithmic differentiation of a piece of program code: overloading and code transformation. In the first case, a new data type is developed that is capable of computing the derivative along with the function value. This new data type is used instead of the simple one in the code piece. The drawback of this method is the lack of automatic optimization during the derivative computation. FADBAD++ [19] is a generic library implementing this approach for arbitrary user-defined basic data types. The technique of code transformation presupposes the development of a compiler that takes the original code fragment and the set of differentiation rules as its input and produces a program delivering derivatives as its output. This approach might be difficult to implement for large pieces of code which are self-contained programs themselves. However, derivatives can be evaluated more efficiently with this technique. An implementation is offered in the library ADOL-C [20].

This list of tools is not supposed to be complete. All of the above mentioned packages are implemented (or have versions) in C++, an important criterium from our point of view since MOBILE is also implemented in this language.

## 2.2 Theory Overview

In this Subsection, we consider the basic principles of validated computations using the example of interval arithmetic. First, elementary operations in this arithmetic are described. Then a basic interval algorithm for solving IVPs is outlined to give an impression of the difference to floating point analogues. In particular, the latter passage makes clear why automatic differentiation is unavoidable while simulating dynamics of mechanical systems, that is, solving systems of differential equations.

An interval  $[\underline{x}; \bar{x}]$ , where  $\underline{x}$  is the lower,  $\bar{x}$  the upper bound, is defined as  $[\underline{x}; \bar{x}] = \{x \in \mathbb{R} : \underline{x} \leq x \leq \bar{x}\}$ . For any operation  $\circ = \{+, -, \cdot, /\}$  and intervals  $[\underline{x}; \bar{x}]$ ,  $[\underline{y}; \bar{y}]$ , the corresponding interval operation can be defined as  $[\underline{x}; \bar{x}] \circ [\underline{y}; \bar{y}] =$

$$[\min(\underline{x} \circ \underline{y}, \underline{x} \circ \bar{y}, \bar{x} \circ \underline{y}, \bar{x} \circ \bar{y}); \max(\underline{x} \circ \underline{y}, \underline{x} \circ \bar{y}, \bar{x} \circ \underline{y}, \bar{x} \circ \bar{y})] .$$

Note that the result of an interval operation is also an interval. Every possible combination of  $x \circ y$ , where  $x \in [\underline{x}; \bar{x}]$  and  $y \in [\underline{y}; \bar{y}]$ , lies inside this interval. (For division, it is assumed that  $0 \notin [\underline{y}; \bar{y}]$ .)

To be able to work with this definition on a computer using a finite precision arithmetic, a concept of a machine interval is necessary. The machine interval has machine numbers as the lower and upper bounds. To obtain the corresponding machine interval for the real interval  $[\underline{x}; \bar{x}]$ , the lower bound is rounded down to the largest machine number equal or less than  $\underline{x}$ , and the upper bound is rounded up to the smallest machine number equal or greater than  $\bar{x}$ .

Consider an algorithm for solving the IVP

$$\begin{cases} \dot{x}(t) = f(x(t)), \\ x(t_0) \in [x_0], \end{cases} \quad (1)$$

where  $t \in [t_0, t_n] \subset \mathbb{R}$  for some  $t_n > t_0$ ,  $f \in C^{p-1}(\mathcal{D})$  for some  $p > 1$ ,  $\mathcal{D} \subseteq \mathbb{R}^m$  is open,  $f : \mathcal{D} \mapsto \mathbb{R}^m$ , and  $[x_0] \subset \mathcal{D}$ . The problem is discretized on a grid  $t_0 < t_1 < \dots < t_n$  with  $h_{k-1} = t_k - t_{k-1}$ . Denote the solution with the initial condition  $x(t_{k-1}) = x_{k-1}$  by  $x(t; t_{k-1}, x_{k-1})$  and the set of solutions  $\{x(t; t_{k-1}, x_{k-1}) \mid x_{k-1} \in [x_{k-1}]\}$  by  $x(t; t_{k-1}, [x_{k-1}])$ . The goal is to find interval vectors  $[x_k]$  for which the relation  $x(t_k; t_0, [x_0]) \subseteq [x_k]$ ,  $k = 1, \dots, n$  holds.

The (simplified)  $k$ th time step of the algorithm consists of two stages [21] :

**1. Proof of existence and uniqueness.** Compute a step size  $h_{k-1}$  and an a priori enclosure  $[\tilde{x}_{k-1}]$  of the solution such that

- (i)  $x(t; t_{k-1}, x_{k-1})$  is guaranteed to exist for all  $t \in [t_{k-1}; t_k]$  and all  $x_{k-1} \in [x_{k-1}]$ ,
- (ii) the set of solutions  $x(t; t_{k-1}, [x_{k-1}])$  is a subset of  $[\tilde{x}_{k-1}]$  for all  $t \in [t_{k-1}; t_k]$ .

Here, Banach's fixed-point theorem is applied to the Picard iteration.

**2. Computation of the solution.** Compute a tight enclosure  $[x_k] \subseteq [\tilde{x}_{k-1}]$  of the solution of the IVP such that  $x(t_k; t_0, [x_0]) \subseteq [x_k]$ . The prevailing algorithm is as follows.

**2.1.** Choose a one-step method

$$x(t; t_k, x_k) = x(t; t_{k-1}, x_{k-1}) + h_{k-1}\varphi(x(t; t_{k-1}, x_{k-1})) + z_k ,$$

where  $\varphi(\cdot)$  is an appropriate method function, and  $z_k$  is the local error which takes into account discretization effects. The usual choice for  $\varphi(\cdot)$  is a Taylor series expansion.

**2.2.** Find an enclosure for the local error  $z_k$ . For the Taylor series expansion of order  $p - 1$ , this enclosure is obtained as  $[z_k] = h_{k-1}^p f^{[p]}([\tilde{x}_{k-1}])$ , where  $f^{[p]}([\tilde{x}_{k-1}])$  is an enclosure of the  $p$ th Taylor coefficient of the solution over the state enclosure  $[\tilde{x}_{k-1}]$  determined by the Picard iteration in Stage One.

**2.3.** Compute a tight enclosure of the solution. If mean-value evaluation for computing the enclosures of the ranges of  $f^{[i]}([x_k])$ ,  $i = 1, \dots, p - 1$ , instead of the direct evaluation of  $f^{[i]}([x_k])$  is used, tighter enclosures can be obtained.

Note that Taylor coefficients and their Jacobians (used in the mean-value evaluation) are necessary to be able to use this algorithm.

### 3 SMARTMOBILE

In this Section, we first describe the main features of MOBILE in short to provide a better understanding of the underlying structure of SMARTMOBILE. The implementation and features of this latter tool, which produces guaranteed results in the constraints

of the given model, are summarized afterwards. Note that not only simulation, but modeling itself can be enhanced in SMARTMOBILE by taking into account the uncertainty in parameters, which might result, for example, from measurements.

### 3.1 MOBILE

MOBILE is an object oriented C++ environment for modeling and simulation of kinematics and dynamics of mechanical systems based on the multibody modeling method. Its central concept is a transmission element which maps motion and force between system states. For example, an elementary joint modeling revolute and prismatic joints is such a transmission element. Mechanical systems are considered to be concatenations of these entities. In this way, serial chains, tree type or closed loop systems can be modeled. With the help of the global kinematics, the transmission function of the complete system chain can be obtained from transmission functions of its parts. The inverse kinematics and the kinetostatic method [22] help to build dynamic equations of motion, which are solved with common IVP solvers. MOBILE belongs to the numerical type of modeling software, that is, it does not produce a symbolic description of the resulting model. Only the values of output parameters for the user-defined values of input parameters and the source code of the program itself are available. In this case, it is necessary to integrate verified techniques into the core of the software itself, as opposed to the tools of the symbolical type, where the task is basically reduced to the application of the validated methods to the obtained system of equations.

All transmission elements in MOBILE are derived from the abstract class `MoMap`, which supplies their main functionality including the methods `doMotion()` and `doForce()` for transmission of motion and force. For example, elementary joints are modeled by the class `MoElementaryJoint`. Besides, there exist elements for modeling mass properties and applied forces. Transmission elements are assembled to chains implemented by the class `MoMapChain`. The methods `doMotion()` and `doForce()` can be used for a chain representing the system to determine the corresponding composite transmission function. The class `MoEqmBuilder` is responsible for generation of equations of motion, which are subsequently transferred into their state-space form by `MoMechanicalSystem`. Finally, the corresponding IVP is solved by an appropriate integrator algorithm, for example, Runge–Kutta's using the class `MoRungeKuttaIntegrator` derived from the basic class `MoIntegrator`.

### 3.2 Main Features of SMARTMOBILE

The focus of SMARTMOBILE is to model and simulate dynamics of mechanical systems in a guaranteed way. The concept behind MOBILE, however, presupposes that kinematics is also modeled (almost as a by-product) and so it is easy to simulate it afterwards. That is why SMARTMOBILE is one of the rare validated tools that possess both functionalities.

To simulate dynamics, an IVP for the differential(-algebraic) equations of motion of the system model in the state space form has to be solved. As already mentioned, validated IVP solvers need derivatives of the right side of these equations. They can be obtained using algorithmic differentiation, the method that is practicable but might



consume a lot of CPU time in case of such a large program as MOBILE. An alternative is to make use of the system's mechanics for this purpose. This option is not provided by MOBILE developers yet and seems to be rather difficult to algorithmize for (arbitrary) higher orders of derivatives. That is why it was decided to employ the first possibility in SMARTMOBILE.

To obtain the derivatives, SMARTMOBILE uses the overloading technique. In accordance with Subsection 2.1, all relevant occurrences of `MoReal` (an alias of `double` in MOBILE) have to be replaced with an appropriate new data type. Almost each validated solver needs a different basic validated data type (cf. Table 1). Therefore, the strategy in SMARTMOBILE is to use pairs `type/solver`. To provide interval validation with the help of VNODE-based solver `TMoAWAIntegrator`, the basic data type `TMoInterval` including data types necessary for algorithmic differentiation should be used. The data type `TMoFInterval` enables the use of `TMoValenciaIntegrator`, an adjustment of the basic version of VALENCIA-IVP. The newly developed `TMoRiotIntegrator` is based on the IVP solver from the library RIOT, an independent C++ version of COSY and COSY VI, and requires the class `TMoTaylorModel`, a SMARTMOBILE-compatible wrapper of the library's own data type `TaylorModel`. Analogously, to be able to use an adjustment of COSY VI, the wrapper `RDAInterval` is necessary. Modification of the latter solver for SMARTMOBILE is currently work in progress.

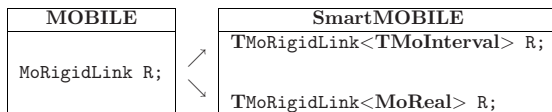
In general, kinematics can be simulated with the help of all of the above mentioned basic data types. However, other basic data types might become necessary for more specific tasks such as finding of equilibrium states of a system since they require specific solvers. SMARTMOBILE provides an option of modeling equilibrium states in a validated way with the help of the interval-based data type `MoFInterval` and the class `MoIGradientStaticEquilibriumFinder`, a version of the zero-finding algorithm from the C-XSC TOOLBOX.

**Table 1.** Basic validated data types and the corresponding solvers in SMARTMOBILE.

Data type	Solver
<code>class TMoInterval{   INTERVAL Enclosure;   TINTERVAL TEnclosure;   TFINTERVAL TEnclosure;}</code>	<code>TMoAWAIntegrator</code>
<code>TMoFInterval= {double, INTERVAL, FINTERVAL}</code>	<code>TMoValenciaIntegrator</code>
<code>TMoTaylorModel={TaylorModel}</code>	<code>TMoRiotIntegrator</code>
<code>RDAInterval={Cosy}</code>	—
<code>MoFInterval= {INTERVAL, FINTERVAL}</code>	<code>MoIGradientStaticEquilibriumFinder</code>

The availability of several basic data types in SMARTMOBILE points out its second feature: the general data type independency through its template structure. That is, `MoReal` is actually replaced with a placeholder and not with a concrete data type. For example, the transmission element `MoRigidLink` from MOBILE is replaced with

its template equivalent `TMOrigidLink`, the content of the placeholder for which (e.g. `TMOInterval` or `MoReal`, cf. Fig. 1) can be defined at the final stage of the system assembly. This allows us to use a suitable pair consisting of the data type and solver depending on the application at hand. If only a reference about the form of the solution is necessary, `MoReal` itself and a common numerical solver (e.g. Runge-Kutta's) can be used. If a relatively fast validation of dynamics without much uncertainty in parameters is of interest, `TMOInterval` and `TMOAWAIntegrator` might be the choice. For validation of highly nonlinear systems with a considerable uncertainty, the slower combination of `TMOTaylorModel` and `TMOriOTIntegrator` can be used.



**Fig. 1.** Template usage.

A `MOBILE` user can easily switch to `SMARTMOBILE` because the executable programs for the models in both environments are similar (cf. Fig. 4). In the validated environment, the template syntax should be used. The names of transmission elements are the same aside from the preceding letter `T`. The methods of the classes have the same names, too. Only the solvers are, of course, different, although they follow the same naming conventions.

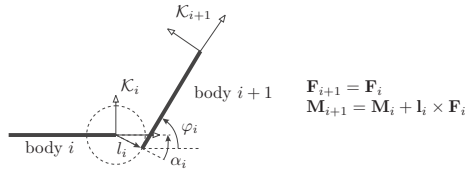
## 4 Applications

In this Section, we consider applications of `SMARTMOBILE` to validating kinematics and dynamics of mechanical systems with uncertainties. For kinematic simulations, a five arm manipulator is chosen because it also provides a comparison to stochastic methods of handling uncertainty. The opportunities `SMARTMOBILE` offers for validated simulation of dynamics are illustrated by a double pendulum. More close to life systems are treated in [23], [24], [25]. After that we determine equilibrium states of the same double pendulum to demonstrate how validated techniques improve non-validated results. Finally, we give an outlook on the application of `SMARTMOBILE` to biomechanics.

### 4.1 Validated Kinematics of a Five Arm Manipulator

The modeling of the five arm manipulator, the system defined in detail in [26], can be enhanced in `SMARTMOBILE` by using so called sloppy joints [27] instead of usual revolute ones. In the transmission element responsible for the modeling of the sloppy joint, it is no longer assumed that the joint connects the rotational axes of two bodies exactly concentrically. Instead, the relative distance between the axes is supposed to be within a specific (small) range. Two additional parameters are necessary to describe the sloppy joint (cf. Fig. 2): radius  $l_i \in [0; l_{max}]$  and the relative orientation angle  $\alpha_i \in$

$[0; 2\pi)$  (the parameter  $\varphi_i$  that describes the relative orientation between two connected bodies is the same both for the sloppy and the usual joint).



**Fig. 2.** A sloppy joint.

**Table 2.** Widths of the position enclosures.

	x	y	CPU (s)
TMoInterval	1.047	1.041	0.02
TMoTaylorModel	0.163	0.290	0.14

The considered system consists of five arms of the lengths  $l_0 = 6.5\text{m}$ ,  $l_1 = l_2 = l_3 = 4\text{m}$ , and  $l_4 = 3.5\text{m}$ , each with the uncertainty of  $\pm 1\%$ . The arms are connected with five sloppy joints, for which  $l_{max} = 2\text{mm}$  and initial angle constellation is  $\varphi_0 = 60^\circ$ ,  $\varphi_1 = \varphi_2 = \varphi_3 = -20^\circ$ , and  $\varphi_4 = -30^\circ$ . Each of these angles has an uncertainty of  $\pm 0.1^\circ$ .

In Fig. 3, left, the (abridged) SMARTMOBILE model of the manipulator is shown (the system geometry described above is omitted). First, all coordinate frames are defined with the help of the array `K`. Analogously, the required rigid links `L` and sloppy joints `R`, with which arms and their connections are modeled, are declared. They are defined later inside the `for` loop. The array `l` characterizes the lengths of the rigid links, and `phi` is used to define the angles of sloppy joints. All elements are assembled into one system using the element `manipulator`. By calling the method `doMotion()` for this element, we can obtain the position of the tip of the manipulator, which equals the rotational matrix `R` multiplied by the translational vector `r`, both stored in the last coordinate frame `K[10]`.

We simulate kinematics with the help of intervals and Taylor models. That is, the placeholder `type` is either `TMoInterval` or `TMoTaylorModel`. Both position enclosures are shown in Fig. 3, right. Note that enclosures obtained with intervals are wider than those obtained with Taylor models (cf. also Table 2, where the widths of the corresponding intervals are shown). Taylor models are bounded by intervals to provide a comparison. Numbers are rounded up to the third digit after the decimal point. CPU times are measured on a Pentium 4, 3.0 GHz PC under `CYWIN`.

The statistic analysis of the same system with the help of the Monte-Carlo method [28] carried out in [29] shows that the results, especially those for Taylor models, are acceptable. Although the set of all possible positions obtained statistically with the help of 50,000 simulations (solid black area in Fig. 3) is not as large as even the rectangle obtained with `TMoTaylorModel`, there might exist parameter constellations which lead to the results from this rectangle. Besides, statistical simulations require a lot of

CPU time, which is not the case with SMARTMOBILE. Additionally, the results are proven to be correct there through the use of validated methods.

## 4.2 Validated Dynamics of a Double Pendulum

The next example is the double pendulum with an uncertain initial angle of the first joint from [8]. We study the dynamics of the double pendulum using a SMARTMOBILE model from Fig. 4, where it is shown in comparison to the corresponding model from MOBILE. The lengths of both massless arms of the pendulum are equal to 1m and the two point masses amount to 1kg each with the gravitational constant  $g = 9.81 \frac{\text{m}}{\text{s}^2}$ . The initial values for angles (specified in rad) and angular velocities (in  $\frac{\text{rad}}{\text{s}}$ ) are given as

$$\left[ \frac{3\pi}{4} \pm 0.01 \cdot \frac{3\pi}{4} \quad -\frac{11\pi}{20} \quad 0.43 \quad 0.67 \right]^T.$$

The interval enclosures of the two angles  $\beta_1$  and  $\beta_2$  of the double pendulum are shown for identical time intervals in Fig. 5. Besides, Table 3 summarizes the results. The line “Break-down” contains the time in seconds after which the corresponding method no longer works. That is, the correctness of results cannot be guaranteed after that point. This happens here due to the chaotic character of the considered system and the resulting overestimation. The last line indicates the CPU time (in seconds) which the solvers take to obtain the solution over the integration interval  $[0; 0.4]$ . Note that the CPU times are provided only as a rough reference since the solvers can be further optimized in this respect.

The use of TMoValenciaIntegrator improves both the tightness of the resulting enclosures and the CPU time in comparison to TMoAWAIntegrator for this example. Although TMoRiOTIntegrator breaks down much later than the both former solvers, it needs a lot of CPU time.

**Table 3.** Performance of validated integrators for the double pendulum over  $[0; 0.4]$ .

Strategy	TMoAWAIntegrator	TMoRiOTIntegrator	TMoValenciaIntegrator
Break-down	0.424	0.820	0.504
CPU time	1248	9312	294

## 4.3 Equilibrium of a Double Pendulum

We consider the previous example once again but without the uncertainty in  $\beta_1$ . To find equilibrium states of this system, we use the basic data type MoFInterval instead of TMoInterval and apply MoIGradientStaticEquilibriumFinder to the element manipulator instead of using TMoMechanicalSystem and an integrator. All four possible equilibria (stable and unstable) are found by the validated solver MoIGradientStaticEquilibriumFinder in the starting interval  $[-3.15; 1]$  for all coordinates (shown rounded up to the third digit after the decimal point):

$$\begin{array}{ll} 1: ([-3.142; -3.142]; [-3.142; -3.142]) & 3: ([-0.000; 0.000]; [-3.142; -3.142]) \\ 2: ([-3.142; -3.142]; [-0.000; 0.000]) & 4: ([-0.000; 0.000]; [-0.000; 0.000]) \end{array}$$

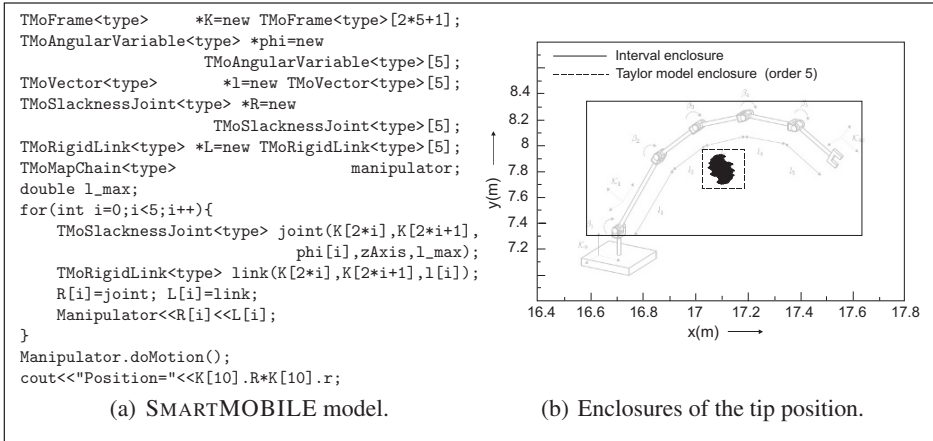


Fig. 3. Kinematics of the five arm manipulator with uncertain parameters.

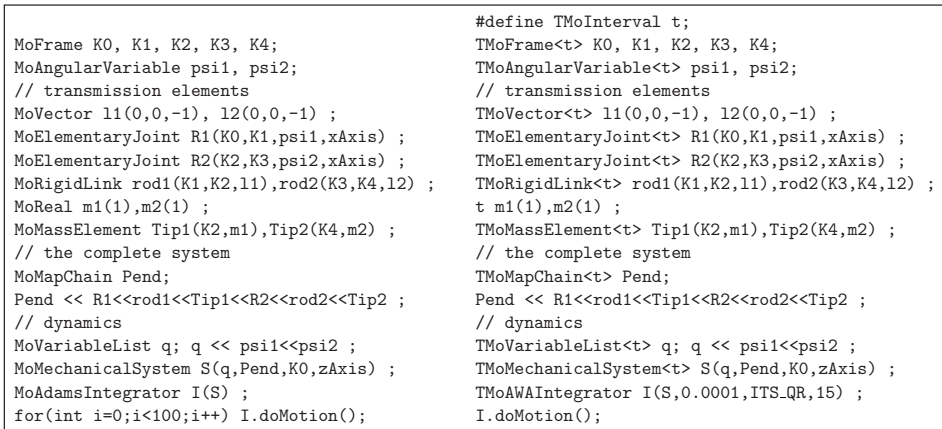


Fig. 4. The double pendulum in MOBILE (left) and SMARTMOBILE (right).

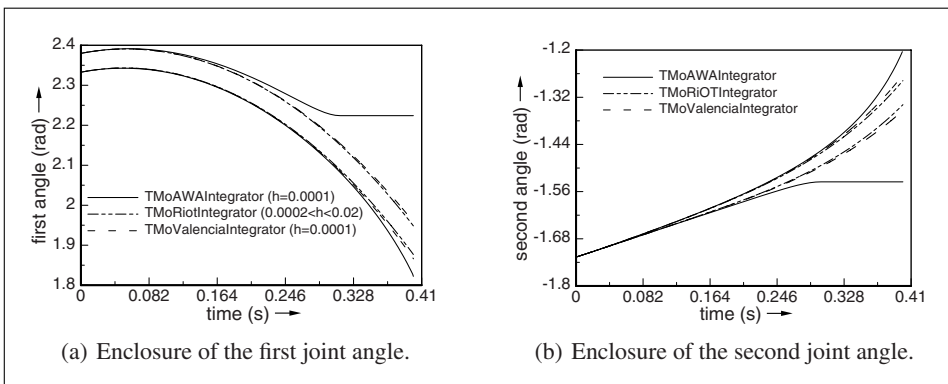


Fig. 5. Interval enclosures for the first and second state variable of the double pendulum.

Since we do not have any uncertainties in the model, the intervals obtained are very close to point intervals, that is,  $\underline{\beta}_i \approx \overline{\beta}_i$ . The difference is noticeable only after the 12-th digit after the decimal point. Note that if the same problem is modeled using the non-verified model in MOBILE, only one (unstable) equilibrium state  $[\beta_1, \beta_2] = [3.142; -3.142]$  is obtained (using the identical initial guess).

#### 4.4 Outlook: Validating a Simplified Muscle Activation Model

One of the possible application areas of validated methods is biomechanics. Especially in the context of surgical interventions, the information about the contribution of a single muscle to joint moments during motion can enable the physician to assess a therapy before applying it to a patient. This is a still open problem in biomechanics. Typical solutions usually have high computation times, which make them unsuitable for online motion approximation. Recently, a fast method has been developed to roughly identify muscle activation profiles [30]. However, the approach produces results which differ from the prototype ones.

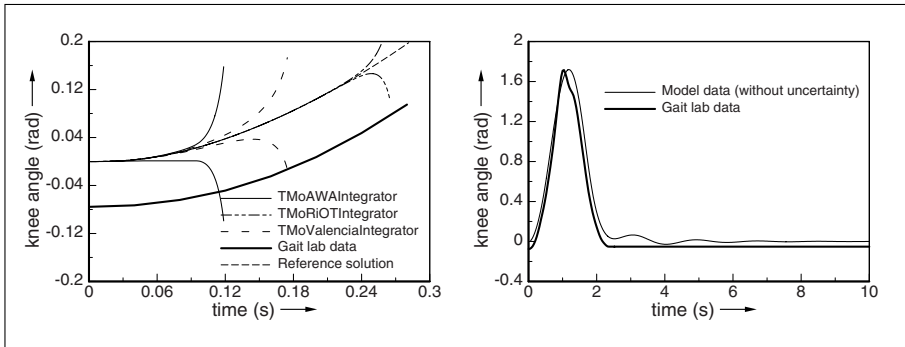
Since most of the model parameters cannot be measured exactly, and the above mentioned approach does not take the uncertainty in consideration, this might be a possible discrepancy source. On the other hand, the deviation might also result from the inadequately chosen muscle models. Considerable model simplifications performed for the first validated study do not allow us to make certain statements about that yet. However, a step in this direction is the validated analysis the influence of small changes in parameters on the model behavior in order to determine a critical set.

The model under investigation represents a simplified subsystem of the human leg described in more detail in [23]. It consists of pelvis, thigh and shank. To drive the model in forward dynamics simulations, the muscle *biceps femoris short head* is included, which is responsible for knee flexion. The force law of the involved muscle model is roughly simplified with respect to the HILL-type muscle model. Moreover, for the purposes of the first validated study, the overall model is simplified so that it is everywhere continuously differentiable.

The model behavior was studied using the same solvers as in Subsection 4.2. In Fig. 6, right, the discrepancy between validated simulations without model uncertainty and gait lab data is illustrated. On the left of the same Figure, the considerable influence of the  $\pm 0.1\%$  uncertainty in the thigh length on the overall simulation results is demonstrated. In [23], we showed preliminarily that the most sensitive parameters were the thigh length, the shank length and the  $z$  coordinate of the proximal muscle insertion point. In particular, the first parameter should be measured exactly up to the order of a micrometer to have little influence on the system. Future research in this direction will include a more formal validated sensitivity analysis of the model with respect to all parameters.

## 5 Conclusions

In this paper, we presented the tool SMARTMOBILE for guaranteed modeling and simulation of kinematics and dynamic of mechanical systems. With its help, the behavior of



**Fig. 6.** Interval enclosures of the knee angle with (left) and without(right)  $\pm 0.1\%$  uncertainty in the thigh length.

different classes of systems can be obtained with the guarantee of correctness, the option which is not given by tools based on floating point arithmetics. Besides, the uncertainty in parameters can be taken into account in a natural way. Moreover, SMARTMOBILE is flexible and allows the user to choose the kind of underlying arithmetics according to the task at hand. The tool was applied to four mechanical problems.

The main directions of the future development will include enhancement of validated options for modeling and simulation of closed-loop systems in SMARTMOBILE as well as integration of further verified solvers into its core.

**Acknowledgements.** This work is carried out in the project TellHiM&S funded by the German Research Council. Thanks to Professor A. Kecskeméthy for providing the source code of MOBILE and general cooperation.

## References

1. Kecskeméthy, A.: Objektorientierte Modellierung der Dynamik von Mehrkörpersystemen mit Hilfe von Übertragungselementen. PhD thesis, Gerhard Mercator Universität Duisburg (1993)
2. Huckle, T.: Software Bugs, [www5.in.tum.de/~huckle/bugse.html](http://www5.in.tum.de/~huckle/bugse.html) (2005)
3. Arnold, D.N.: Some Disasters Attributable to Bad Numerical Computing, [www.ima.umn.edu/~arnold/disasters/](http://www.ima.umn.edu/~arnold/disasters/) (1998)
4. Moore, R.E.: Interval Analysis. Prentice-Hall, New York (1966)
5. de Figueiredo, L.H., Stolfi, J.: Affine Arithmetic: Concepts and Applications. Numerical Algorithms 37 (2004) 147–158
6. Neumaier, A.: Taylor Forms — Use and Limits. Reliable Computing 9 (2002) 43–79
7. Lohner, R.: On the Ubiquity of the Wrapping Effect in the Computation of the Error Bounds. In Kulisch, U., Lohner, R., Facius, A., eds.: Perspectives on Enclosure Methods, Springer Wien New York (2001) 201–217
8. Auer, E., Rauh, A., Hofer, E.P., Luther, W.: Validated Modeling of Mechanical Systems with SMARTMOBILE: Improvement of Performance by VALENCIA-IVP. In: Proc. of Dagstuhl Seminar 06021: Reliable Implementation of Real Number Algorithms: Theory and Practice. Lecture Notes in Computer Science (2006) To appear.

9. Knüppel, O.: PROFIL/BIAS — A Fast Interval Library. *Computing* 53 (1994) 277–287
10. Lerch, M., Tischler, G., Wolff von Gudenberg, J., Hofschuster, W., Krämer, W.: The Interval Library *filib++ 2.0*: Design, Features and Sample Programs. Technical Report 2001/4, Bergische Universität GH Wuppertal (2001)
11. Klatte, R., Kulisch, U., Wiethoff, A., Lawo, C., Rauch, M.: *C–XSC: A C++ Class Library for Extended Scientific Computing*. Springer-Verlag (1993)
12. Stolfi, J.: *LIBAFFA*, <http://savannah.nongnu.org/projects/libaffa> (2003)
13. Berz, M., Makino, K.: *COSY INFINITY Version 8.1. User's Guide and Reference Manual*. Technical Report MSU HEP 20704, Michigan State University (2002)
14. Hammer, R., Hocks, M., Kulisch, U., Ratz, D.: *C++ Toolbox for Verified Computing I - Basic Numerical Problems*. Springer-Verlag, Heidelberg and New York (1995)
15. Lohner, R.: *Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen*. PhD thesis, Universität Karlsruhe (1988)
16. Nedialkov, N.S.: *The Design and Implementation of an Object-Oriented Validated ODE Solver*. Kluwer Academic Publishers (2002)
17. Berz, M., Makino, K.: *Verified Integration of ODEs and Flows Using Differential Algebraic Methods on High-Order Taylor Models*. *Reliable Computing* 4 (1998) 361–369
18. Griewank, A.: *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM (2000)
19. Stauning, O., Bendtsen, C.: *FADBAD++*, [www.fadbad.com](http://www.fadbad.com) (2005)
20. Griewank, A., Juedes, D., Utke, J.: *ADOL–C, A Package for the Automatic Differentiation of Algorithms Written in C/C++*. *ACM Trans. Math. Software* 22(2) (1996) 131–167
21. Nedialkov, N.S.: *Computing Rigorous Bounds on the Solution of an Initial Value Problem for an Ordinary Differential Equation*. PhD thesis, University of Toronto (1999)
22. Kecskeméthy, A., Hiller, M.: *An Object-oriented approach for an effective formulation of multibody dynamics*. *CMAME* 115 (1994) 287–314
23. Auer, E., Tändl, M., Strobach, D., Kecskeméthy, A.: *Toward Validating a Simplified Muscle Activation Model in SMARTMOBILE*. In: *CD Proceedings of SCAN 2006, IEEE Computer Society* (2007)
24. Auer, E.: *Interval Modeling of Dynamics for Multibody Systems*. In: *Journal of Computational and Applied Mathematics*, Elsevier (2006) Online.
25. Auer, E., Kecskeméthy, A., Tändl, M., Traczinski, H.: *Interval Algorithms in Modeling of Multibody Systems*. In Alt, R., Frommer, A., Kearfott, R., Luther, W., eds.: *LNCS 2991: Numerical Software with Result Verification*, Springer, Berlin Heidelberg New York (2004) 132 – 159
26. Traczinski, H.: *Integration von Algorithmen und Datentypen zur validierten Mehrkörpersimulation in MOBILE*. PhD thesis, University of Duisburg-Essen (2006)
27. Hörsken, C., Traczinski, H.: *Modeling of Multibody Systems with Interval Arithmetic*. In Krämer, W., Wolff von Gudenberg, J., eds.: *Scientific Computing, Validated Numerics, Interval Methods*, Dordrecht, Kluwer Academic Publishers (2001) 317–328
28. Metropolis, N., Ulam, S.: *The Monte Carlo Method*. *Journal of the American Statistical Association* 44 (1949) 335–341
29. Hörsken, C.: *Methoden zur rechnergestützten Toleranzanalyse in Computer Aided Design und Mehrkörpersystemen*. PhD thesis, University of Duisburg-Essen (2003)
30. Strobach, D., Kecskeméthy, A., Steinwender, G., Zwick, B.: *A Simplified Approach for Rough Identification of Muscle Activation Profiles via Optimization and Smooth Profile Patches*. In: *CD Proceedings of the International ECCOMAS Thematic Conference on Advances in Computational Multibody Dynamics*, Madrid, Spain, ECCOMAS (June 21 – 24 2005)



# Path Planning for Cooperating Unmanned Vehicles over 3-D Terrain

Ioannis K. Nikolos and Nikos C. Tsourveloudis

Intelligent Systems and Robotics Laboratory, Department of Production Engineering and Management, Technical University of Crete, University Campus, 73100, Chania, Greece  
jnikolo@dpem.tuc.gr, nikost@dpem.tuc.gr

**Abstract.** In this paper we suggest an off-line/on-line path planner for cooperating unmanned vehicles that takes into account the mission objectives and constraints through an optimization procedure. The cooperating vehicles can be either Unmanned Aerial Vehicles (UAVs) or Autonomous Underwater Vehicles (AUVs); these two categories of vehicles share common features as far as path planning is concerned and these features are used in this work for the development of a unified approach to the path planning problem over 3-D terrains. A number of unmanned vehicles of the same category are launched from the same or different known initial locations. The main issue is to produce 3-D trajectories (represented by 3-D B-Spline curves) that ensure a collision free path, respect the mission objectives and constraints, and guide the vehicles to a common final destination. The off-line planner is designed for known environments. The on-line one generates paths in unknown static environments, by exchanging acquired information from the cooperating vehicles' on-board sensors. For each vehicle a near optimum path is generated that guides it safely to an intermediate position within the already scanned area. The process is repeated for each vehicle until the final destination is reached by one or more members of the team. Then, each one of the remaining vehicles can either turn into the off-line mode to reach the target, moving through the already scanned area, or continue with the on-line mode. Both off-line and on-line path planning problems are formulated as optimization problems, and a Differential Evolution algorithm is used as the optimizer.

**Keywords.** 3-D Path Planning, Navigation, Vehicles Cooperation, UAVs, AUVs, Evolutionary Algorithms, Differential Evolution, B-Splines.

## 1 Introduction

Path planning is the generation of a space path between an initial location and the desired destination that has an optimal or near-optimal performance under specific constraints [1]. The main concerns during the comparison of various candidate solutions are *feasibility* and *optimality* [2]. Searching for optimality is not a trivial task and in most cases results in non-affordable computation time, even in simple problems. Therefore, in most cases we search for suboptimal or just feasible solutions.

In this work the path planning for cooperating unmanned vehicles moving over a 3-D terrain is considered; the vehicles can be either Unmanned Aerial Vehicles

(UAVs) or Autonomous Underwater Vehicles (AUVs). UAVs and AUVs share the common feature of performing inside a 3-D environment and having six degrees of freedom, although their kinematic characteristics are not the same. The upper ceiling for AUVs is the sea surface, while a similar upper ceiling exists for UAVs due to stealth considerations or flight envelop restrictions.

Path planning for UAVs and AUVs imply special characteristics that have to be considered [3], [4], [5], such as: (a) physical feasibility, (b) performance related to mission, (c) real-time implementation, (d) cooperation between the vehicles, (e) stealth (low observability due to the selected path). Besides their common features, differences also exist between the two categories, as far as coordination and path planning is concerned, which are mainly related with the different sensors and electronic equipment that are needed in order to cooperate and perform their mission.

Cooperation between robotic vehicles has gained recently an increased interest as systems of multiple vehicles engaged in cooperative behavior show specific benefits compared to a single one [6] [7].

Path planning problems are computationally demanding multi-objective multi-constraint optimization problems [8]. The problem complexity increases when multiple vehicles should be used. Various approaches have been reported for UAVs coordinated route planning, such as Voronoi diagrams [9], mixed integer linear programming [10], [11] and dynamic programming [12] formulations.

In Beard et al. [9] the motion-planning problem was decomposed into a waypoint path planner and a dynamic trajectory generator. The path-planning problem was solved via a Voronoi diagram and Eppstein's k-best paths algorithm, while the trajectory generator problem was solved via a real-time nonlinear filter.

In [13] the motion-planning problem for a limited resource of Mobile Sensor Agents (MSAs) is investigated, in an environment with a number of targets larger than the available MSAs. The problem is formulated as an optimization one, whose objective is to minimize the average time duration between two consecutive observations of each target.

Computational intelligence methods, such as Neural Networks [14], Fuzzy Logic [15] and Evolutionary Algorithms (EAs) [5], [16] have been successfully used to produce trajectories for guiding mobile robots in known, unknown or partially known environments. Besides their computational cost, EAs are considered as a viable candidate to solve path planning problems effectively; the reasons are their high robustness, their ease of implementation, and their high adaptability to different optimization problems, with or without constraints [16].

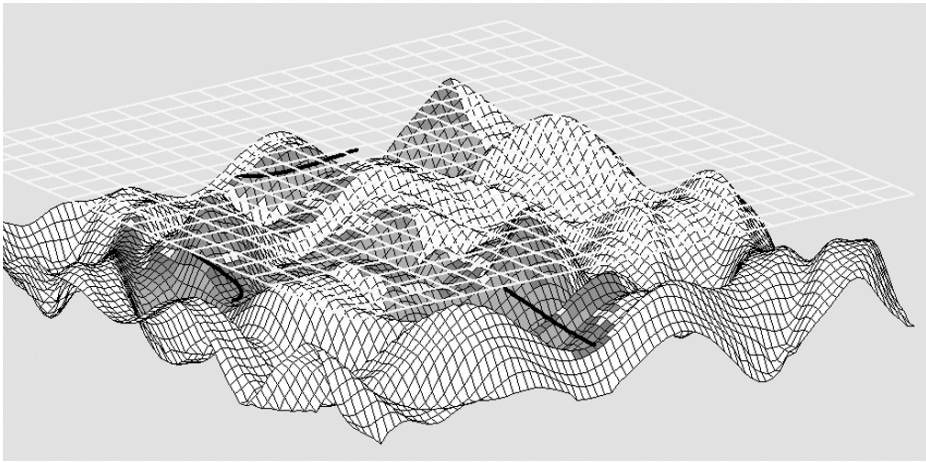
EAs have been successfully used in the past for the solution of the path-finding problem in ground based or sea surface navigation [17], [18], [19], or for solving the path-finding problem in a 3-D environment for underwater vehicles [20], [21].

Changwen Zheng et al. [5] proposed a route planner for UAVs, based on evolutionary computation. The generated routes enable the vehicles to arrive at their destination simultaneously by taking into account the exposure of UAVs to potential threats. The flight route consists of straight-line segments, connecting the way points from the starting to the goal points. The cost function penalizes the route length the high altitude flights or routes that come dangerously close to known ground threats.

In [22] a multi-task assignment problem for cooperating UAVs is formulated as a combinatorial optimization problem; a Genetic Algorithm is utilized for assigning the multiple agents to perform various tasks on multiple targets.

In [16] an EA based framework was utilized to design an off-line / on-line path planner for UAVs autonomous navigation. The path planner calculates a curved path line, represented using B-Spline curves in a 3-D terrain environment. The on-line planner gradually produces a smooth 3-D trajectory aiming at reaching a predetermined target in an unknown environment; the produced trajectory consists of smaller B-Spline curves smoothly connected to each other.

In this work the following scenario was considered: having a number of autonomous vehicles (either UAVs or AUVs), at the same or different known initial locations with predefined initial directions, we calculate 3-D smooth trajectories, which connect the initial locations with a single destination location, ensuring a collision free operation with respect to mission constraints. Each vehicle is assumed to be a point and its actual size is taken into account by equivalent obstacle growing.



**Fig. 1.** A representation of the proposed concept: three vehicles are moving along curved path lines over a 3-D terrain; an upper ceiling is enforced (either sea surface or the maximum allowed flying height); on-board sensors are scanning the environment within a certain range in front of each vehicle.

Initially the off-line planner will be presented; it generates collision free paths in environments with known characteristics and flight restrictions. The on-line planner, being an extension of the off-line one, was developed to generate collision free paths in unknown environments. As each vehicle moves towards its destination, its on-board sensors are scanning the environment within a certain range and certain angles; this information is exchanged between the members of the team, resulting in a gradual mapping of the environment (Fig. 1). The on-line planner uses the acquired knowledge of the environment to generate a near optimum path for each vehicle that will guide it safely to an intermediate position within the known territory. The process is repeated until the corresponding final position is reached by one or more members of the team. Then, each one of the remaining members of the team either uses the off-line planner to compute a path that connects its current position and the final destination, or continues in the on-line mode until it reaches the common destination. Both path planning problems are formulated as minimization problems, where

specially constructed functions take into account mission and cooperation objectives and constraints, with a Differential Evolution algorithm to serve as the optimizer.

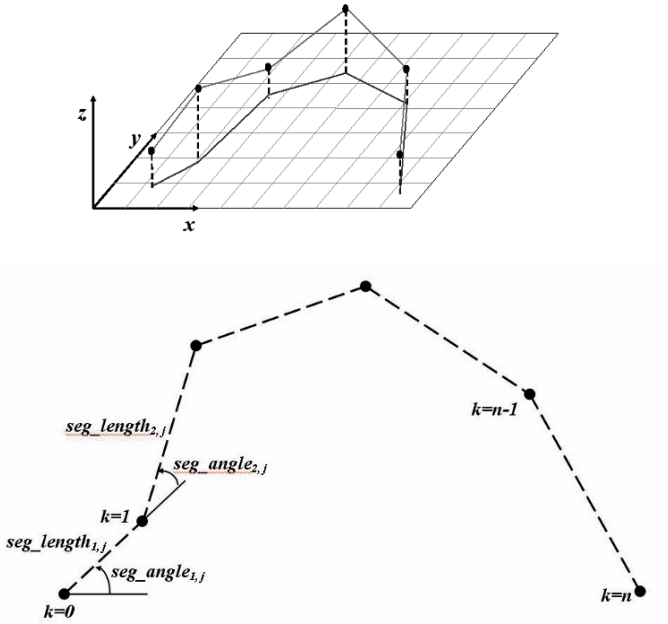
The rest of the paper is organized as follows: in section 2 the off-line path planner for a single vehicle will be briefly discussed. Section 3 deals with the concept of on-line path planning for cooperating vehicles. The problem formulation is described, including assumptions, objectives, constraints, cost function definition and path modeling. Simulations results are presented in section 4, followed by discussion in section 5.

## 2 Off-Line Path Planner

The off-line planner generates collision free paths in environments with known characteristics and flight restrictions, where the solid boundaries are interpreted as 3-D surfaces. The derived path line for each vehicle is a single continuous 3-D B-Spline curve with fixed starting and ending control points. A third point, placed in a pre-specified distance from the starting one, is also fixed, determining the initial flight direction for the corresponding vehicle. Between the fixed control points, free-to-move control points determine the shape of the curve. For each path, the number of the free-to-move control points is user-defined.

Straight line segments that connect a number of way points have been used in the past to model UAV paths in 2D or 3D space [23], [5]. However, these simplified paths cannot be used for an accurate simulation of UAV's flight, unless a large number of way points is used. In [9], paths from the initial vehicle location to the target location are derived from a graph search of a Voronoi diagram that is constructed from the known threat locations. The resulting paths, consisting of line segments, are subsequently smoothed around each way point. Dubins [24] car formulation has been proposed as an alternative approach to the modeling of UAV dynamics [25]. This approach seems inefficient to model scenarios including 3D terrain avoidance and following of stealthy routes. However, this approach seems to be sufficient enough for task assignment purposes to cooperating UAVs flying at safe altitudes [13], [22], [25].

B-Spline curves have been used in the past for trajectory representation in 2-D [26] or in 3-D environments [16], [27]. They are well fitted in an optimization procedure as they need a few variables (the coordinates of their control points) to define complicated curved paths [28], [29]. The use of B-Spline curves for the determination of a path-line provides the advantage of describing complicated non-monotonic 3-dimensional curves with controlled smoothness with a small number of design parameters, i.e. the coordinates of the control points. Another valuable characteristic of the adopted B-Spline curves is that the curve is tangential to the control polygon at the starting and ending points. This characteristic can be used in order to define the starting or ending direction of the curve, by inserting an extra fixed point after the starting one, or before the ending control point.



**Fig. 2.** Schematic representation of the B-Spline control polygon (top) and its projection on the horizontal plane (bottom).

In this work each path is constructed using a 3-D B-Spline curve; each B-Spline control point is defined by its three Cartesian coordinates  $x_{k,j}$ ,  $y_{k,j}$ ,  $z_{k,j}$  ( $k=0, \dots, n$ ,  $j=1, \dots, N$ ,  $N$  being the number of vehicles, while  $n+1$  is the number of control points in each B-Spline curve, the same for all curves). The first ( $k=0$ ) and last ( $k=n$ ) control points of the control polygon are the initial and target points of the  $j^{\text{th}}$  UAV, which are predefined by the user. The second ( $k=1$ ) control point is positioned in a pre-specified distance from the first one, in a given altitude, and in a given direction, in order to define the initial direction of the corresponding path.

The control polygon of each B-Spline curve is defined by successive straight line segments (Fig. 2). Each segment of the control polygon is defined using its projection on the horizontal plane (Fig. 2); the length  $seg\_length_{k,j}$  and the direction  $seg\_angle_{k,j}$  of this projection are used as design variables ( $k=2, \dots, n-1$ ,  $j=1, \dots, N$ ). Design variables  $seg\_angle_{k,j}$  are defined as the difference between the direction (in deg) of the current segment's projection and the projection of the previous one. For the first segment ( $k=1$ ) of each control polygon  $seg\_angle_{1,j}$  is measured with respect to the  $x$ -axis (Fig. 2). Additionally, the control points' altitudes  $z_{k,j}$  are used as design variables, except for the three fixed points ( $k=0$ ,  $k=1$ , and  $k=n$ ), which are predefined. For the first segment ( $k=1$ ),  $seg\_length_{1,j}$  and  $seg\_angle_{1,j}$  are pre-specified in order to define the initial direction of the path, and they are not included in the design variables of the optimization procedure.

The horizontal coordinates of each B-Spline control point  $x_{k,j}$  and  $y_{k,j}$  can be easily calculated by using  $seg\_length_{k,j}$  and  $seg\_angle_{k,j}$  along with the coordinates of the previous control point  $x_{k-1,j}$  and  $y_{k-1,j}$ . The use of  $seg\_length_{k,j}$  and  $seg\_angle_{k,j}$  as design

variables instead of  $x_{k,j}$  and  $y_{k,j}$  was adopted for three reasons. The first reason is the fact that abrupt turns of each flight path can be easily avoided by explicitly imposing short lower and upper bounds for the  $seg\_angle_{k,j}$  design variables. The second reason is that by using the proposed design variables a better convergence rate was achieved compared to the case with the B-Spline control points' coordinates  $(x_{k,j}, y_{k,j}, z_{k,j})$  as design variables. The latter observation is a consequence of the shortening of the search space, using the proposed formulation. The third reason is that by using  $seg\_length_{k,j}$  as design variables, an easier determination of the upper bound for each curve's length is achieved, along with a smoother variation of the lengths of each curve's segments. The lower and upper boundaries of each independent design variable are predefined by the user.

For the case of a single vehicle the optimization problem to be solved minimizes a set of five terms, connected to various objectives and constraints; they are associated with the feasibility of the curve, its length and a safety distance from the ground. The cost function to be minimized is defined as:

$$f = \sum_{i=1}^5 w_i f_i \quad (1)$$

Term  $f_1$  penalizes the non-feasible curves that pass through the solid boundary. In order to compute this term, discrete points along each curve are computed, using B-Spline theory [28] [29] and a pre-specified step for B-Spline parameter  $u$ . The value of  $f_1$  is proportional to the number of discrete curve points located inside the solid boundary. Term  $f_2$  is the length of the curve (non-dimensional with the distance between the starting and destination points) and is used to provide shorter paths. Term  $f_3$  is designed to provide flight paths with a safety distance from solid boundaries. For each discrete point  $i$  ( $i=1, \dots, nline$ , where  $nline$  is the number of discrete curve points) of the B-Spline curve its distance from the ground is calculated (the ground is described by a mesh of  $n_{ground}$  discrete points). Then the minimum distance of the curve and the ground  $d_{min}$  is computed. Term  $f_3$  is then defined as:

$$f_3 = \left( d_{safe} / d_{min} \right)^2, \quad (2)$$

while  $d_{safe}$  is a safety distance from the solid boundary.

Term  $f_4$  is designed to provide B-Spline curves with control points inside the pre-specified space. If a control point results with an  $x$  or  $y$  coordinate outside the pre-specified limits, a penalty is added to term  $f_4$  which is proportional to the violation of the following constraints:

$$\begin{aligned} \text{if } x_{k,j} > x_{max} &\Rightarrow f_4 = f_4 + C_1 * |x_{k,j} - x_{max}| \\ \text{if } y_{k,j} > y_{max} &\Rightarrow f_4 = f_4 + C_1 * |y_{k,j} - y_{max}| \\ \text{if } x_{k,j} < x_{min} &\Rightarrow f_4 = f_4 + C_1 * |x_{k,j} - x_{min}| \\ \text{if } y_{k,j} < y_{min} &\Rightarrow f_4 = f_4 + C_1 * |y_{k,j} - y_{min}| \\ \forall k, k = 0, \dots, n, \quad \forall j, j = 1, \dots, N, \end{aligned} \quad (3)$$

where  $C_1$  is a constant, and  $x_{min}, x_{max}, y_{min}, y_{max}$  define the borders of the working space. An additional penalty is added to  $f_4$  in case that its value is greater than zero, in order



to ensure that curves inside the pre-specified space have a smaller cost function than those having control points outside of it. This can be formally written as

$$\text{if } f_4 > 0 \Rightarrow f_4 = f_4 + C_2, \quad (4)$$

where  $C_2$  is a constant.

Term  $f_3$  was designed to provide path lines within the already scanned terrain. Each control point of the B-Spline curve is checked for whether it is placed over a known territory. The ground is modeled as a mesh of discrete points and the algorithm computes the mesh shell (on the  $x$ - $y$  plane) that includes each B-Spline control point. If the corresponding mesh shell is characterized as unknown then a constant penalty is added to  $f_3$ . A mesh shell is characterized as unknown if all its 4 nodes are unknown (have not been detected by a sensor).

Weights  $w_i$  are experimentally determined, using as criterion the almost uniform effect of the last four terms in the objective function. Term  $w_i f_i$  has a dominant role in Eq. 1 providing feasible curves in few generations, since path feasibility is the main concern. The minimization of Eq. 1 results in a set of B-Spline control points, which actually represent the desired path.

For the solution of the minimization problem a Differential Evolution (DE) [30] algorithm is used. The classic DE algorithm evolves a fixed size population, which is randomly initialized. After initializing the population, an iterative process is started and at each generation  $G$ , a new population is produced until a stopping condition is satisfied. At each generation, each element of the population can be replaced with a new generated one. The new element is a linear combination between a randomly selected element and the difference between two other randomly selected elements. A detailed description of the DE algorithm used in this work can be found in [31].

### 3 On-Line Path Planning for Cooperating Vehicles

The on-line path planner was designed for navigation and collision avoidance of a small team of autonomous vehicles moving over a completely unknown static 3-D terrain. The general constraint of the problem is the collision avoidance between the vehicles and the ground. The route constraints are: (a) predefined initial and target coordinates for all vehicles, (b) predefined initial directions for all vehicles, (c) predefined minimum and maximum limits of allowed-to-move space. The first two route constraints are explicitly taken into account by the optimization algorithm. The third route constraint is implicitly handled by the algorithm, through the cost function. The cooperation objective is that all members of the team should reach the same target point.

The on-line planner is based on the ideas developed in [16] for a single UAV. The on-line planner rapidly generates a near optimum path, modeled as a 3-D B-Spline curve that will guide each vehicle safely to an intermediate position within the already scanned area. The information about the already scanned area by each vehicle is passed to the rest cooperating vehicles, in order to maximize the knowledge of the environment. The process is repeated until the final position is reached by one or more members of the team (it is possible some members of the team to reach simultaneously the target – in the same number of on-line steps). Then the rest members of the team turn into the off-line mode and a single B-Spline path for each

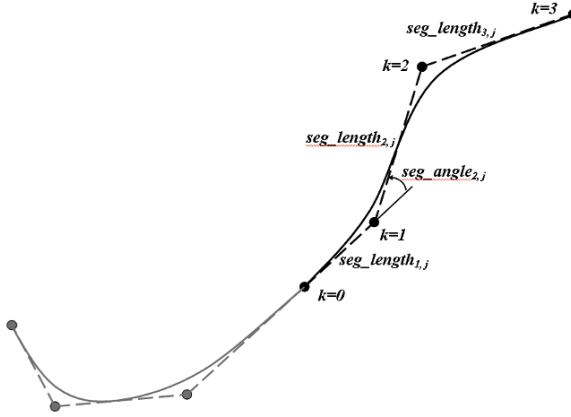
vehicle is computed to guide it from its current position, through the already scanned territory to the common final destination. An alternative approach, which was also tested, is to keep the remaining vehicles in the on-line mode, and not to turn into the off-line mode after a vehicle has reached the target.

In the on-line problem only four control points define each B-Spline curve, the first two of which are fixed and determine the direction of the path of the current vehicle. The remaining two control points are allowed to take any position within the already scanned space, taking into consideration given constraints. The second control is used to make sure that at least first derivative continuity of the two connected curves is provided at their common point. Hence, the second control point of the next curve should lie on the line defined by the last two control points of the previous curve (Fig. 3). The design variables that define each B-Spline segment are the same as in the off-line case, i.e.  $seg\_length_{k,j}$ ,  $seg\_angle_{k,j}$ , and  $z_{k,j}$  ( $k=2, 3$ , and  $j=1, \dots, N$ ).

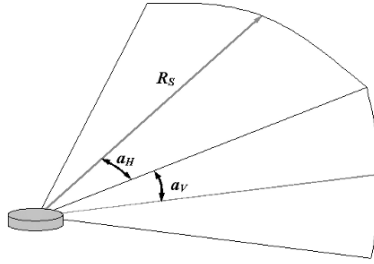
The path-planning algorithm considers the scanned surface as a group of quadratic mesh nodes. All ground nodes are initially considered unknown. An algorithm is used to distinguish between nodes visible by the on-board sensors and nodes not visible. The algorithm uses a predefined range  $R_s$  for each sensor as well as two angles, one for the horizontal  $a_h$  and one for the vertical scanning  $a_v$  (Fig. 4). The range and the two angles are predefined by the user and depend on the type of the sensors used. A node is not visible by a sensor if it is not within the sensor's range and angles of sight, or if it is within the sensor's range and angles of sight but is hidden by a ground section that lies between it and the vehicle. The corresponding algorithm, simulates the sensor and checks whether the ground nodes within the sensor's range are "visible" or not and consequently "known" or not. If a newly scanned node is characterized as "visible", it is added to the set of scanned ground nodes, which is common for all cooperating vehicles.

The information from its sensors is used to produce the first path line segment for the corresponding vehicle. As the vehicle is moving along its first segment and until it has traveled about 3/4 of its length, its sensor scans the surrounding area, returning a new set of visible nodes, which are subsequently added to the common set of scanned nodes. This (simulated) scanning is performed for 11 intermediate positions along each path segment. The on-line planner, then, produces a new segment for each vehicle, whose first point is the last point of the previous segment and whose last point lies somewhere in the already scanned area, its position being determined by the on-line procedure. The on-line process is repeated until the ending point of the current path line segment of one vehicle lies close to the final destination. Then the rest members of the team either can turn into the off-line process, in order to reach the target using B-Spline curves that pass through the scanned terrain, or may remain in the on-line mode.





**Fig. 3.** Schematic representation of the formation of the complete path by successive B-Spline segments (projected on the horizontal plane).



**Fig. 4.** Schematic representation of the scanned area in front of each vehicle;  $a_H$  and  $a_V$  are the solid angles in the horizontal and vertical directions that define the scanned sector.

The position at which the algorithm starts to generate the next path line segment for each vehicle (here taken as the 3/4 of the segment length) depends on the range of the sensors, vehicle's velocity and the computational demands of the algorithm. The computation of intermediate path segments for each vehicle is formulated as a minimization problem. The cost function to be minimized is formulated as the weighted sum of seven different terms

$$f = \sum_{i=1}^7 w_i f_i, \quad (5)$$

where  $w_i$  are the weights and  $f_i$  are the corresponding terms described below.

Terms  $f_1$ ,  $f_2$ , and  $f_3$  are similar to terms  $f_p$ ,  $f_s$ , and  $f_4$  respectively of the off-line procedure. Term  $f_1$  penalizes the non-feasible curves that pass through the solid boundary. Term  $f_2$  is designed to provide flight paths with a safety distance from solid boundaries. Only already scanned ground points are considered for this calculation. Additionally, the points that are lower than a pre-specified (small) vertical distance from the current level of flight are not considered for this calculation. Term  $f_3$  is

designed to provide B-Spline curves with control points inside the pre-specified working space.

Term  $f_4$  is designed to provide flight segments with their last control point having a safety distance from solid boundaries. This term was introduced to ensure that the next path segment will not start very close to a solid boundary (which may lead to infeasible paths or paths with abrupt turns). The minimum distance  $D_{min}$  from the ground is calculated for the last control point of the current path segment. Only already scanned ground points are considered for this calculation. As in term  $f_2$  the points that are lower than a pre-specified (small) vertical distance from the current level of flight are not considered for this calculation. Term  $f_4$  is then defined as

$$f_4 = \left( d_{safe} / D_{min} \right)^2, \quad (6)$$

while  $d_{safe}$  is a safety distance from the solid boundary.

The value of term  $f_5$  depends on the potential field strength between the current starting point (of the corresponding path segment) and the final target. This potential field between the two points is the main driving force for the gradual development of each path line in the on-line procedure. The potential is similar to the one between a source and a sink, defined as

$$\Phi = \ln \frac{r_2 + c \cdot r_0}{r_1 + c \cdot r_0}, \quad (7)$$

where  $r_1$  is the distance between the last point of the current curve and the initial point for the current curve segment,  $r_2$  is the distance between the last point of the current curve and the final destination,  $r_0$  is the distance between the initial point of the current curve and the final destination and  $c$  is a constant. This potential allows for selecting curved paths that bypass obstacles lying between the starting and ending point of each B-Spline curve [16].

Term  $f_6$  is designed to prevent the vehicles from being trapped in small regions and to force them move towards unexplored areas. Term  $f_6$  repels it from the points of the already computed path lines (of all vehicles). This term has the form

$$f_6 = \frac{1}{N_{po}} \sum_{k=1}^{N_{po}} \frac{1}{r_k}, \quad (8)$$

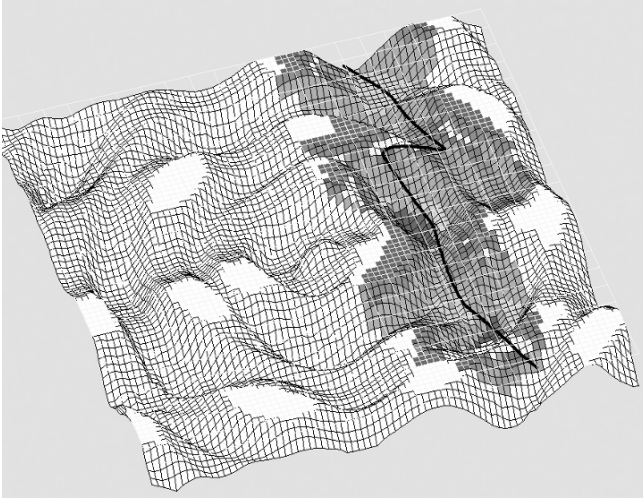
where  $N_{po}$  is the number of the discrete curve points produced so far by all vehicles and  $r_k$  is their distance from the last point of the current curve segment.

Term  $f_7$  represents another potential field, which is developed around the final target and has the form

$$f_7 = r_2^2, \quad (9)$$

where  $r_2$  is the distance between the last point of the current curve and the final destination. Thus, when the vehicle is near its target, the value of this term is quite small and prevents the vehicle from moving away.

Weights  $w_i$  in Eq. 5 are experimentally determined, using as criterion the almost uniform effect of all the terms, except the first one. Term  $w_1 f_1$  has a dominant role, in order to provide feasible curve segments in a few generations, since path feasibility is the main concern.

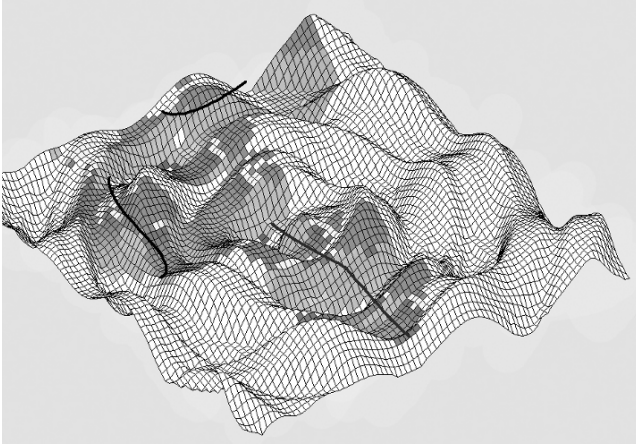


**Fig. 5.** Test Case 1: On-line path planning for a single UAV. The maximum allowed height for the vehicle is shown using a cutting plane.

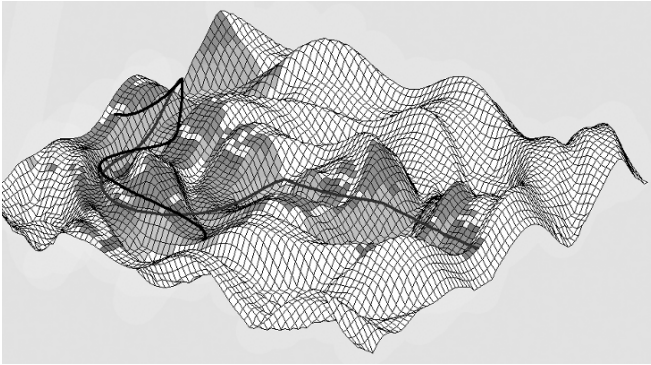
## 4 Simulation Results

The same artificial environment was used for all the test cases considered, with different starting and target points. The artificial environment is constructed within a rectangle of  $20 \times 20$  (non-dimensional distances). The (non-dimensional) range of the sensors ( $R_s$ ) that scan the environment was set equal to 4 for all vehicles. The safety distance from the ground was set equal to  $d_{safe} = 0.25$ . The (experimentally optimized) settings of the Differential Evolution algorithm during the on-line procedure were as follows: *population size* = 20,  $F = 0.6$ ,  $C_r = 0.45$ , *number of generations* = 70. For the on-line procedure we have two free-to-move control points, resulting in 6 design variables. The corresponding settings during the off-line procedure were as follows: *population size* = 30,  $F = 0.6$ ,  $C_r = 0.45$ , *number of generations* = 70. For the off-line procedure eight control points were used to construct each B-Spline curve (including the initial ( $k=0$ ) and the final one ( $k=7$ )). These correspond to five free-to-move control points, resulting in 15 design variables. All B-Spline curves have a degree equal to 3.

All experiments have been designed in order to search for path lines between “mountains”. For this reason, an upper ceiling has been enforced in the optimization procedure, by explicitly providing an upper boundary for the  $z$  coordinates of all B-Spline control points. Test Case 1 corresponds to the on-line path planning for a single vehicle over an unknown environment (Fig. 5). The horizontal and vertical angles  $a_h$  and  $a_v$ , used for the sensor’s simulation, were set equal to 45 degrees. The complete path consists of 6 B-Spline segments; the final curve is smooth enough to be followed by a vehicle. The first turn in the path line is due to the presence of an obstacle (solid ground) in front of the vehicle (Fig. 5); the second turn forces the vehicle towards its final destination.

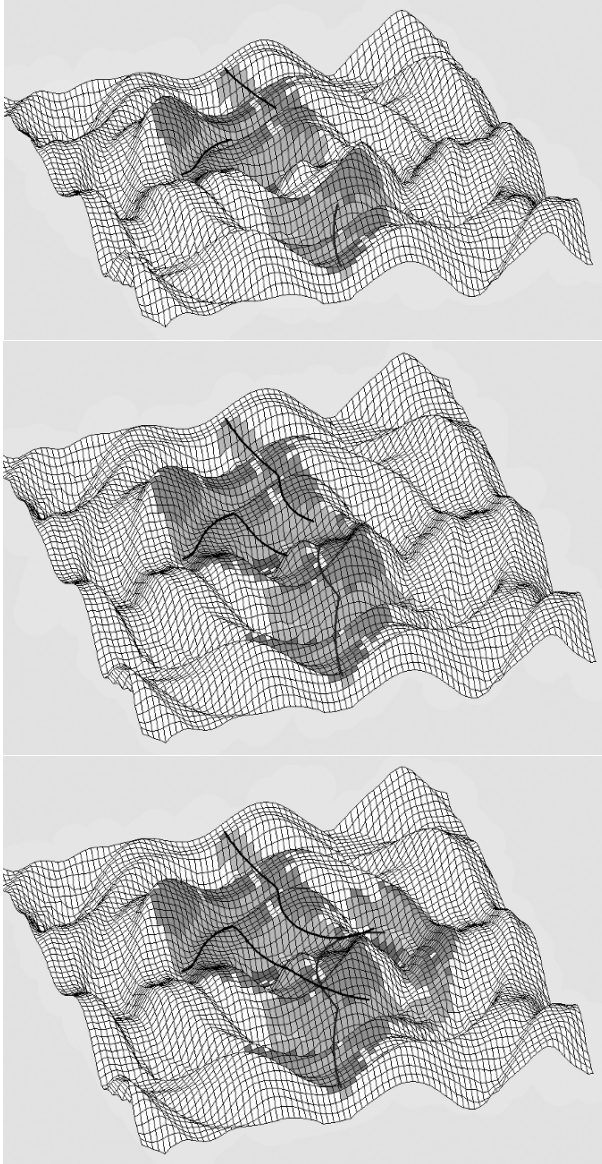


**Fig. 6.** Test Case 2 corresponds to the on-line path planning for 3 vehicles. The picture shows the status of the path lines when the first vehicle (near the upper corner) reaches the target.



**Fig. 7.** The final status of the path lines of Test Case 2. The off-line path planner was used by the remaining vehicles to drive them, from their current position to the final destination, through already scanned area.

Test Case 2 corresponds to the on-line path planning for 3 unmanned vehicles (Fig. 1, 6, and 7). The horizontal and vertical angles  $a_H$  and  $a_V$ , used for the sensor's simulation were set equal to 45 and 30 degrees respectively. Figure 1 shows the status of the three path lines when the first line segment has been computed for all three vehicles. Figure 6 shows the status of the three path lines when the first vehicle reaches the target, after two steps in the on-line procedure. The final status is demonstrated in Fig. 7; the remaining two vehicles turn into off-line mode to reach the target. A curved path is computed for each one of the remaining vehicles, which drives the vehicle from its current position to the target, through the already scanned area.

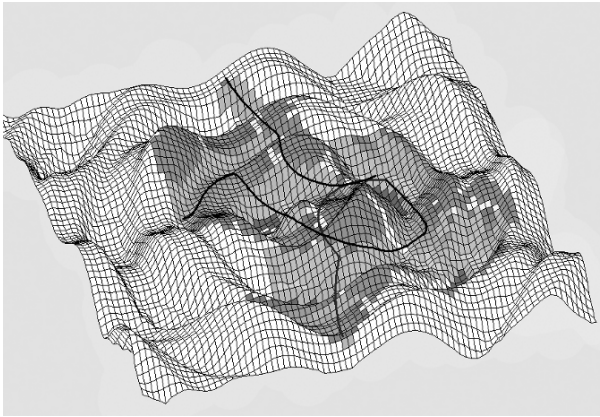


**Fig. 8.** Test Case 3: Successive snapshots of the path-line for three vehicles, computed using only the on-line planner. Two of the vehicles are reaching the target using 3 segments.

An alternative strategy was considered in Test Case 3. Instead of turning into off-line mode when a vehicle (or more) is reaching the target, the on-line path planner is always used to guide all vehicles to the target. In this case three vehicles are considered. The horizontal and vertical angles  $a_H$  and  $a_V$ , used for the sensor's simulation were set equal to 45 degrees for both angles. Figure 8 contains successive snapshots of the path lines produced using the on-line path planner. As it can be



observed the two vehicles arrive to the target after the same number of steps (Fig. 8). Two more steps of the procedure are needed for the third vehicle to reach the target (Fig. 9).



**Fig. 9.** Test Case 3: Two more steps are needed for the third vehicle to reach the target using the on-line path planner.

## 5 Discussion

The proposed methodology is applicable to cooperating UAVs but also to cooperating AUVs; in the later case the enforced upper ceiling of the searching space will be the sea surface. Actually, in the case of AUVs the application of the proposed methodology might be easier, due to the lower speed of an AUV compared to an UAV and due to the dynamics of such vehicles. However, in the case of AUVs the suit of the on board sensors will be completely different and the knowledge of the environment will be based on sonar-type sensors. Concerning the application of the proposed methodology to cooperating UAVs, the VTOL type of UAVs seems to be the best choice. The main reason is that the hovering capability of a helicopter may provide the necessary additional time to overcome a computational intensive problem during the calculation of successive curve segments. Additionally, a helicopter has a higher capability to handle abrupt turns, compared to a fixed-wing UAV.

Two issues have to be considered for the application of the proposed methodology to real world scenarios. The first one is the lack of lightweight radar sensors, capable to fit into small UAVs (like small helicopters). Although radar sensors for indoor applications have been already presented (with an effective range of some meters), there is a need for lightweight radar sensors with a range of hundreds of meters, with a weight suitable for small UAVs. The second issue is the communication between the cooperating vehicles. The communication devices should be capable to securely transfer an amount of data (related to the scanned territory by each vehicle), between all cooperating vehicles. Available RF connections for UAV applications are adequate enough for the problem at hand. Acoustic communication links should be used for the communication between AUVs.

## References

1. Gilmore, J.F.: Autonomous Vehicle Planning Analysis Methodology. In: Association of Unmanned Vehicles Systems Conference. Washington, DC, pp. 503–509 (1991)
2. LaValle, S.M.: Planning Algorithms. Cambridge University Press (2006)
3. Bortoff, S.: Path Planning for UAVs. In: Amer. Control Conf., Chicago, IL, pp. 364–368 (2000)
4. Szczerba, R.J., Galkowski, P., Glickstein, I.S., and Ternullo, N.: Robust Algorithm for Real-Time Route Planning. *IEEE Trans. on Aerosp. Electr. Syst.* 36, 869–878 (2000)
5. Zheng, C., Li, L., Xu, F., Sun, F., Ding, M.: Evolutionary Route Planner for Unmanned Air Vehicles. *IEEE Trans. on Rob.* 21, 609–620 (2005)
6. Uny Cao, Y., Fukunaga, A.S., Kahng, A.B.: Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots*, 4, 7–27(1997)
7. Schumacher, C.: Ground Moving Target Engagement by Cooperative UAVs. In: 2005 American Control Conference, June 8-10, Portland, OR, USA (2005)
8. Mettler, B., Schouwenaars, T., How, J., Paunicka, J., and Feron E.: Autonomous UAV Guidance Build-up: Flight-Test Demonstration and Evaluation Plan. In: AIAA Guidance, Navigation, and Control Conference, AIAA-2003-5744 (2003)
9. Beard, R.W., McLain, T.W., Goodrich, M.A., Anderson, E.P.: Coordinated Target Assignment and Intercept for Unmanned Air Vehicles. *IEEE Trans. on Rob. and Autom.* 18 911–922 (2002)
10. Richards, A., Bellingham, J., Tillerson, M., and How., J.: Coordination and Control of UAVs. In: AIAA Guidance, Navigation and Control Conference, Monterey, CA, (2002)
11. Schouwenaars, T., How, J., and Feron, E.: Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA-2004-5141 (2004)
12. Flint, M., Polycarpou, M., and Fernandez-Gaucherand, E.: Cooperative Control for Multiple Autonomous UAV's Searching for Targets. In: 41st IEEE Conference on Decision and Control (2002)
13. Tang, Z., and Ozguner, U.: Motion Planning for Multi-Target Surveillance with Mobile Sensor Agents. *IEEE Trans. on Rob.* 21, 898–908 (2005)
14. Gomez Ortega, J., and Camacho, E.F.: Mobile Robot Navigation in a Partially Structured Static Environment, using Neural Predictive Control. *Control Eng. Practice*, 4, 1669–1679 (1996)
15. Kwon, Y.D., and Lee, J.S.: On-Line Evolutionary Optimization of Fuzzy Control System based on Decentralized Population. *Intelligent Automation and Soft Computing*, 6, 135–146 (2000)
16. Nikolos, I.K., Valavanis, K.P., Tsourveloudis, N.C., Kostaras, A.: Evolutionary Algorithm Based Offline / Online Path Planner for UAV Navigation. *IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics*, 33, 898-912 (2003)
17. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer (1999)
18. Smierzchalski, R.: Evolutionary Trajectory Planning of Ships in Navigation Traffic Areas. *Journal of Marine Science and Technology*, 4, 1–6 (1999)
19. Smierzchalski, R., and Michalewicz Z.: Modeling of Ship Trajectory in Collision Situations by an Evolutionary Algorithm. *IEEE Trans. on Evol. Comp.* 4, 227–241 (2000)
20. Sugihara, K., and Smith, J.: Genetic Algorithms for Adaptive Motion Planning of an Autonomous Mobile Robot. In: 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, California, 138–143 (1997)
21. Sugihara, K., and Yuh, J.: GA-Based Motion Planning for Underwater Robotic Vehicles. UUST-10, Durham, NH (1997)

22. Shima, T., Rasmussen, S.J., Sparks, A.G.: UAV Cooperative Multiple Task Assignments using Genetic Algorithms. In: 2005 American Control Conference, June 8-10, Portland, OR, USA (2005)
23. Moitra, A., Mattheyses, R.M., Hoebel, L.J., Szczerba, R.J., Yamrom, B.: Multivehicle Reconnaissance Route and Sensor Planning. *IEEE Trans. on Aerospace and Electronic Syst.* 37, 799–812 (2003)
24. Dubins, L.: On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Position. *Amer. J. of Math.* 79, 497–516 (1957)
25. Shima, T., Schumacher, C.: Assignment of Cooperating UAVs to Simultaneous Tasks Using Genetic Algorithms. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco (2005)
26. Martinez-Alfaro H., and Gomez-Garcia, S.: Mobile Robot Path Planning and Tracking using Simulated Annealing and Fuzzy Logic Control. *Expert Systems with Applications*, 15, 421–429 (1988)
27. Nikolos, I.K., Tsourveloudis, N., and Valavanis, K.P.: Evolutionary Algorithm Based 3-D Path Planner for UAV Navigation. In: 9th Mediterranean Conference on Control and Automation, Dubrovnik, Croatia (2001)
28. Piegl, L., Tiller, W.: *The NURBS Book*. Springer (1997)
29. Farin, G.: *Curves and Surfaces for Computer Aided Geometric Design, a Practical Guide*. Academic Press (1988)
30. Price, K.V., Storn, R.M., Lampinen, J.A.: *Differential Evolution, a Practical Approach to Global Optimization*. Springer-Verlag, Berlin Heidelberg (2005)
31. Nikolos, I.K., Tsourveloudis, N., Valavanis, K.: Evolutionary Algorithm Based Path Planning for Multiple UAV Cooperation. In: Valavanis, K. (ed.), *Advances in Unmanned Aerial Vehicles, State of the Art and the Road to Autonomy*, pp. 309–340. Springer (2007)



# Tracking of Manoeuvring Visual Targets

C. Pérez<sup>1</sup>, N. García<sup>1</sup>, J. M. Sabater<sup>1</sup>, J. M. Azorín<sup>1</sup> and L. Gracia<sup>2</sup>

<sup>1</sup> Miguel Hernández University, Avda. de la Universidad S/N, 03202-Elche, Spain  
{carlos.perez, nicolas.garcia, j.sabater, jm.azorin}@umh.es  
<http://www.isa.umh.es/vr2>

<sup>2</sup> Technical University of Valencia, Camino de Vera S/N, 46022-Valencia, Spain  
luigraca@isa.upv.es

**Abstract.** Tracking of manoeuvring visual targets or visual features is an important issue in robotic vision. For this purpose, predictive techniques are used. In this sense, this work presents a new predictor that decreases the tracking error compared with classic filters for abrupt motion changes and can be used for unknown object's dynamics. The proposed predictor is based on a fuzzy mix of several *Kalman* filters, but it can be extended to other algorithms like Circular Predictors. This fuzzy mix depends on what filter is working closer to their optimal settings. The performance and robustness of the proposed algorithm is verified by simulation and experiment and it is compared with other robust methods.

## 1 Introduction

During the last few years, the use of visual servoing and visual tracking has been more and more common due to the increasing power of algorithms and computers.

Visual servoing and visual tracking are techniques that can be used to control a mechanism according to visual information. This visual information is available with a time delay, therefore, the use of predictive algorithms are widely extended (notice that prediction of the object's motion can be used for smooth movements without discontinuities).

The *Kalman* filter [1] has become a standard method to provide predictions and solve the delay problems (considered the predominant problem of visual servoing) in visual based control systems [2], [3] and [4].

The time delay is one of the bigger problems in this type of systems. For practically all processing architectures, the vision system requires a minimum delay of two cycles, but for on-the-fly processing, only one cycle of the control loop is needed [5].

Authors of [6] demonstrate that steady-state *Kalman* filters ( $\alpha\beta$  and  $\alpha\beta\gamma$  filters) performs better than the KF in the presence of abrupt changes in the trajectory, but not as good as the KF for smooth movements. Some research works about the motion estimation are presented in [7] and [8]. Further, some motion understanding and trajectory planning based on the *Frenet-Serret* formula are described in [9], [10] and [11]. Using the knowledge of the motion and the structure, identification of the target dynamics may be accomplished.

To solve delay problems, taking into account these considerations, we propose a new prediction algorithm. This new filter can be called Fuzzy predictor. This filter minimizes

the tracking error and works better than the classic KF because it decides what of the used filters ( $\alpha\beta^{slow}/\alpha\beta^{fast}$  [5],  $\alpha\beta\gamma$ ,  $Kv$ ,  $Ka$  and  $Kj$ ) must be employed. The transition between them is smooth avoiding discontinuities.

These five filters should be used in a combination because: The *Kalman* filter is considered one of the reference algorithms for position prediction (but we must consider the right model depending on the object's dynamics: velocity–acceleration–jerk). When the object is outside the image plane, the best prediction is given by steady-state filters ( $\alpha\beta/\alpha\beta\gamma$  depending on the object's dynamics: velocity–acceleration). Obviously, considering more filters and more behaviour cases, the Fuzzy predictor can be improved but computational cost of additional considerations can be a problem in real-time execution. These five filters are considered by authors as the best consideration (solution taking into account the prediction quality and the computational cost). This is the reason to combine these five filters to obtain the Fuzzy predictor.

This work is focused on the new Fuzzy prediction filter and is structured as follows: in section 2 we present the considered dynamics, the considered dynamics is a Jerk model with adaptable parameters obtained by KFs [12], [13] and [14]. In section 3, we present the block diagram for the visual servoing task. This block diagram is widely used in several works like [2] or [5]. Section 4 presents the basic idea applied in our case (see [15] and [15]), but the main work done is focused in one of the blocks described in section 3, the Fuzzy predictor is described in section 5.

In section 6, we can see the results with simulated data. These results show that the Fuzzy predictor can be used to improve the high speed visual servoing tasks. This section is organized in two parts: in the first one (Subsection 6.1), the analysis of the Fuzzy predictor behaviour is focussed and in the second one (Subsection 6.2) their results are compared those with achieved by Chroust and Vince [5] and with CPA [16] algorithm (algorithm used for aeronautic/aerospace applications). Conclusions and future work are presented in section 7.

## 2 The Dynamics of a Moving Object

The object's movement is not known (a priori) in a general visual servoing scheme. Therefore, it is treated as an stochastic disturbance justifying the use of a KF as a stochastic observer. The KF algorithm presented by *Kalman* [1] starts with the system description given by 1 and 2.

$$x_{k+1} = F \cdot x_k + G \cdot \xi_k \quad (1)$$

$$y_k = C \cdot x_k + N \cdot \eta_k \quad (2)$$

where  $x_k \in \mathbb{R}^{n \times 1}$  is the state vector and  $y_k \in \mathbb{R}^{m \times 1}$  is the output vector. The matrix  $F \in \mathbb{R}^{n \times m}$  is the so-called system matrix which describes the propagation of the state from  $k$  to  $k+1$  and  $C \in \mathbb{R}^{m \times n}$  describes the way in which the measurement is generated out of the state  $x_k$ . In our case of visual servoing  $m$  is 1 (because only the position is measured) and  $n = 4$ . The matrix  $G \in \mathbb{R}^{n \times 1}$  distributes the system noise  $\xi_k$  to the states and  $\eta_k$  is the measurement noise. In the KF the noise sequences  $\eta_k$  and  $\xi_k$  are assumed

to be gaussian, white and uncorrelated. The covariance matrices of  $\xi_k$  and  $\eta_k$  are  $Q$  and  $R$  respectively (these expressions consider 1D movement). A basic explanation for the assumed gaussian white noise sequences is given in [17].

In the general case of tracking, the usual model considered is a constant acceleration model [5], but in our case, we consider a constant jerk model described by matrices  $F$  and  $C$  are:

$$F = \begin{bmatrix} 1 & T & T^2/2 & T^3/6 \\ 0 & 1 & T & T^2/2 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad C = [1 \ 0 \ 0 \ 0]$$

where  $T$  is the sampling time. This model is called a *constant jerk model* because it assumes that the jerk ( $dx^3(t)/dt^3$ ) is constant between two sampling instants.

$F$  and  $C$  matrices are obtained from expression 3 to 7:

$$\frac{a - a_i}{t - t_i} = \frac{\Delta a}{\Delta t} = J_0 \quad (3)$$

$$x(t) = x_i + v_i(t - t_i) + \frac{1}{2}a_i(t - t_i)^2 + \frac{1}{6}J_0(t - t_i)^3 \quad (4)$$

$$v(t) = v_i + a_i(t - t_i) + \frac{1}{2}J_0(t - t_i)^2 \quad (5)$$

$$a(t) = a_i + J_0(t - t_i) \quad (6)$$

$$J(t) = J_0 \quad (7)$$

where,  $x$  is the position,  $v$  is the velocity,  $a$  is the acceleration and  $J$  is the jerk. So the relation between them is:

$$x(t) = f(t); \quad \dot{x}(t) = v(t); \quad \ddot{x}(t) = a(t); \quad \dddot{x}(t) = J(t)$$

### 3 Description of the Control System

The main objective of the visual servoing is to bring the target to a position of the image plane and to keep it there for any object's movement. In Fig. 1 we can see the visual control loop presented by Corke in [2]. The block diagram can be used for a moving camera and for a fixed camera controlling the motion of a robot. Corke use a KF to incorporate a feed-forward structure. We incorporate the Fuzzy prediction algorithm in the same structure (see Fig. 2) but reordering the blocks for an easier comprehension.

$V(z)$  in Fig. 2 represents the camera behaviour, which is modeled as a simple delay:  $V(z) = k_v \cdot z^{-2}$  (see [2], [18], [19], [20] and [21]).  $C(z)$  is the controller (A simple proportional controller is implemented in experiments presented in this work).  $R(z)$  is the robot (for this work:  $R(z) = z/z - 1$ ) and the *Prediction filter* generates the feedforward signal by prediction the position of the target. The variable for been minimized is  $\Delta x$  (generated by the vision system) that represents the deviation of the target respect to the desired position (error). The controller calculates a velocity signal  $\dot{x}_d$  which moves

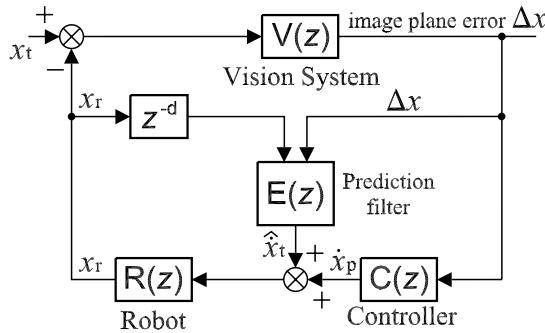


Fig. 1. Operation diagram presented by Corke using KF for the  $E(z)$  block.

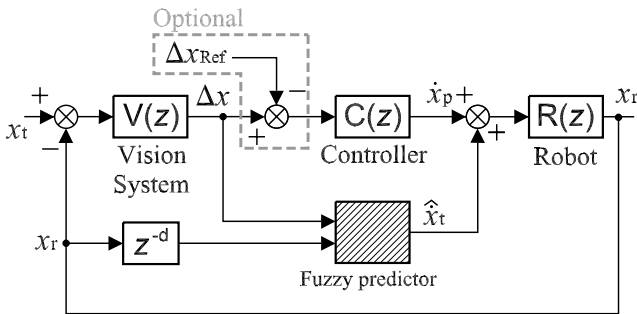


Fig. 2. Operation diagram using the Fuzzy predictor.

the robot in the right direction to decrease the error. Using this approach, no path planning is needed (the elimination of this path planning is important because it decreases the computational load [2]).

The transfer function of the robot describes the behaviour from the velocity input to the position reached by the camera, which includes a transformation in the image plane. Therefore, the transfer function considered is [5]:

$$R(z) = \frac{z}{z - 1}$$

The Fuzzy predictor block is explained in the next sections (sections 4 and 5).

## 4 Theoretical Background of the Fuzzy Predictor

The most common fuzzy inference process used is known as Mamdani’s fuzzy inference method, but on the other hand, we can find a so-called *Sugeno*, or *Takagi-Sugeno-Kang*, method of fuzzy inference. It was introduced in 1985 [22] and is similar to the Mamdani’s method in many respects. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The

main difference between Mamdani and *Sugeno* is that the *Sugeno* output membership functions are either linear or constant (for more information see [23]).

For *Sugeno* regulators, we have a linear dynamic system as the output function so that the  $i^{th}$  rule has the form:

If  $\tilde{z}_1$  is  $\tilde{A}_1^j$  and  $\tilde{z}_2$  is  $\tilde{A}_2^k$  and, ..., and  $\tilde{z}_p$  is  $\tilde{A}_p^l$  Then  $\dot{x}^i(t) = U_i x(t) + V_i u(t)$

where  $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$  is the state vector,

$u(t) = [u_1(t), u_2(t), \dots, u_m(t)]^T$ ,  $U_i$  and  $V_i$  are the state and input matrices and  $z(t) = [z_1(t), z_2(t), \dots, z_p(t)]^T$  is the input to the fuzzy system, so:

$$\dot{x}(t) = \frac{\sum_{i=1}^R (U_i x(t) + V_i u(t)) \mu(z(t))}{\sum_{i=1}^R (\mu(z(t)))}$$

or

$$\dot{x}(t) = \left( \sum_{i=1}^R (U_i \xi_i(z(t))) \right) x(t) + \left( \sum_{i=1}^R (V_i \xi_i(z(t))) \right) u(t)$$

where

$$\xi^T = [\xi_1, \dots, \xi_R] = \frac{1}{\sum_{i=1}^R \mu_i} [\mu_1, \dots, \mu_R]$$

Our work is based on this idea and these expressions (see [23] for more details). We have mixed the Mamdani's and the *Sugenos*'s idea because we have implemented an algorithm similar to *Sugeno* but not for linear systems. We obtain a normalized weighting of several non linear recursive expressions. The system works like we can see in Fig. 3 (see section 5).

## 5 The Fuzzy Predictor

We have developed a new filter that mixes different types of *Kalman* filters depending on the conditions of the object's movement. The main advantage of this new algorithm is the non-abrupt change of the filter's output.

Consider the nonlinear dynamic system

$$\dot{x} = f_1(x, u); \quad y = g_1(x, u)$$

as each one of the filters used. The application of the fuzzy regulator in our case produces the next space-state expression:

$$\sum_{i=1}^N f_i(x, u) \cdot \omega(x, u)$$

where

$$\omega(x, u) = \frac{\mu_i(x, u)}{\sum_{i=1}^N \mu_j(x, u)}$$

The final system obtained has the same structure than filters used:

$$\dot{x} = f_2(x, u); \quad y = g_2(x, u)$$

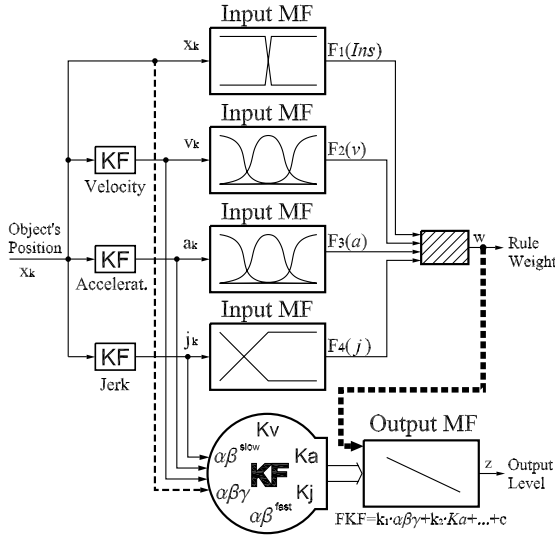


Fig. 3. Fuzzy predictor proposed.

Figure 3 shows the Fuzzy prediction block diagram (*In this work is presented a Fuzzy predictor using different types of Kalman filters, therefore in this case it can be named as Fuzzy predictor using Kalman Filters or FKF, although the Fuzzy predictor wants to be a general idea*). In this figure, we can see that the general input is the position sequence of the target ( $x_k$ ). Using this information, we estimate the velocity, acceleration and jerk of the target in three separate KFs (Nomura and Naito present the advantages of this hybrid technique in [12]). This information is used as 'Input MF' to obtain  $F_1(Ins)$ ,  $F_2(v)$ ,  $F_3(a)$  and  $F_4(j)$ . These MF inputs are the fuzzy membership functions defined in Fig. 4. The biggest KF block (rounded) shown in this figure is a combination of all used algorithms in the fuzzy filter ( $\alpha\beta^{slow}$  and  $\alpha\beta^{fast}$  [5],  $\alpha\beta\gamma$ ,  $K_v$ ,  $K_a$  and  $K_j$ ). This block obtains the output of all specified filters. The 'Output MF' calculates the final output using the  $R_i$  rules.

Now, we present the rules ( $R_i$ ) considered for the fuzzy filter:

$R_1$ : IF object IS inside AND velocity IS low AND acceleration IS low AND jerk IS low THEN Fuzzy-prediction= $K_v$

- $R_2$ : IF object IS inside AND velocity IS medium AND acceleration IS low AND jerk IS low THEN Fuzzy-prediction= $Kv$
- $R_3$ : IF object IS outside AND velocity IS low AND acceleration IS low AND jerk IS low THEN Fuzzy-prediction= $\alpha\beta^{slow}$
- $R_4$ : IF object IS outside AND velocity IS medium AND acceleration IS low AND jerk IS low THEN Fuzzy-prediction= $\alpha\beta^{fast}$
- $R_5$ : IF object IS inside AND velocity IS high AND acceleration IS low AND jerk IS low THEN Fuzzy-prediction= $Kv$
- $R_6$ : IF object IS inside AND acceleration IS medium AND jerk IS low THEN Fuzzy-prediction= $0.2 \cdot \alpha\beta\gamma + 0.8 \cdot Ka$
- $R_7$ : IF object IS outside AND acceleration IS medium AND jerk IS low THEN Fuzzy-prediction= $0.8 \cdot \alpha\beta\gamma + 0.2 \cdot Ka$
- $R_8$ : IF object IS inside AND acceleration IS high AND jerk IS low THEN Fuzzy-prediction= $Ka$
- $R_9$ : IF object IS outside AND acceleration IS high AND jerk IS low THEN Fuzzy-prediction= $\alpha\beta\gamma$
- $R_{10}$ : IF jerk IS high THEN Fuzzy-prediction= $Kj$

These rules have been obtained empirically, based on the authors experience using the *Kalman* filter in different applications.

Notice that rule  $R_{10}$  (when jerk is high) shows that the best filter considered is  $Kj$  and it does not depend on the object's position (inside or outside) velocity/acceleration value (low, medium or high).

We have used a product inference engine, singleton fuzzifier and centre average defuzzifier. Figure 4 presents the fuzzy sets definition where  $(u_{max}, v_{max})$  is the image size,  $\mu_{vel} = \mu_{acc} = 2m/s$ ,  $\sigma_{vel} = \sigma_{acc} = 0.5$ ,  $c_{vel} = c_{acc} = 1$ ,  $d_{vel} = d_{acc} = 3$ ,  $i_{vel} = i_{acc} = 1$  and  $j_{vel} = j_{acc} = 1$  (these values have been empirically obtained).

## 6 Results

This section is composed by two different parts: first (section 6.1), we analyze the prediction algorithm presented originally in this work (Fuzzy prediction block diagram shown in Fig. 3) and second (section 6.2), some simulations of the visual servoing scheme (see Fig. 2) are done including the Fuzzy prediction algorithm.

### 6.1 Fuzzy Predictor Results

In Fig. 5, we show the effectiveness of our algorithm's prediction compared with the classical KF methods. In this figure, we can see positions  $P_k^r$  (actual object position),  $P_{k-1}^r$  (object position in  $k - 1$ ) and  $P_{k-2}^r$  (object position in  $k - 2$ ). Next real position of the object will be  $P_{k+1}^r$ , and points from  $\tilde{P}_{k+1}^1$  to  $\tilde{P}_{k+1}^6$ , represent the prediction obtained by each single filter. The best prediction is given by the Fuzzy filter presented as a novelty in this work. This experiment is done for a parabolic trajectory of an object affected by the gravity acceleration. (See Fig. 5 and Fig. 6).

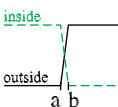

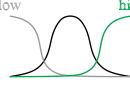
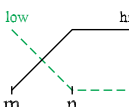
Jerk	Acceleration	Velocity	Position
 <p>inside outside a b</p> <p><math>b=(U_{max}, V_{max})</math> <math>a=0.95 \cdot (U_{max}, V_{max})</math> lineal transition</p>	 <p>medium</p> <p>normal function: <math display="block">f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}</math></p>	 <p>low high</p> <p>sigmoide function: <math display="block">f(x) = \frac{1}{1+e^{-c/(x-d)}}</math> <math display="block">f(x) = \frac{1}{1+e^{i/(x-j)}}</math></p>	 <p>low high m n</p> <p><math>m=0</math> <math>n=2 \text{ m/s}^3</math> lineal transition</p>

Fig. 4. Parameter definition of the fuzzy system.

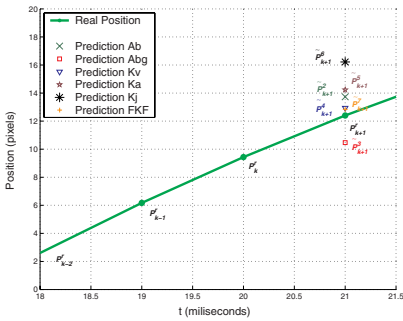


Fig. 5. Real position vs prediction

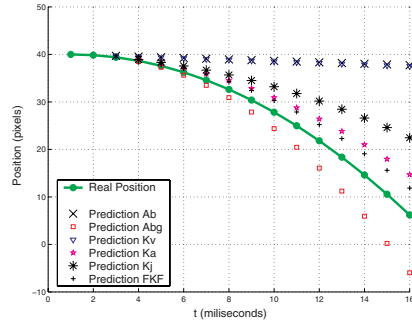


Fig. 6. Prediction of a smooth trajectory.

We have done a lot of experiments for different movements of the object and we have concluded that our Fuzzy prediction algorithm works better than the others filters compared (filters compared are:  $\alpha\beta$ ,  $\alpha\beta\gamma$ ,  $Kv$ ,  $Ka$ ,  $Kj$  and CPA -see section 6.2- with our Fuzzy predictor). Figure 6 shows the real trajectory and the trajectory predicted for each filter. For this experiment, we have used the first four real positions of the object as input for all filters and they predict the trajectory using only this information. As we can see in this figure, the best prediction is again the Fuzzy.

## 6.2 Visual Servoing Control Scheme Results

To prove the control scheme presented in Fig. 2, we have used the object motion shown in Fig. 7 (up). This target motion represents a ramp-like motion between  $1 < t < 4$  seconds and a sinusoidal motion for  $t > 6$  seconds. This motion model is corrupted with a noise of  $\sigma=1$  pixel. This motion is used by Stefan Chroust and Markus Vincze in [5] to analyze the *switching Kalman filter* (SKF).

For this experiment, we compare the proposed filter (Fuzzy) with a well known filter, the *Circular Prediction Algorithm* (CPA) [16]. In Fig. 7 (down), we can see the



results of Fuzzy predictor and CPA algorithms. For changes of motion behaviour, the Fuzzy produce less error than CPA. For the change in  $t=1$ , the error of the Fuzzy predictor is  $[+0.008,-0]$  and  $[+0.015,-0.09]$  for the CPA. For the change in  $t=4$ , Fuzzy predictor error =  $[+0,-0.0072]$  and CPA error =  $[+0.09,-0.015]$ . For the change in  $t=6$ , Fuzzy predictor error =  $[+0.022,-0]$  and CPA error =  $[+0.122,-0.76]$ . For the region  $6 < t < 9$  (sinusoidal movement between 2.5m and 0.5m) both algorithms works quite similarly: Fuzzy predictor error =  $[\pm 0.005]$  and CPA error =  $[\pm 0.0076]$ . CPA filter works well because it is designed for movements similar to a sine shape, but we can compare this results with the SKF filter proposed in [5] and SKF works better (due to the AKF (Adaptive Kalman Filter) effect). Therefore, the Fuzzy predictor filter proposed works better than CPA for all cases analyzed but comparing Fuzzy predictor with SKF, Fuzzy predictor is better for  $t=1$ ,  $t=4$  and  $t=6$  but not for  $6 < t < 9$  (sinusoidal movement).

Figure 9 shows the zoom region  $0 < t < 2$  and  $-0.02 < \Delta x_p < 0.02$  of the same experiment. In this figure, we can see the fast response of the Fuzzy predictor.

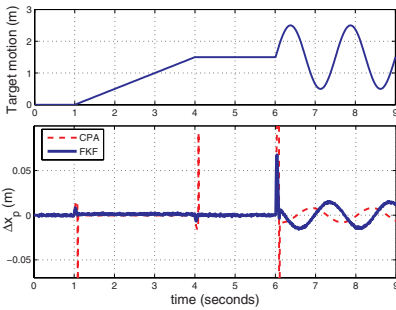


Fig. 7. Simulation result for tracking an object.

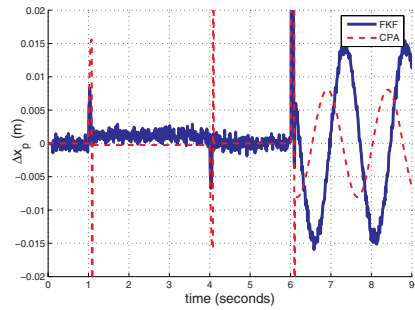


Fig. 8. Zoom of the simulation.

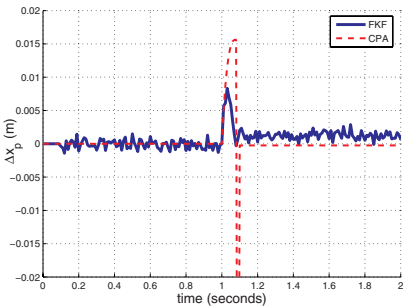


Fig. 9. Zoom between 0 and 2 seconds.

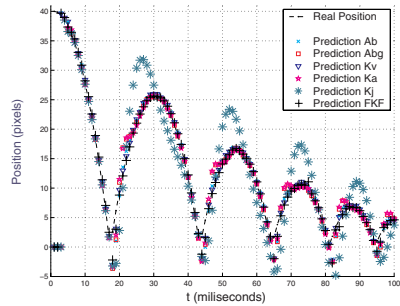


Fig. 10. Bounce of the ball on the ground Data.

### 6.3 Experimental Results

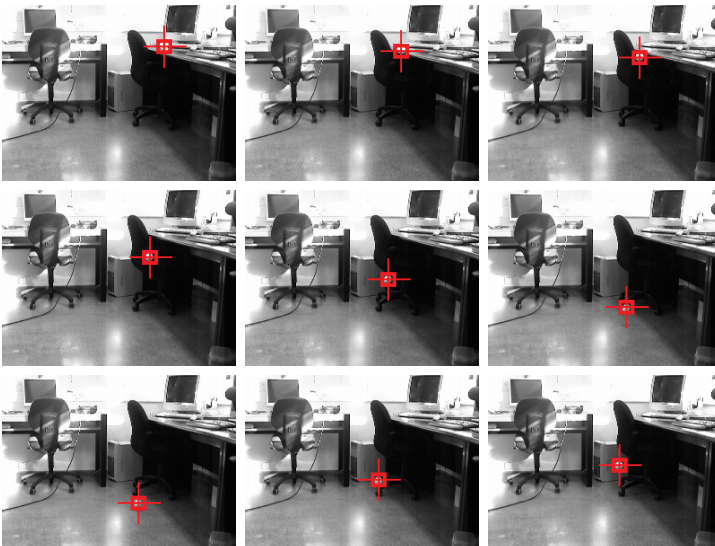
Experimental results are obtained for this work using the following setup: Pulnix GE series high speed camera (200 frames per second), Intel PRO/1000 PT Server Adapter card, 3.06GHz Intel processor PC computer, *Windows XP Professional* O.S. and *OpenCV* blob detection library.

For this configuration, the bounce of a ball on the ground is processed to obtain data shown in Fig. 10. Results of this experiment are presented in table 1. In this table, we can see the dispersion of several filters. The Fuzzy predictor dispersion is less than  $\alpha\beta$ ,  $\alpha\beta\gamma$ ,  $Kv$ ,  $Ka$  and  $Kj$  although the Fuzzy predictor is a combination of them. This table contains data from this particular experiment (the bounce of a ball on the ground). For this experiment, the position of the ball is introduced to the filters to prove the behaviour of them. The filter proposed (Fuzzy predictor) is the best analyzed.

**Table 1.** Numerical comparative for dispersion value of all filters implemented (bounce of a ball experiment).

<i>Init. pos.</i>	$\alpha\beta$	$\alpha\beta\gamma$	$Kv$	$Ka$	$Kj$	<i>Fuzzy predictor</i>
40	0.619	0.559	0.410	0.721	0.877	0.353
40(bis)	0.547	0.633	0.426	0.774	0.822	0.340
50	0.588	0.663	0.439	0.809	0.914	0.381
70	0.619	0.650	0.428	0.700	0.821	0.365
90	0.630	0.661	0.458	0.818	0.857	0.343
150	0.646	0.682	0.477	0.848	0.879	0.347

In Fig. 11 we can see some frames of the experiment 'bounce of a ball on the ground'. For each frame the center of gravity of the tennis ball is obtained.



**Fig. 11.** Bounce of the ball on the ground Frames.

## 7 Conclusions and Future Work

In section 6.1 (Fig. 5 and Fig. 6), we can see the quality of the new filter presented (Fuzzy predictor) which shows good behaviour for smooth and discontinuous motions. The object's position is estimated even when it is inside the image plane and when it is outside the image plane. Therefore, combine classic filters (KF) when inside and steady-state filters ( $\alpha\beta/\alpha\beta\gamma$ ) when outside.

We have compared our filter with  $\alpha\beta$ ,  $\alpha\beta\gamma$ ,  $Kv$ ,  $Ka$  and  $Kj$  in experiments of pure prediction. We have compared too, our filter with *Circular Prediction Algorithm* (CPA) in this work reproducing the same experiment as [5] for a direct comparison with the work done by Chroust and Vincze. The filter proposed works very well but not better than SKF for all conditions, therefore, the addition of a AKF action can improve the filter behaviour (future work).

The Fuzzy predictor is evaluated with a ramp-like and sinusoidal motions.  $\Delta x_p$  is reduced in all tests done and the overshoot is decreased significantly.

Results presented in this work are obtained for  $C(z) = K_P$ . Other controllers like PD, PID, ... will be implemented in future work.

## References

1. Kalman, R.: A new approach to linear filtering and prediction problems. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Computer Society (1960)
2. Corke, P.: Visual Control of Robots: High Performance Visual Visual Servoing. 1996 edn. Research Studies Press, Wiley, New York (1998)
3. Dickmanns, E., Graefe, V.: Dynamic monocular machine vision. In: Applications of dynamic monocular machine vision, Machine Vision and Applications (1988)
4. Wilson, W., Williams Hulls, C., Bell, G.: Relative end-effector control using cartesian position based visual servoing. In: IEEE Transactions on Robotics and Automation, IEEE Computer Society (1996)
5. Stefan, C., Markus, V.: Improvement of the prediction quality for visual servoing with a switching kalman filter. I. J. Robotic Res. **22** (2003) 905–922
6. Chroust, S., Zimmer, E., Vincze, M.: Pros and cons of control methods of visual servoing. In: In Proceedings of the 10th International Workshop on Robotics in the Alpe-Adria-Danube Region, IEEE Computer Society (2001)
7. Soatto, S., Frezza, R., Perona, P.: Motion estimation via dynamic vision. In: IEEE Transactions on Automatic Control, IEEE Computer Society (1997)
8. Duric, Z., Fayman, J., Rivlin, E.: Function from motion. In: Transactions on Pattern Analysis and Machine Intelligence, IEEE Computer Society (1996)
9. Angeles, J., Rojas, A., Lopez-Cajun, C.: Trajectory planning in robotics continuous-path applications. In: Journal of Robotics and Automation, IEEE Computer Society (1988)
10. Duric, Z., Rivlin, E., Rosenfeld, A.: Understanding the motions of tools and vehicles. In: Proceedings of the Sixth International Conference on Computer Vision, IEEE Computer Society (1998)
11. Duric, Z., Rivlin, E., Davis, L.: Egomotion analysis based on the frenet-serret motion model. In: Proceedings of the 4th International Conference on Computer Vision, IEEE Computer Society (1993)
12. Nomura, H., Natio, T.: Integrated visual servoing system to grasp industrial parts moving on conveyor by controlling 6dof arm. In: International Conference on Systems, Man, and Cybernetics, IEEE Computer Society (2000)

13. Li, X., Jilkov, V.: A survey of maneuvering target tracking: Dynamic models. In: Signal and Data Processing of Small Targets, The International Society for Optical Engineering (2000)
14. Mehrotra, K., Mahapatra, P.: A jerk model for tracking highly maneuvering targets. In: Transactions on Aerospace and Electronic Systems, IEEE Computer Society (1997)
15. Wang, L.: Course in Fuzzy Systems and Control Theory. Pearson US Imports & PHIPes. Pearson Higher Education (1997)
16. Tenne, D., Singh, T.: Circular prediction algorithms-hybrid filters. In: American Control Conference, IEEE Computer Society (2002)
17. Maybeck, P.: Stochastic Models, Estimation and Control. Academic Press, New York (1982)
18. Hutchinson, S., Hager, G., Corke, P.: Visual servoing: a tutorial. In: Transactions on Robotics and Automation, IEEE Computer Society (1996)
19. Markus, V., Gregory, D.: Robust Vision for Vision-Based Control of Motion. SPIE Press / IEEE Press, Bellingham, Washington (2000)
20. Vincze, M., Weiman, C.: On optimizing window size for visual servoing. In: International Conference on Robotics and Automation, IEEE Computer Society (1997)
21. Vincze, M.: Real-time vision, tracking and control dynamics of visual servoing. In: International Conference on Robotics and Automation, IEEE Computer Society (2000)
22. Sugeno, M.: Industrial applications of fuzzy control. Elsevier Science Publications Company (1985)
23. Passino, K., Yourkovich, S.: Fuzzy Control. Addison-Wesley, Ohio, USA (1988)

# Motion Control of an Omnidirectional Mobile Robot

Xiang Li and Andreas Zell

Wilhelm-Schickard-Institute, Department of Computer Architecture  
University of Tübingen, Sand 1, 72076 Tübingen, Germany  
{xiang.li, andreas.zell}@uni-tuebingen.de

**Abstract.** This paper focuses on the motion control problem of an omnidirectional mobile robot. A new control method based on the inverse input-output linearized kinematic model is proposed. As the actuator saturation and actuator dynamics have important impacts on the robot performance, this control law takes into account these two aspects and guarantees the stability of the closed-loop control system. Real-world experiments with an omnidirectional middle-size RoboCup robot verify the performance of this proposed control algorithm.

## 1 Introduction

Recently, omnidirectional wheeled robots have received more attention in mobile robots applications, because they have full mobility in the plane, which means that they can move at each instant in any direction without any reorientation [1]. Unlike nonholonomic robots, such as car-like robots, having to rotate before implementing any desired translation velocity, omnidirectional robots have higher maneuverability and are widely used in dynamic environments, for example, in the middle-size league of the annual RoboCup competition.

Most motion control methods of mobile robots are based on dynamic models [2–5] or kinematic models [6–8] of robots. A dynamic model directly describes the relationship between the forces exerted by the wheels and the robot movement, with the applied voltage of each wheel as the input and the robot movement in terms of linear and angular accelerations as the output. But the dynamic variations caused by the changes in the robot's inertia moment and perturbations from the mechanic components [9] make the controller design more complex. With the assumption that no slippage of wheels occurs, sensors have high accuracy and ground is planar enough, kinematic models are widely used in designing robots behaviors because of the simpler model structures. As the inputs of kinematic models are robot wheels velocities, and outputs are the robot linear and angular velocities, the actuator dynamics of the robot are assumed to be fast enough to be ignored, which means that the desired wheel velocities can be achieved immediately. However, the actuator dynamics limit and even degrade the robot performance in real situations.

Another important practical issue of robot control is actuator saturation. Because the commanding motor speeds of the robot wheels are bounded by the saturation limits, the actuator saturation can affect the robot performance, even destroy the stability of the controlled robot systems [10, 11].

This paper presents a motion control method for an omnidirectional robot, based on the inverse input-output linearization of the kinematic model. It takes into account not only the identified actuator dynamics but also the actuator saturation in designing a controller, and guarantees the stability of the closed-loop control system.

The remainder of this paper introduces the kinematic model of an omnidirectional middle-size Robocup robot in section 2; Path following and orientation tracking problems are solved based on the inverse input-output linearized kinematic model in section 3, where the actuator saturation is also analyzed; section 4 presents the identification of actuator dynamics and their influence on the control performance. Finally, the experiment results and conclusions are discussed in sections 5 and 6, respectively.

## 2 Robot Kinematic Model

The mobile robot used in our case is an omnidirectional robot, whose base is shown in Fig. 1. It has three Swedish wheels mounted symmetrically with 120 degrees from each other. Each wheel is driven by a DC motor and has a same distance  $L$  from its center to the robot's center of mass  $R$ .

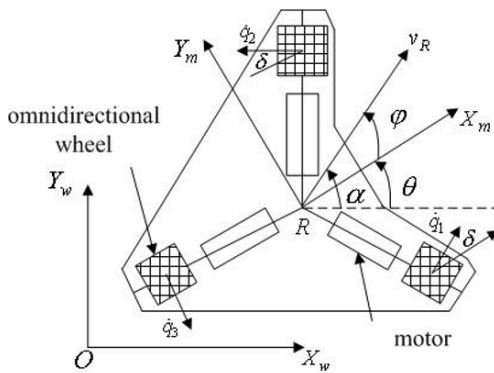


Fig. 1. Kinematics diagram of the base of an omnidirectional robot.

Besides the fixed world coordinate system  $[X_w, Y_w]$ , a mobile robot fixed frame  $[X_m, Y_m]$  is defined, which is parallel to the floor and whose origin locates at  $R$ .  $\theta$  denotes the robot orientation, which is the direction angle of the axis  $X_m$  in the world coordinate system.  $\alpha$  and  $\varphi$  denote the direction of the robot translation velocity  $v_R$  observed in the world and robot coordinate system, respectively. The kinematic model with respect to the robot coordinate system is given by :

$$\mathbf{v} = \begin{bmatrix} \sqrt{3}/3 & -\sqrt{3}/3 & 0 \\ 1/3 & 1/3 & -2/3 \\ 1/(3L) & 1/(3L) & 1/(3L) \end{bmatrix} \dot{\mathbf{q}}, \tag{1}$$

where  $\mathbf{v} = [\dot{x}_R^m \ \dot{y}_R^m \ \omega]^T$  is the vector of robot velocities observed in the robot coordinate system;  $\dot{x}_R^m$  and  $\dot{y}_R^m$  are the robot translation velocities;  $\omega$  is the robot rotation velocity.

$\dot{\mathbf{q}}$  is the vector of wheel velocities  $[\dot{q}_1 \ \dot{q}_2 \ \dot{q}_3]^T$ , and  $\dot{q}_i (i = 1, 2, 3)$  is the  $i$ -th wheel's velocity, which is equal to the wheel's radius multiplied by the wheel's angular velocity.

Introducing the transformation matrix from the robot coordinate system to the world coordinate system as

$${}^w R_m = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \tag{2}$$

the kinematic model with respect to the world coordinate system is deduced as:

$$\dot{\mathbf{x}} = \begin{bmatrix} \frac{2}{3} \cos(\theta + \delta) & -\frac{2}{3} \cos(\theta - \delta) & \frac{2}{3} \sin \theta \\ \frac{2}{3} \sin(\theta + \delta) & -\frac{2}{3} \sin(\theta - \delta) & -\frac{2}{3} \cos \theta \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \dot{\mathbf{q}}, \tag{3}$$

where  $\dot{\mathbf{x}} = [\dot{x}_R \ \dot{y}_R \ \dot{\theta}]^T$  is the vector of robot velocities with respect to the world coordinate system;  $\dot{x}_R$  and  $\dot{y}_R$  are the robot translation velocities;  $\dot{\theta}$  is the robot rotation velocity;  $\delta$  refers to the wheel's orientation in the robot coordinate system and is equal to 30 degrees.

It is important to notice that the transformation matrix in the kinematic models is full rank, which denotes that the translation and rotation of the robot are decoupled, and guarantees the separate control of these two movements.

For the high level control laws without considering the wheel velocities, the kinematic model

$$\dot{\mathbf{x}} = G\mathbf{v} \tag{4}$$

is used in our control method, where the transformation matrix  $G$  is equal to  $[\begin{smallmatrix} {}^w R_m & 0 \\ 0 & 1 \end{smallmatrix}]$ . Because  $G$  is full rank, the characteristics of decoupled movement is also kept.

### 3 Inverse Input-Output Linearization based Control

The trigonometric functions of angle  $\theta$  in the transformation matrix  $G$  determine the nonlinearities of the kinematic model (4). Since the matrix  $G$  is full rank, this nonlinear model can be exactly linearized by introducing a simple compensator  $C = G^{-1}$ . The linearized system becomes  $\dot{\mathbf{x}} = \mathbf{u}$  with a new input vector  $\mathbf{u} = [u_1 \ u_2 \ u_3]^T$ .

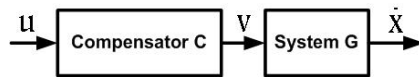


Fig. 2. Linearized system by the compensator  $C$ .

This linear system shown in Fig. 2 is completely decoupled and allows the controlling of the robot's translation and rotation in a separate way. When a controller  $K$  is designed based on this simple linear system, the controller of the original system is generated as  $CK$ . The overall control loop, which consists of the nonlinear system, the compensator and the controller, is shown in Fig. 3,

where  $\mathbf{x}$  denotes the robot state vector  $[x_R \ y_R \ \theta]^T$  and  $\mathbf{x}_d$  is the desired state vector;  $x_R$  and  $y_R$  are robot position observed in the world coordinate system.

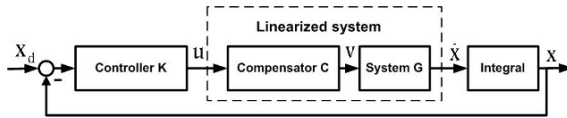


Fig. 3. Closed-loop control system.

Based on this input-output linearized system, path following and orientation tracking problems are analyzed with respect to the robot translation and rotation control in the following subsections. The influence of actuator saturation is also accounted to keep the decoupling between the translation and rotation movements.

### 3.1 Path Following Control

As one high-level control problem, path following is chosen in our case to deal with the robot translation control. The path following problem is illustrated in Fig. 4.  $P$  denotes the given path. Point  $Q$  is the orthogonal project of  $R$  on the path  $P$ . The path coordinate system  $x_t Q x_n$  moves along the path  $P$  and the coordinate axes  $x_t$  and  $x_n$  direct the tangent and normal directions at point  $Q$ , respectively.  $\theta_P$  is the path tangent direction at point  $Q$ .

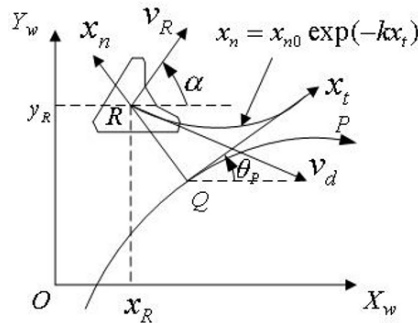


Fig. 4. Illustration of the path following problem.

Based on the above definitions, the path following problem is to find proper control values of the robot translation velocity  $v_R$  and angular velocity  $\dot{\alpha}$  such that the deviation distance  $x_n$  and angular error  $\tilde{\theta}_R = \alpha - \theta_P$  tend to zero.

To solve this problem, a Lyapunov candidate function

$$V = \frac{1}{2} K_d x_n^2 + \frac{1}{2} K_\theta \tilde{\theta}_R^2 \tag{5}$$

can be considered, where  $K_d$  and  $K_\theta$  are positive constants. The time derivation of  $V$  results in

$$\dot{V} = K_d x_n \dot{x}_n + K_\theta \tilde{\theta}_R \dot{\tilde{\theta}}_R. \tag{6}$$

Mojaev [12] presents a simple control law based on the deviation  $x_n$ , where  $R$  is controlled to move along an exponential curve and to converge to the axis  $x_t$ . The



exponential curve is expressed as

$$x_n = x_{n_0} \exp(-kx_t), \quad (7)$$

where  $x_{n_0}$  is the initial deviation and the positive constant  $k$  determines the convergence speed of the deviation. Differentiating (7) with respect to  $x_t$ , we get the tangent direction of the exponential curve as

$$\tilde{\theta}_R = \arctan\left(\frac{dx_n}{dx_t}\right) = \arctan(-kx_n). \quad (8)$$

Therefore, for a non-zero constant desired velocity  $v_d$ , the translation velocity of robot in the coordinate system  $x_t Q x_n$  results in

$$\dot{x}_n = v_d \sin \tilde{\theta}_R, \quad (9)$$

$$\dot{x}_t = v_d \cos \tilde{\theta}_R. \quad (10)$$

Substituting the time derivative of  $\tilde{\theta}_R$  into (6), we get

$$\dot{V} = K_d x_n \dot{x}_n + k K_\theta \arctan(-kx_n) \frac{-\dot{x}_n}{1 + (kx_n)^2} < 0, \quad (11)$$

because  $x_n \dot{x}_n = x_n v_d \sin(\arctan(-kx_n)) < 0$  and  $\dot{x}_n \arctan(kx_n) < 0$ . This solution of  $\dot{V}$  guarantees the global stability of the equilibrium at  $x_n = 0, \tilde{\theta}_R = 0$ , which means this control law solves the path following problem.

Transforming the robot velocity into the world coordinate system, we get the control values of the linearized system as

$$u_1 = v_d \cos \alpha, \quad (12)$$

$$u_2 = v_d \sin \alpha, \quad (13)$$

where  $\alpha = \tilde{\theta}_R + \theta_P$ .

The input of controller (12) and (13) is the deviation distance between point  $R$  and the given path, which normally can be directly obtained by the sensors on the robot. Moreover, the deviation converges smoothly to zero with the speed controlled by parameter  $k$ , which can be chosen according to the performance requirement.

### 3.2 Orientation Tracking

Unlike a car-like wheeled robot, the orientation of an omnidirectional robot can be different from the direction of the robot translation velocity by any angle  $\varphi$ . This relationship is denoted as  $\alpha = \theta + \varphi$ . That means the robot orientation can track any angle when the robot is following a given path. Based on the linearized model, the orientation tracking task is to find a suitable  $u_3$ , which is equal to the robot rotation velocity  $\omega$ , such that

$$\lim_{t \rightarrow \infty} (\theta_d(t) - \theta(t)) = 0, \quad (14)$$

where  $\theta_d(t)$  is the desired orientation.

As the system between input variable  $u_3$  and output variable  $\theta$  is an integrator, a commonly used PD controller can be designed to fulfill the orientation tracking task.

### 3.3 Actuator Saturation

Based on the inverse input-output linearization, the translation and rotation of an omnidirectional robot can be easily achieved in a separate way. This linearization is with respect to the input-output relationship, which requires the internal parts having sufficient capability to achieve the desired inputs. However, the power of the robot motors is bounded and the actuators will saturate when the commanding velocities are too large. The presence of actuator saturation can influence the decoupling between robot translation velocity and rotation velocity, such that the system performance and stability is severely impacted. Therefore, it is necessary to deal with the actuator saturation in the controller design.

For our omnidirectional robot, the maximal velocity of each wheel is limited by  $\dot{q}_m$ , namely  $|\dot{q}_i| \leq \dot{q}_m$ . Substituting the above control values from equations (12) (13) and  $u_3$  into the inverse kinematic models (2) and (1), the wheel velocities are computed as:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} v_d \cos(\alpha - \theta - \delta) + Lu_3 \\ -v_d \cos(\alpha - \theta + \delta) + Lu_3 \\ v_d \sin(\theta - \alpha) + Lu_3 \end{bmatrix}, \quad (15)$$

To achieve orientation tracking based on the above path following control, the desired translation velocity's magnitude  $V_d$  is assumed to be not bigger than  $\dot{q}_m$ . Substituting  $\dot{q}_m$  into (15),

$$|v_d \cos(\alpha - \theta - \delta) + Lu_3| \leq \dot{q}_m \quad (16)$$

$$|-v_d \cos(\alpha - \theta + \delta) + Lu_3| \leq \dot{q}_m \quad (17)$$

$$|v_d \sin(\theta - \alpha) + Lu_3| \leq \dot{q}_m, \quad (18)$$

the lower and upper boundary of  $u_3$  with respect to each wheel ( $l_{b_i}$  and  $u_{b_i}$ ,  $i = 1, 2, 3$ ) can be calculated as follows,

$$l_{b_1} = -\dot{q}_m - v_d \cos(\alpha - \theta - \delta) \leq Lu_3 \leq \dot{q}_m - v_d \cos(\alpha - \theta - \delta) = u_{b_1} \quad (19)$$

$$l_{b_2} = -\dot{q}_m + v_d \cos(\alpha - \theta + \delta) \leq Lu_3 \leq \dot{q}_m + v_d \cos(\alpha - \theta + \delta) = u_{b_2} \quad (20)$$

$$l_{b_3} = -\dot{q}_m - v_d \sin(\theta - \alpha) \leq Lu_3 \leq \dot{q}_m - v_d \sin(\theta - \alpha) = u_{b_3}. \quad (21)$$

Then the dynamic boundary values of  $u_3$  are computed as

$$\begin{aligned} l_b &= \max(l_{b_1}, l_{b_2}, l_{b_3})/L \\ u_b &= \min(u_{b_1}, u_{b_2}, u_{b_3})/L, \end{aligned} \quad (22)$$

where  $l_b$  and  $u_b$  are the low and up boundary.

Considering the saturation function

$$x_2 = \begin{cases} u_b, & \text{if } x_1 > u_b \\ x_1, & \text{if } l_b \leq x_1 \leq u_b \\ l_b, & \text{if } x_1 < l_b, \end{cases} \quad (23)$$

and its gain characteristics illustrated in Fig. 5, we can take the saturation function as a dynamic gain block  $k_a$ , which has maximum value one and converges to zero when

the input saturates. Then the closed-loop system of controlling the robot orientation is as shown in Fig. 6, in which a PD controller is used to control the robot orientation converging to the ideal  $\theta_d$ ,

$$\omega = k_1(e_\theta + k_2\dot{e}_\theta), \tag{24}$$

where  $e_\theta = \theta_d - \theta$ ,  $k_1$  and  $k_2$  are the proportional and derivative gains, respectively. It can be obtained that the closed-loop has only one pole  $\frac{-k_a k_1}{1+k_a k_1 k_2}$  and one zero  $-1/k_2$ . Therefore, when  $k_2$  and  $k_1$  are positive, the stability of the closed-loop system can be guaranteed whenever  $k_a$  decreases.

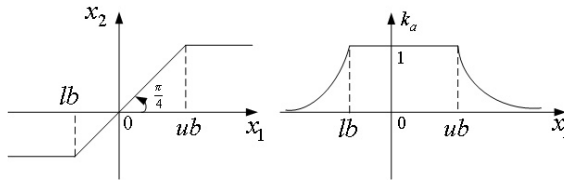


Fig. 5. Saturation function and its gain characteristics.

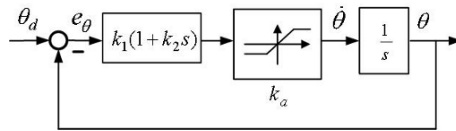


Fig. 6. Closed-loop of robot orientation control.

## 4 Actuator Dynamics

The results in the last section are only practical when we assume that the low level actuator dynamics is faster than the kinematics, or the delay of actuator dynamics can be ignored. Therefore, it is necessary to analyze the actuator dynamics and take it into account when designing a controller. In the following subsections, the actuator dynamics is identified based on the observed input-output data, and its influence on the robot motion control is presented.

### 4.1 Actuator Dynamics Identification

The system identification problem is to estimate a model based on the observed input-output data such that a performance criterion is minimized. Because the full rank transformation matrix in the low level dynamics model (1) denotes the outputs  $\dot{x}_R^m$ ,  $\dot{y}_R^m$  and  $\omega$  are not relevant, we identify the actuator models for these three values. The inputs of the actuator models are required velocity values ( $\dot{x}_{R_c}^m$ ,  $\dot{y}_{R_c}^m$  and  $\omega_c$ ), and the outputs are

corresponding measured values. As one commonly used parametric model, ARMAX is chosen as the identified model, which has the following structure

$$A(z)y(t) = B(z)u(t - n_k) + C(z)e(t), \tag{25}$$

$$A(z) = 1 + a_1z^{-1} + \dots + a_{n_a}z^{-n_a}, \tag{26}$$

$$B(z) = 1 + b_1z^{-1} + \dots + b_{n_b}z^{-n_b+1}, \tag{27}$$

$$C(z) = 1 + c_1z^{-1} + \dots + c_{n_c}z^{-n_c}. \tag{28}$$

$n_k$  denotes the delay from input  $u(t)$  to output  $y(t)$ .  $e(t)$  is white noise.  $z$  is the shift operator resulting in  $q^{-1}u(t) = u(t - 1)$ .  $n_a$ ,  $n_b$  and  $n_c$  are the orders of polynomials  $A(z)$ ,  $B(z)$  and  $C(z)$ , respectively. To choose the optimal parameters of this model, we use the prediction error method, which is to find the optimal  $n_k$  and parameters of  $A(z)$ ,  $B(z)$  and  $C(z)$  such that the prediction error  $E$  is minimized, namely

$$[A(z), B(z), C(z), n_k]_{opt} = \underset{t=1}{\operatorname{argmin}} \sum^N E^2 \tag{29}$$

$$E = y_o(t) - A^{-1}(z)(B(z)u(t - n_k) + C(z)e(t)), \tag{30}$$

where  $y_o(t)$  denotes the measured output data.

The system identification toolbox of Matlab has been used to identify the actuator dynamics model. Figures 7, 8 and 9 show the optimal parameters and the comparison between models outputs and measured outputs with respect to the actual inputs.

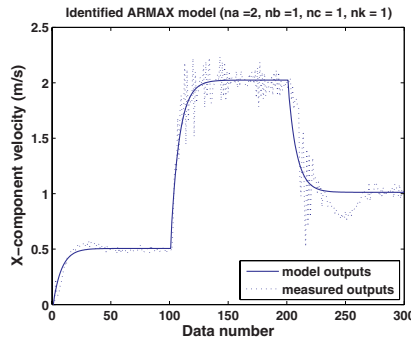


Fig. 7. Identified model for  $\dot{x}_R^m$ .

To coincide with the robot's continuous model, the identified models are transformed from discrete ones into continuous ones using the 'zoh' (zero-order hold) method,

$$\dot{x}_R^m = \frac{8.7948(s + 58.47)}{(s + 73.66)(s + 6.897)} \dot{y}_{R_c}^m, \tag{31}$$

$$\dot{y}_R^m = \frac{2.4525(s + 48.83)(s + 6.185)}{(s + 28.45)(s^2 + 6.837s + 25.97)} \dot{y}_{R_c}^m, \tag{32}$$

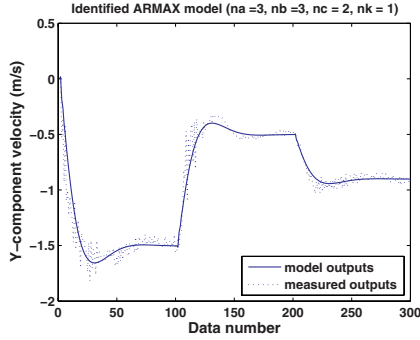


Fig. 8. Identified model for  $\dot{y}_R^m$ .

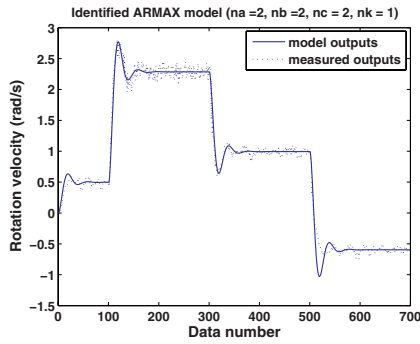


Fig. 9. Identified model for  $\omega$ .

$$\omega = \frac{1.667(s + 45.37)}{(s^2 + 6.759s + 76.11)} \omega_c. \tag{33}$$

### 4.2 Actuator Influence

With consideration of the actuator, the whole structure of the control system is shown in Fig. 10, where  $\mathbf{v}_c = [\dot{x}_{Rc}^m \ \dot{y}_{Rc}^m \ \omega_c]$  is the commanding robot velocity vector with

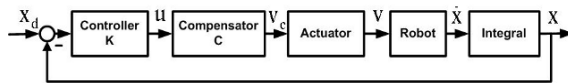


Fig. 10. Closed-loop control system including actuator dynamics.

respect to the robot coordinate system. Because the poles of the actuators dynamics (31) and (32) have negative real parts, these two systems are stable. That means there exists a finite short time  $t^*$ , after which the real velocities  $\dot{x}_R^m$  and  $\dot{y}_R^m$  can converge to the desired ones  $\dot{x}_{Rc}^m$  and  $\dot{y}_{Rc}^m$ , and the inputs  $u_1$  and  $u_2$  begin to take effect. Therefore, the

above path following law can also guarantee the robot approach to the reference path, although during  $t^*$  the deviation distance  $x_n$  and angular error  $\tilde{\theta}_R$  may increase.

In the orientation tracking control, as the dynamic system (33) adds another two poles to the closed-loop system, shown in Fig. 11, the controller parameters decided in the above section may result the system losing the stability.

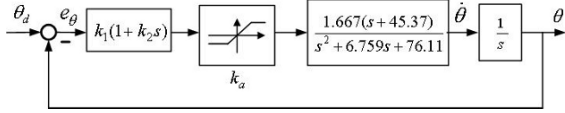


Fig. 11. Closed-loop of robot orientation control including actuator dynamics.

By setting the positions of poles and zeros of the closed-loop system with the locus technique, we obtain that the conditions  $k_1 > 0$  and  $k_2 > 0.0515$  can guarantee the closed-loop system’s stability, even when the actuators saturate. Fig. 12 shows the root locus of an open-loop system in the critical situation with  $k_2 = 0.0515$ , where all the poles of the closed-loop system locate in the left-half plane whatever positive value  $k_a k_1$  is. Otherwise, when  $k_2$  is less than 0.0515, the root locus may cross the imaginary axis, and the poles of closes-loop system may move to the right-half plane when  $k_a$  goes to zero.

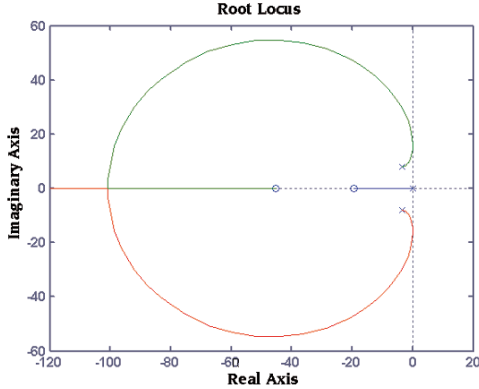
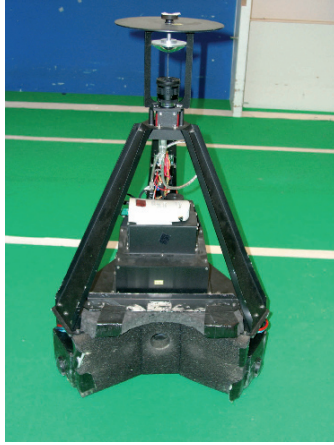


Fig. 12. Root locus of the open-loop model.

### 5 Experiment

The control algorithm discussed above has been tested in our robot laboratory having a half-field of the RoboCup middle size league. The omnidirectional robot is shown in Fig. 13.

An AVT Marlin F-046C color camera with a resolution of  $780 \times 580$  is assembled pointing up towards a hyperbolic mirror, which is mounted on the top of the omnidirectional robot, such that a complete surrounding map of the robot can be captured. A



**Fig. 13.** The real omnidirectional robot.

self-localization algorithm described in [13] based on the 50 Hz output signal of the camera gets the robot's position in the play field in real time. The wheels are driven by three 60W Maxon DC motors and the maximum wheel velocity is  $1.9m/s$ . Three wheel encoders measure the real wheel velocities, which are steered by three PID controllers.

An eight-shaped path is adopted as the reference path, while its geometrical symmetry and sharp changes in curvature make the test challenging. With a scale variable  $s$ , the chosen eight-shaped path is calculated as

$$\begin{aligned} x_r &= 1.8\sin(2s) \\ y_r &= 1.2\sin(s), \end{aligned} \tag{34}$$

The robot was controlled to follow the eight-shaped path with a constant translation velocity  $v_d = 1m/s$ , and the parameters of our control algorithm were chosen as  $k = 2.5$ ,  $k_1 = 4.15$ ,  $k_2 = 3$ . The first experiment selected the path tangent direction  $\theta_p$  as the desired robot orientation. Figures 14, 16, 18 and 20 show us that the proposed control method steers the robot center  $R$  converging to the given path and the robot orientation tracking the desired ones with acceptable errors, where the actuator saturation did not appear. In order to check the influence of the actuator saturation, the second experiment selected the desired robot orientation as

$$\theta_d = \theta_p + 0.9c_P v_d^2, \tag{35}$$

where  $c_P$  is the path curvature at point  $Q$ . The results illustrated in figures 15, 17, 19 and 21 show us that the robot center  $R$  converges to the given path, even though the wheels velocities come in the saturation when the path turns sharply.

## 6 Conclusions

In this paper a new motion control method for an omnidirectional robot is presented. This approach is based on the inverse input-output linearized robot kinematic model,

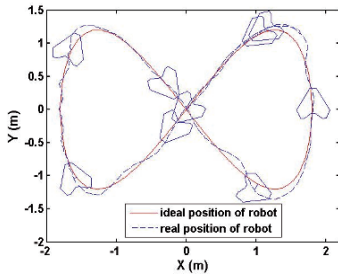


Fig. 14. Reference path and robot path.

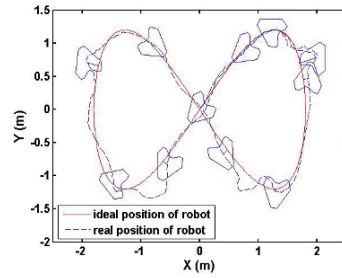


Fig. 15. Reference path and robot path.

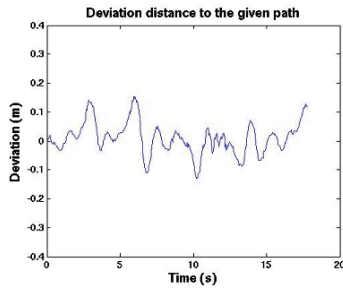


Fig. 16. Distance error.

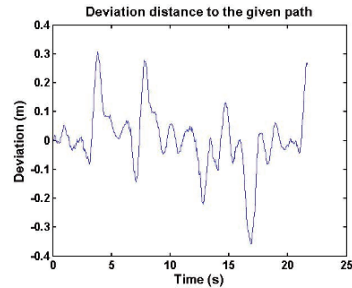


Fig. 17. Distance error.

which completely decoupled the robot translation and rotation. The robot translation is steered to follow a reference path, and the robot rotation is controlled to track the desired orientation. Because the actuator dynamics and saturation can greatly affect the robot performance, they are taken into account when designing the controller. With the Lyapunov stability theory, the global stability of the path following control law has been proven. The locus technique is used to analyze and choose the suitable parameters of the PD controller, such that the robot orientation can converge to the desired one even when the wheels velocities saturate.

In real-world experiments, the robot was controlled to follow an eight-shaped curve with a constant translation velocity of  $1\text{ m/s}$ , and to track sharp changing orientations. The results show the effectiveness of the proposed control method in the case of both actuator saturation and non-saturation.

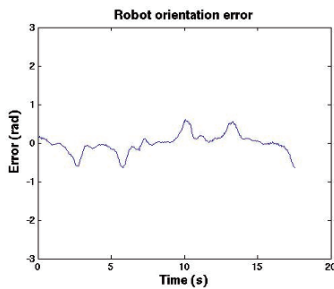


Fig. 18. Orientation error.

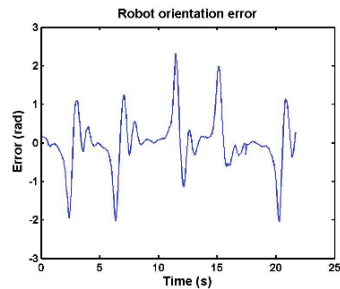


Fig. 19. Orientation error.



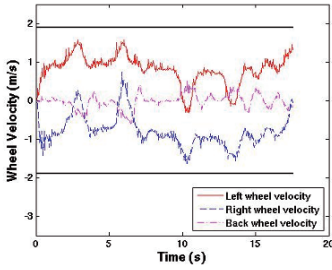


Fig. 20. Real wheel velocities.

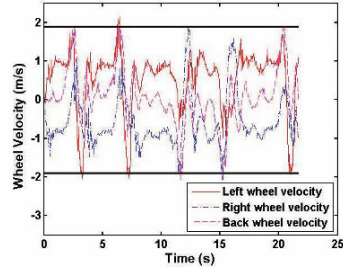


Fig. 21. Real wheel velocities.

## References

1. Campion, G., Bastin, G., D'Andréa-Novel, B.: Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. In: IEEE Transactions on Robotics and Automation. Volume 12. (1996)
2. Watanabe, K.: Control of an omnidirectional mobile robot. In: KES'98, 2th International Conference on Knowledge-Based Intelligent Electronic Systems. (1998)
3. Liu, Y., Wu, X., Zhu, J.J., Lew, J.: Omni-directional mobile robot controller design by trajectory linearization. In: ACC'03, Proceeding of the 2003 American Control Conference. (2003)
4. Purwin, O., Andrea, R.D.: Trajectory generation and control for four wheeled omnidirectional vehicles. In: Robotics and Autonomous Systems. Volume 54(1). (2006)
5. Tsai, C.C., Huang, H.C., Wang, T.S., Chen, C.M.: System design, trajectory planning and control of an omnidirectional mobile robot. In: 2006 CACS Automatic Control Conference. (2006)
6. Muir, P.F., Neuman, C.P.: Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot. In: Autonomous Robot Vehicles, Springer-Verlag (1990)
7. Terashima, K., Miyoshi, T., Urbano, J., Kitagawa, H.: Frequency shape control of omnidirectional wheelchair to increase user's comfort. In: ICRA'04, Proceedings of the 2004 IEEE International Conference on Robotics and Automation. (2004)
8. Rojas, R., Förster, A.G.: Holonomic Control of a Robot with an Omni-directional Drive. BöttcherIT Verlag, Bremen (2006)
9. Scolari Conceição, A., j. Costa, P., Moreira, A.: Control and model identification of a mobile robot's motors based in least squares and instrumental variable methods. In: MMAR'05, 11st International Conference on Metgids abd Models in Automation and Robotics. (2005)
10. Indiveri, G., Paulus, J., Plöger, P.G.: Motion control of swedish wheeled mobile robots in the presence of actuator saturation. In: 10th annual RoboCup International Symposium. (2006)
11. Scolari Conceição, A., Moreira, A., j. Costa, P.: Trajectory tracking for omni-directional mobile robots based on restrictions of the motor's velocities. In: SYROCO'06, 8th International IFAC Symposium on Robot Control. (2006)
12. Mojaev, A., Zell, A.: Tracking control and adaptive local navigation for nonholonomic mobile robot. In: Proceedings of the IAS-8 conference. (2004)
13. Heinemann, P., Rueckstiess, T., Zell, A.: Fast and accurate environment modelling using omnidirectional vision. In: Dynamic Perception, Infix (2004)

# A Strategy for Exploration with a Multi-robot System

Jonathan A. Rogge and Dirk Aeyels

SYSTeMS Research Group, Ghent University  
Technologiepark Zwijnaarde 914, 9000 Gent, Belgium  
jonathan.rogge, dirk.aeyels@ugent.be

**Abstract.** The present paper develops a novel strategy for the exploration of an unknown environment with a multi-robot system. Communication between the robots is restricted to line-of-sight and to a maximum inter-robot distance. The algorithm we propose is related to methods used for complete coverage of an area, where all free space is physically covered. In the present paper it is required that the entire free space is covered by the *sensors* of the robots, enabling us to scan more space in less time, compared to complete coverage algorithms. The area to be scanned contains disjoint convex obstacles of unknown size and shape. The geometry of the robot group has a zigzag shape, which is stretched or compressed to adapt to the environment. The robot group is allowed to split and rejoin when passing obstacles. A direct application of the algorithm is mine field clearance.

**Keywords.** Multi-robot systems, coverage, exploration, demining.

## 1 Introduction

The research domain of multi-agent robot systems can be divided into subdomains according to the task given to the robot group [1]. At present well-studied subdomains are motion-planning (also called path-planning), formation-forming, region-sweeping, and combinations of the foregoing. The problem considered in the present paper belongs to the discipline comprising region-sweeping. In this discipline two different robot tasks are usually considered.

In the first task a group of robots receives the order to *explore/map* an unknown region. The goal is to obtain a detailed topography of the desired area. A typical approach to tackle the above problem with multiple robots assumes unlimited communication [2]: since exploration algorithms are already devised for a single robot it seems straightforward to divide the area to be explored into disjunct regions, each of which is assigned to a single robot. The robots communicate to each other the area they have explored so that no part of the free space will be explored twice unnecessarily. At no point during the task are the robots trying to form a fixed formation. Each robot explores a different part of the unknown region and sends its findings to a central device which combines the data received from the robots into one global map of the area.

Closely related to the exploring/mapping task is the second task, called *complete coverage*, where the robots have to move over all of the free surface in configuration space. Typical applications are mine field clearance, lawn mowing and snow cleaning. The coverage problem has been addressed in the literature both in a deterministic and

a probabilistic setting. In the probabilistic approach the robots are considered as if they were fluid or gas molecules satisfying the appropriate physical laws of motion [3], [4]. Just as a gas by diffusion fills an entire space, the robots will cover all free space when time tends to infinity. In the remainder of the paper we focus on the deterministic setting. In this setting the robot group typically forms (partial) formations to solve the task. Reference [5] gives a short overview of existing techniques for multi-robot coverage problems. Different approaches to the coverage problem are found in [6], [7], [8], [9] [10] and [11].

The problem statement of the present paper does not differ that much from the common exploration/mapping task and the complete coverage problem, but is rather a combination of both. It is required that all of the free space is *sensed* by the robots, but not necessarily physically covered. However, unlike the common exploration case, the sensing of the area does not have as goal to map the topography of the free space and the location of the obstacles in it. Our aim is to locate several unknown targets within the free space. Moreover, similar to the complete coverage setting we demand a 100% certainty that *all free space* has been covered by the sensors at the end of the exploration procedure, implying that all targets have been found. Since the robots no longer have to cover all free space physically, the novel algorithm will yield a time gain compared to complete coverage strategies. It is assumed that the space to be explored does not have a maze-like structure with many narrow corridors, but is an open space containing only convex obstacles sparsely spread throughout. Our algorithm is presented in Section 2 of the paper. A short comparison between the sensor coverage algorithm presented here and the physical coverage algorithm of [10] is given in Section 3.

A specific application we have in mind is mine field clearance using chemical vapor microsensors [12]. Once a landmine is deployed, the environment near the mine becomes contaminated with explosives derived from the charge contained in the mine. The vapor microsensors are able to detect the chemical vapor signature of the explosives emanating from the landmines. This implies that complete coverage algorithms may be too restrictive with respect to the demining problem. Performing the algorithm of the present paper, with the weaker requirement of sensor coverage, will result in a gain of time.

The algorithm can be used in problems where a robot group has to traverse a terrain containing sparsely spread obstacles. There is a natural trade-off between coherence of the formation and avoidance of the obstacles. The robot group is allowed to split in order to pass the obstacles, resulting in faster progress of the group across the terrain. The algorithm ensures that once the obstacle is passed, the robots regroup.

## 2 An Algorithm for Complete Sensor Coverage

### 2.1 Setting

Consider a population of  $N$  identical robots, with  $N$  even. Each robot is equipped with two types of sensors. One type serves as a means to detect the goal targets to be found in the assigned area, e.g. landmines; the other type is used to detect and locate other

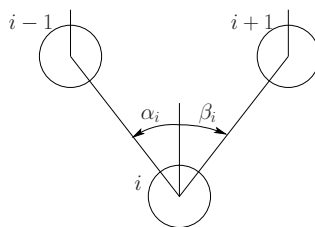
robots and obstacles in the neighborhood of the robot<sup>1</sup>. Both sensors have a maximum detection range  $s_t$  and  $s_r$  respectively. It is assumed that targets which come within the radius of the corresponding sensor area  $s_t$  or  $s_r$  of the robot are always detected, and that if they are located farther away than the distance  $s_t, s_r$  they are never detected. The robot configuration allows limited communication. First, this is expressed by the maximum detection range  $s_r$  as described above. Second, line-of-sight communication is assumed: two robots can only sense each other if they are sufficiently close to each other and if there is no obstacle located on the straight line connecting both robots.

Two robots are called *connected* to each other when they sense each other. Every robot is assigned an index number. The initial state of the robot configuration is such that robot  $i$  is connected to robots  $i - 1$  and  $i + 1, \forall i \in \{2, \dots, N - 1\}$ . (Robot 1 is only connected to robot 2 and robot  $N$  is only connected to robot  $N - 1$ .) Furthermore, each robot keeps a constant distance  $d < s_r$  with its neighbors and observes them at preferred angles with respect to its forward direction. With notation from Figure 1 these angles are defined as follows. For robots with indices  $i < \frac{N}{2}$ , the angles are

$$\alpha_i = \begin{cases} \pi/6, & i \text{ even,} \\ 5\pi/6, & i \text{ odd,} \end{cases} \tag{1}$$

$$\beta_i = \begin{cases} -\pi/6, & i \text{ even,} \\ -5\pi/6, & i \text{ odd} \setminus \{N/2 + 1\}. \end{cases}$$

To obtain the angles of the remaining robots, with indices  $i \geq \frac{N}{2}$ , simply replace  $i$  by  $i + \frac{N}{2}$  in the formulas above and define  $\beta_{\frac{N}{2}} := -\pi/2$  and  $\alpha_{\frac{N}{2}+1} := \pi/2$ . Each robot is equipped with a compass. Together with the above defined angles, the forward direction of each robot (the same for all robots) is imposed at the initialization of the algorithm. The above conditions imply a robot formation with zigzag shape, as depicted in Figure 2 for  $N = 6$ . The dashed circles have radius  $s_t$  and signify the sensed area

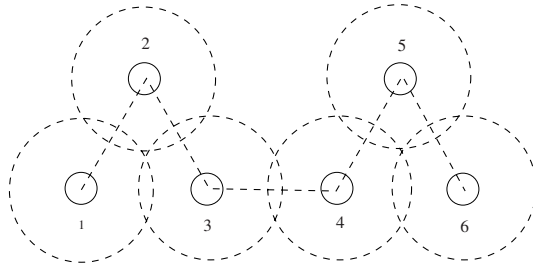


**Fig. 1.** Defining the angles of the preferred robot configuration.

for goal targets of each robot. It is assumed that

$$\frac{s_r}{2} < s_t < s_r. \tag{2}$$

<sup>1</sup> In practice the latter type consists of two distinct minimally interfering IR-sensors: one sensing obstacles and the other sensing robots. Since this is not relevant for the theoretical description of the algorithm, these sensors are considered as if they are one and the same.



**Fig. 2.** Overlapping sensor areas in a possible robot configuration.

The lower bound on  $s_t$  in (2) ensures that the areas sensed for goal targets of neighboring robots partially overlap, as illustrated by Figure 2.

## 2.2 The Scanning Algorithm

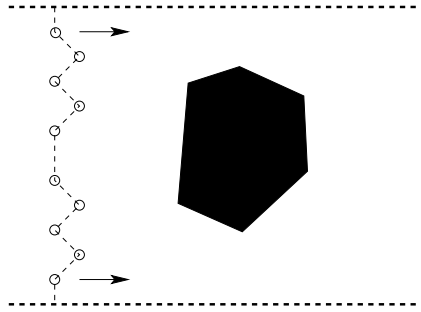
Assume for simplicity that the area to be explored is a rectangular subset  $S$  of  $\mathbb{R}^2$ . All obstacles contained in  $S$  are assumed disjoint and convex. Divide the set  $S$  into parallel (scanning) strips of width  $(\frac{N}{2} - 1)d$ . This choice of the value of the width will be motivated later, in Section 2.4. Furthermore assume that  $N$  and  $d$  are such that all obstacles in  $S$  have a diameter smaller than the width of a scanning strip. Fix the maximum allowed diameter at  $(\frac{N}{2} - 3)d$ . The main idea of the algorithm is to let the group of robots sweep the area  $S$  strip after strip in a zigzag-like pattern. Clearly, when there is a sufficient number of robots available the set  $S$  can be regarded as one big strip, simplifying the algorithm since no transitions between consequent strips have to be performed.

In a first case we consider a strip where no objects are located on the boundary (see Figure 3). Robots 1 and  $N$  are allocated the task to follow the boundaries of the strip at a constant distance at the constant velocity  $v$ . They can be considered leaders of the robot group. These two leader robots do not try to stay in the preferred formation, i.e. the condition on the corresponding angles  $\alpha_N, \beta_1$  is removed, and they do not maintain a fixed interdistance  $d$  with their neighbors. The remaining robots, however, still maintain the preferred formation. When no obstacles are present in the strip, the robots scan the strip for goal targets in the above defined preferred (rigid) formation moving at a velocity  $v$ .

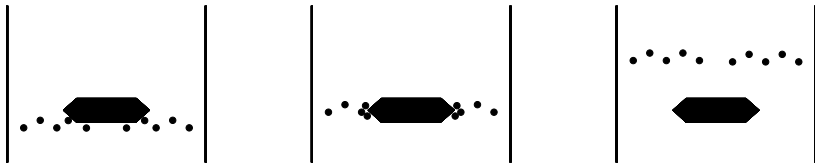
When an obstacle is encountered the algorithm aims to guide the robot group past it in a time-optimal way. The leader robots start driving at a preset velocity  $v_0 < v$  (see Section 3); the group is split into two subgroups in order to move around the obstacle. The subgroups rejoin after passing the obstacle to resume the preferred formation structure.

Consider the situation where robot  $m$  encounters an object on its path such that it cannot stay in the preferred formation any longer. More precisely, the sensors of robot  $m$  measure

- an interdistance between the obstacle and the robot smaller than a preset distance  $d_o < s_r$ ,



**Fig. 3.** A depiction of the algorithm. The arrows indicate the constant velocity of both leader robots. The dashed lines represent the strip boundaries.



**Fig. 4.** A group of 10 robots passing an obstacle.

- the position of the obstacle at an angle with its forward direction inside the interval  $(-\gamma, \gamma)$ , with  $\gamma$  a fixed value inside the interval  $(0, \frac{\pi}{4})$ .

The presence of the obstacle is communicated to all the robots in the group. Each robot takes on a different role such that two subgroups will be formed. The robots with index  $i \in S_1 := \{2, \dots, N/2\}$  now follow the neighboring robot with corresponding index  $i - 1$ . Similarly, robots with index  $i \in S_2 := \{N/2 + 1, \dots, N - 1\}$  follow the neighboring robot with index  $i + 1$ . More precisely, the robot with index  $i$  tries to reach the following coordinates:

$$\begin{cases} (x_{i-1} + d \sin \frac{\pi}{6}, y_{i-1} + (-1)^i d \cos \frac{\pi}{6}), & \text{if } i \in S_1, \\ (x_{i+1} - d \sin \frac{\pi}{6}, y_{i+1} + (-1)^i d \cos \frac{\pi}{6}), & \text{if } i \in S_2. \end{cases} \quad (3)$$

These coordinates are considered with respect to a right-handed  $(x, y)$ -frame with the  $y$ -axis parallel to the strip boundary, and directed into the driving direction of the leader robots. Each robot still tries to stay in the preferred formation, but in order to do so only takes information of one neighbor into account. Moreover, the condition on the relative position between the neighboring robots  $N/2$  and  $N/2 + 1$  is suspended, which will lead to the splitting of the robot group. Notice that indifferent of the robot that observes the obstacle first, the group will split between robots  $N/2$  and  $N/2 + 1$ . This choice is motivated in Section 2.4.

Consider the situation for robot  $i$  where one of the following occurs:

- The desired position (3) cannot be reached,

- The obstacle is blocking the straight path between the present position of robot  $i$  and its desired position,
- Robot  $i$  does not detect its neighbor necessary to determine its preferred position.

If this situation occurs, the robot receives the order to follow the edge of the obstacle, keeping the obstacle on its right if  $i \in S_1$ , or its left if  $i \in S_2$ . This behavior is called wall-following. The robot continues to wall-follow around the obstacle until none of the above conditions is satisfied. After that, it assumes its desired position again. If all robots have past the obstacle, each robot is again able to reach its desired position in the preferred formation. In particular, robots  $N/2$  and  $N/2 + 1$  will meet again in their desired relative position. When this happens a signal is sent to all robots with the message that the group has past the obstacle. A simulation of the above described algorithm is presented in Figure 4 with  $N = 10$ .

*Remark.* It may occur that a robot cannot reach its desired position because it is located too far away from its present position. Then the robot simply rides towards the desired position at the maximum allowed velocity, trying to catch up.

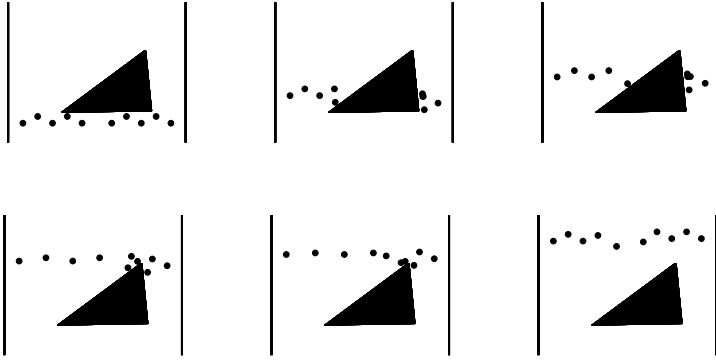
### 2.3 Multiple Obstacles

Suppose the robot group is already split into two subgroups and a robot in one of the subgroups encounters a second obstacle. The above obstacle avoidance algorithm can be made modular in order to solve this problem. A group can only split if both robots at the extremities of the group are leader robots, similar to the initial configuration. Assume group  $S_1$  encounters a second obstacle. Robot  $N/2$  is then turned into a leader robot. Instead of following a strip boundary it is ordered to follow the edge of the first obstacle, until it meets its neighbor  $N/2 + 1$  or until group  $S_1$  has past the second obstacle. In the latter case, robot  $N/2$  takes on its role as a follower of robot  $N/2 - 1$  again, in the former case it turns into a follower of  $N/2 + 1$ . The group  $S_1$  is split into the middle and the algorithm described in the previous section is performed with leader robots 1 and  $N/2$ . In order for each robot to know which role to assume, it keeps track of how many times its subgroup is split. If the number of robots  $N^*$  in a (sub)group is not even, then the indices of the robots where the robot group splits are  $\lfloor N^*/2 \rfloor$  and  $\lceil N^*/2 \rceil$ , where  $\lfloor \cdot \rfloor$  is the function returning the largest integer less than or equal to its argument, and similarly,  $\lceil \cdot \rceil$  returns the smallest integer greater than or equal to its argument.

Clearly, the number of times this splitting can be repeated is limited. We require a subgroup to consist of at least 3 robots: two leader robots on each side of the group, plus at least one robot in the middle attempting to follow both leaders while maintaining the formation structure. The middle robot ensures that the discs of sensed area of the separate robots overlap for all time instants.

### 2.4 Adaptation of the Basic Algorithm

Consider a worst case scenario as sketched in Figure 5. The robot formation splits into two subgroups, and the group on the left hand side moves through the gap between the



**Fig. 5.** A group of 10 robots passing an obstacle. It is demonstrated how the left subgroup adjusts its angles  $\alpha_i$  and  $\beta_i$  in order to spread and scan the area between left strip boundary and obstacle.

obstacle and the left boundary of the scanning strip. Once past the gap the robots in this subgroup have to spread, since the distance between the obstacle and the left boundary increases and we want to sense all of free space between the boundary and the obstacle. The obstacle has such a shape that the robots have to spread out across almost the entire width of the scanning strip before meeting a robot member of the right subgroup. The basic algorithm is modified as follows. When robot  $N/2$  (resp.  $N/2 + 1$ ) encounters the obstacle, it is now programmed to follow the obstacle's edge *until it meets its neighbor*  $N/2 + 1$  (resp.  $N/2$ ). Additionally, it ensures that its neighbor  $N/2 - 1$  (resp.  $N/2 + 1$ ) stays in its detection range by sending a signal to the other robots of its subgroup to increase the angle  $\pi/4$  of (3). This changes the desired position of each robot in the subgroup resulting in a stretching of the group, as far as necessary.

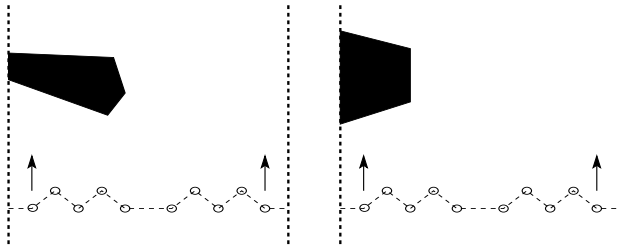
The above modified algorithm justifies our choice of initial formation and width of the scanning strip. If we had naively chosen a value  $(N - 1)d$  as the width of a scanning strip, the initial preferred robot formation would be able to span this entire distance, namely by forming a line with the angles defined in Section 2.1 equal to  $\alpha_i = -\beta_i = \pi/2$ . However, one subgroup, consisting of only half of the number of robots, would not be able to span this distance, resulting in either an error message from the algorithm or in unscanned areas, if a situation described in Figure 5 was encountered.

Closely related to this observation is the choice to split the robot group precisely in the middle. Since the sensor range of each robot is limited and the robots operate in an unknown environment, the shape of each obstacle is unknown. To guarantee that the area around the obstacle is fully covered by the sensors, we have to supply a sufficient number of robots to both sides of the obstacle. For instance, when the shape of the obstacle in Figure 5 is known a priori, one can decide to send more than half of the robots to the left of the obstacle. Consider the case where the obstacle is reflected with respect to the vertical axis. In this case sending less than half of the robots to the right would lead to uncovered areas or an error message in the algorithm. With limited sensor information it is not possible to discriminate between the situation of Figure 5 and its reflected version. This leads us to always split the group into two equal parts.



## 2.5 Obstacles Located on the Boundary between Two Strips

Throughout the paper the obstacles are assumed to have a convex shape, in order to avoid robot groups getting stuck in a dead end. However, there is one case of dead ends we cannot avoid by the above assumption. A dead end can occur when an obstacle is located on the boundary between two strips, as presented on the left hand side of Figure 6. Since the robots have limited sensor information, they cannot conclude a priori whether an encountered obstacle stretches out into a neighboring strip or not. We are forced to let the algorithm run until a dead end is observed.



**Fig. 6.** Two situations where an obstacle is located on the boundary between strips. On the left hand side a dead end situation arises; on the right hand side one of the leader robots guides the group around the obstacle.

Before tackling the dead end problem, let us treat the case presented on the right hand side of Figure 6, which does not lead to a dead end situation. Consider an  $(x, y)$ -frame with the  $y$ -axis parallel to the strip boundary, and directed into the driving direction of the leader robots. When the leader robot encounters the obstacle, the algorithm assigns to this leader a wall-following procedure around the obstacle. The leader keeps the obstacle either on its right or left (depending on its position in the robot formation) while moving into the interior of the strip away from the strip boundary. As can be concluded from the picture, the  $y$ -coordinate of the leader increases while moving around the obstacle. We wish to keep the velocity component of the leader robot parallel to the strip boundary equal to  $v$ . Since the robot deviates from its straight path parallel to the strip boundary, this implies it has to speed up. When the leader reaches the strip boundary again, it switches back to the original task of moving parallel to the boundary.

Now consider the left hand side of Figure 6. A dead end is detected by the algorithm when two conditions are satisfied:

- one of the leader robots cannot move into the desired direction parallel to the strip boundary, because an obstacle is blocking the way.
- when the leader robot starts wall-following the obstacle as described above, the value of its  $y$ -coordinate decreases.

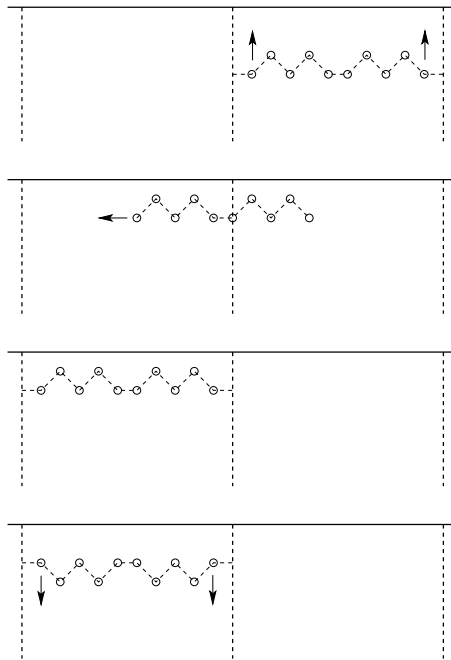
As soon as a dead end is observed by the leader robot, it changes its behavior and stops its wall following algorithm. Instead, it projects its corresponding strip boundary  $(N/2 - 1)d/8$  units outwards and resumes the original scanning algorithm with respect

to the new boundary. If the extra width turns out to be insufficient to guide the robot subgroup around the obstacle outside of the original scanning strip, the boundary is projected a second (third,...) time. This way the subgroup which was stuck in the dead end is guided around the obstacle. When both subgroups reestablish contact, the leader robot returns to the original strip boundary. This behavior is faster and easier to implement than a turning-back scenario, where the subgroup of robots which meets a dead end retraces its steps to go around the obstacle inside the original scanning strip.

*Remark.* The above situation with a solid wall as strip boundary, forcing a turning-back maneuver, is precluded.

### 2.6 The Transition from One Strip to the Next

When the robot group reaches the end of a scanning strip, it needs to be transported to the next strip. This is done in a few easy steps. Consider the situation of Figure 7. First the right leader changes its behavior into that of a robot in the interior of the formation, i.e. it tries to attain the desired formation. The left leader moves  $(N/2 - 1)d$  units to the left perpendicular to the strip boundary. The rightmost robot resumes its leader role and all robots reverse their forward direction with respect to the desired direction in the previous strip. At this moment the robots are not yet positioned in the desired formation: the indices of the robots are reversed. Each robot  $i$  assumes a new index

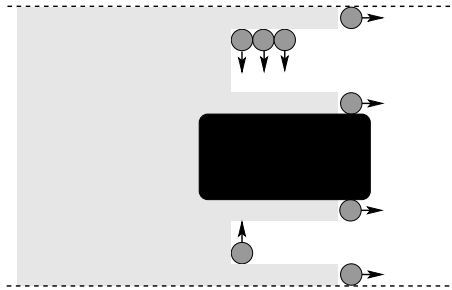


**Fig. 7.** The robot group moves from the end of a scanning strip to the start of the next strip.

number  $f(i) = (N + 1) - i$ , and is ordered to reassume its desired position in the robot group without the leader robots advancing. The preferred formation is attained, and the robots are ready to start the algorithm in the next strip. Naturally, every time the end of a strip is reached, the roles of left and right leader alternate, so that the robot group does not get trapped into a loop consisting of two strips.

### 3 Comparison with Complete Coverage Algorithms

The algorithm presented in this paper is related to the complete coverage algorithm presented in [10], which is depicted in Figure 8. The authors of [10] propose the following robot configuration: 2 leader robots, following the strip boundaries, and a number of interior robots, traveling back and forth between the strip boundaries physically covering all the free space. Contrary to the algorithm proposed in the present paper, the leader robots maintain line-of-sight contact between each other. When an obstacle appears between the two leaders the line-of-sight contact is lost and the obstacle is detected. An appropriate control action is then taken by splitting the platoon and the algorithm is repeated on both sides of the obstacle. The splitting procedure includes the creation of two extra leader robots, as shown in Figure 8. Remark that the leaders are allowed to move ahead of the rest of the robot group and hence group coherence is not maintained or desired, contrary to our approach.



**Fig. 8.** A depiction of the complete coverage algorithm by Rekleitis et al.

In the remainder of this section we will compare speed of performance of the present algorithm with the algorithm of [10]. In order to do so, realistic distance values are considered. Chemical vapor sensors detecting mines have a range  $s_t = 1.70$  m. Obstacles and other robots can be detected by laser based sensors with a range of  $s_r = 3.3$  m such that (2) is satisfied. Assume the robots themselves possess a diameter of 0.3 m and set the fixed interdistance  $d$  between neighboring robots in the preferred formation equal to  $s_r$ . With  $N$  the number of robots in the group, this yields a strip width of  $1.65(N - 2)$  m.

When no obstacles are encountered, the robots are allowed to move at a preset maximum velocity  $v_{\max}$ . In the algorithm of the present paper  $v_{\max}$  is directed parallel to the strip boundary, whereas the interior robots in [10] travel back and forth inside the strip at  $v_{\max}$ . It can be proven that for the latter case with the values given above the speed of progress parallel to the strip boundary is  $v_{\max}/6$ .

In the presence of obstacles a comparison is more difficult. First consider the complete coverage algorithm [10]. As can be concluded from Figure 8, in the presence of an obstacle the robots will advance faster parallel to the strip boundary, since the space occupied by the obstacle does not have to be covered. The robot group will proceed fastest when the shape of the obstacle is such that there is no space left for the robots to travel back and forth between obstacle and strip boundary. Hence, depending on size and shape of the obstacle the robots advance with a speed between  $v_{\max}/6$  and  $v_{\max}$ . Now, consider the algorithm of the present paper. Some interior robots perform wall-following around the obstacles. This implies their path is longer than the path of the leader robots. If the leader robots keep moving at the maximum allowed velocity, those interior robots will never again be able to reach their desired position inside the formation after the obstacle is past. Hence, when an obstacle is encountered the leaders have to take on a velocity  $v_0$  which is smaller than  $v_{\max}$ . This velocity  $v_0$  is determined as follows. The middle robots  $N/2$  and  $N/2 + 1$  transmit their positions via the other robots to their respective leader robots. The leaders adjust their velocity  $v_0$  such that the difference between their  $y$ -coordinate and the  $y$ -coordinate of the corresponding robot  $N/2$  or  $N/2 + 1$  stays at all time within a prespecified bound. The middle robots only slow down the group significantly during the first and last stage of their obstacle following, i.e. when moving away from or towards the strip boundary without significantly advancing parallel to it. As soon as there is enough free space ahead of the middle robots, the subgroup is again allowed to move parallel to the strip boundary with a speed close to  $v_{\max}$ .

From the above observations the following is concluded. The robot group in the present algorithm slows down to pass an obstacle, but for most of the time the speed will be close to  $v_{\max}$ . The robot group of the complete coverage algorithm speeds up when passing an obstacle, but for most obstacles the algorithm still requires a robot group moving back and forth between the obstacle and the strip boundary. This implies that the increased speed will on average be closer to  $v_{\max}/6$  than to  $v_{\max}$ . Hence, in generic cases, the present algorithm performs faster than the complete coverage strategy even in the presence of obstacles.

## 4 Conclusions

The present paper described an algorithm for multi-robot exploration in an unknown environment. The algorithm guarantees that all free space is covered by the robot sensors. The robots form a zigzag-shaped formation which scans the area in strips. In order to pass an obstacle, of which size and shape are not known a priori, the robot group splits in the middle. If necessary, the zigzag shape of each subgroup may stretch out in order to cover the free area between the obstacle and the strip boundary. The algorithm is also able to handle obstacles located on the strip boundary.

**Acknowledgements.** This paper presents research results of the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office. The scientific responsibility rests with its authors.

## References

1. Ota, J.: Multi-agent robot systems as distributed autonomous systems. *Advanced Engineering Informatics* **20** (2006) 59 – 70
2. Burgard, W., Moors, M., Stachniss, C., Schneider, F.: Coordinated multi-robot exploration. *IEEE Transactions on Robotics* **21** (2005) 376–386
3. Kerr, W., Spears, D., Spears, W., Thayer, D.: Two formal gas models for multi-agent sweeping and obstacle avoidance. In: *Formal Approaches to Agent-Based Systems, Third International Workshop*. (2004) 111–130
4. Keymeulen, D., Decuyper, J.: The fluid dynamics applied to mobile robot motion: the stream field method. In: *Proceedings of 1994 IEEE International Conference on Robotics and Automation*, Piscataway, NJ, USA (1994) 378–385
5. Choset, H.: Coverage for robotics – a survey of recent results. *Annals of Mathematics and Artificial Intelligence* **31** (2001) 113–126
6. Cortés, J., Martínez, S., Karatas, T., Bullo, F.: Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation* **20** (2004) 243–255
7. Kurabayashi, D., Ota, J., Arai, T., Yosada, E.: Cooperative sweeping by multiple robots. In: *Proc. 1996 IEEE International Conference on Robotics and Automation*. (1996)
8. Wong, S., MacDonald, B.: Complete coverage by mobile robots using slice decomposition based on natural landmarks. In: *Proc. Eighth Pacific Rim International Conference on Artificial Intelligence. Lecture Notes in Artificial Intelligence. Volume 3157*. (2004) 683–692
9. Zheng, X., Jain, S., Koenig, S., Kempe, D.: Multi-robot forest coverage. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. (2005)
10. Rekleitis, I., Lee-Shue, V., New, A.P., Choset, H.: Limited communication, multi-robot team based coverage. In: *Proc. 2004 IEEE International Conference on Robotics and Automation*. (2004)
11. Kong, C.S., Peng, N.A., Rekleitis, I.: Distributed coverage with multi-robot system. In: *Proceedings of 2006 IEEE International Conference on Robotics and Automation*, Orlando, Florida, USA (2006) 2423–2429
12. Gage, D.: Many-robots mcm search systems. In: *Proceedings of Autonomous Vehicles in Mine Countermeasures Symposium*. (1995)

# Tracking of Constrained Submarine Robot Arms

Ernesto Olguín-Díaz and Vicente Parra-Vega

Robotics and Advanced Manufacturing Division, CINVESTAV-Salttilo Campi, Mexico  
(ernesto.olguin, vicente.parra)@cinvestav.edu.mx

**Abstract.** Despite the appearance of impressive submarine robot arms (SRA), simple posture (position/orientation) regulators are implemented nowadays and tracking still remains an open issue, let alone the force/posture tracking goal. The main challenge to achieve simultaneous tracking of posture and force seems the complex dynamical structure, the difficulty to measure precisely the inertial parameters of SRA and the access to the full state of SRA. In this paper, on one hand the nature of the control problem of SRA is discussed in contrast to typical AUV, and on the other hand, we introduce a model-free second order sliding mode controller to finally yield a model-free smooth control scheme to achieve tracking of force and posture of the SRA. Structural properties of the submarine robot dynamic are used to design a the passivity-based control to produce a viable tracking control scheme for constrained SRA. Helpful and succinct discussions of its inherent open-loop and closed-loop properties are presented, which provide additional insight into the control problem of SRA. A representative simulation study is presented.

**Keywords.** Underwater vehicles, Force/position control, Sliding Modes Control.

## 1 Introduction

Though modern underwater autonomous vehicle (AUV) assumes submarine (underwater) robot arms (SRA) as its end-effector, the control of an AUV poses a rather different problem with respect to control problem of SRA, since the former focusses on the navigation capabilities of its main body while the later study the control of a free-floating robot arm. In this case, the robot arm on the AUV is considered as SRA, and its treatment deserves a separate attention due to the subtle complexities of SRA in its own right. In this section, we elaborate more on the intrinsic dynamic control nature of AUV with focus on the tracking problem of SRA subject to interaction to rigid underwater objects.

### 1.1 The SRA Problem

In the last decade, we have witness a surprising leap on scientific knowledge and technological achievements for AUV, from a simple torpedo to modern AUV. Those vehicles pose at the same time tantamount scientific and technological challenges in robotics, control, man-machine interfaces and mechatronics. Modern efforts around the world on AUV focuses more still in how to provide an acceptable level of (perhaps autonomous

or automatic) navigation capabilities of the main body of the AUV, rather than in the manipulation capabilities of its tools, perhaps a SRA, carried on the AUV. Therefore, we bring the attention of a new breed of AUV which main task is manipulation, perhaps with more than one robot arm, where the underlined issue is that the main body, the AUV, is considered as the fully actuated (or underactuated) free/floating base of the robot arm. In this case, we reasonably assume that the AUV is several times heavier than the SRA so as to provide inertial decoupling between the AUV and the robot arm. In this case, we have  $n$  thrusters to drive the AUV and  $m$  actuators to drive the SRA. Pioneering efforts on SRA were focused on motion control with simple PD regulators in unconstrained motion similar to the case of fixed-base robots in our labs. Acceptable performance of tracking has been proposed using more complicated (saturated or non-linear) PID schemes and few model-based controllers have been proposed for tracking, under lab conditions [16, 5]. Of course, stable contact for SRA is a more complex problem in comparison to the typical force/position control problem of robot manipulators fixed to ground because not only due complementary complex dynamics are presented in SRA, such as buoyancy and added masses, but to de fact that the vehicle reference frame is not longer inertial, see [12, 6].

However, more interesting submarine tasks require the more challenging problem of establishing stable contact while moving along the contact surface, like pushing itself against a wall or polishing a sunken surface vessel surface or manipulating tools on submarine pipe lines, forces are presented, and little is known about the structural properties of these contact forces, let alone exploit them either for design or control. This problem leads us to study the simultaneous force and pose (position and orientation) control of SRA under realistic conditions, thus we have the following assumptions

- *i.* the dynamical model, and its parameters, are hardly known in practice
- *ii.* the full state is not available
- *iii.* the geometric description of the contact surface is not completely known

As a first step to deal with this problem, in this paper we consider assumptions *i*), while *ii*) and *iii*) are assumed available. Notice that in any case, the controlled trust force of the AUV must not only maintain stable contact, but must move, or keep still, the *base* of the SRA in whatever position is required to achieve tracking of desired time-varying trajectories of force and posture of *the end-effector*. How to achieve this is still an open problem, and subject of future study. Finally, we emphasize that issues *i-iii* ( and *iv* mentioned below), posse such challenges nowadays that deserve particular attention away from the already complex issues of AUV, and thus the control problem of SRA requires a particular treatment, which has been been completed addressed in the AUV literature.

## 1.2 The Constrained SRA Problem

The main general reason that force/posture problem remains rather an open problem is that we really know little about. On one hand, how to model and control properly a fully free/floating immersed vehicle constrained by rigid object. On the other hand, the submarine force control technology lies behind system requirements, such as very fast

sampling and uniform rates of sensors and actuators, even when the bandwidth of the submarine robot is very low.

Despite brilliant, for the simplicity of this complex problem, control schemes for free motion submarine robots in the past few years, in particular those of [17, 4, 2] those schemes does not guarantee formally convergence of tracking errors, let alone simultaneous convergence of posture and contact force tracking errors. There are several results that suggest empirically that a simple PD control structure behaves as stiffness control for submarine robots to produce acceptable low performance of contact tasks. However for more precise and fast tasks, the simultaneous convergence of time-varying contact forces and posture remains an open problem. Since  $i$  is of great concern, control schemes which does not depend on the model or its regressor are quite important since for AUV and SRA the role of model-free controllers are very important because it is very hard to known exactly the dynamic model and its dynamic parameters. Neural network could be an option, however because of the limited processing capabilities of typical SRA, we need to resort on other control schemes. Recently, some efforts have focused on how to obtain simple control structures to control the time-varying pose of the AUV under the assumption that the relative velocities are low [2, 8].

For force control of SRA under assumption  $i$ , virtually none complete control system has been published. However, to move forward more complex force controllers, we believe that a better understanding of the structural properties of submarine robots in stable contact to rigid objects are required. To this end, we consider the rigid body dynamics of SRA subject to holonomic constraints (rigid contact), which exhibits similar structural properties of fixed-base constrained robots. Thus, in this paper we have chosen the orthogonalization principle [9] to extend from fix base to free-floating base to propose a simple, yet high performance, controller with advanced tracking stability properties.

### 1.3 The Constrained Optimal SRA Problem

A fourth issue appears here: when the SRA achieves stable contact, must the *SRA's base* be kept still in a given constant position or must it track a given time-varying equilibria to better accommodate the reaction contact torques/force, which are propagated over the AUV+SRA body till its base? That is, we have:  $iv)$  for a given time-varying desired vector of contact forces, which is the best time-varying posture of the *SRA's base* to achieve a certain level of optimal contact, for a given cost function? This cost function may constraint motion of the *SRA's base* within an envelope to achieve better manipulability index or to minimize control effort/energy without compromising maneuverability, while tracking force/posture trajectories. Evidently, at the present stage of this research, this problem is not studied in this paper.

### 1.4 Contribution

Firstly, we draw the attention of the control problem of constrained SRA, which deserves a particular treatment apart to the AUVs control problem. to this end we go through the full dynamic model. Then, a quite simple force/posture model-free decentralized control structure is proposed in this paper, which guarantees robust tracking of



time-varying contact force and posture, conservative knowledge of submarine dynamics. The proposed controller stands itself as a new controller, whose closed-loop stability properties, in the sense of Lyapunov and Variable Structure Systems, are presented. A representative set of simulations under fluid disturbances show the robust tracking.

## 2 The Model

### 2.1 A Free Floating Submarine Robot Arm: SRA

The model of a submarine can be obtained with the momentum conservation theory and Newton's second law for rigid objects in free space via the Kirchhoff formulation [11], the inclusion of hydrodynamic effects such as added mass, friction and buoyancy and the account of external forces/torques like contact effects [6]. The model is then expressed by the next set of equations:

$$M_v \dot{\boldsymbol{\nu}} + C_v(\boldsymbol{\nu})\boldsymbol{\nu} + D_v(\boldsymbol{\nu}, t)\boldsymbol{\nu} + \mathbf{g}_v(\mathbf{q}) = \mathbf{u} + \mathbf{F}_c^{(v)} + \boldsymbol{\eta}_v(\boldsymbol{\nu}, t), \quad (1)$$

$$\boldsymbol{\nu} = J_\nu(\mathbf{q})\dot{\mathbf{q}}. \quad (2)$$

From this set, (1) is called the dynamic equation while (2) is called the kinematic equation. The generalized coordinates vector  $\mathbf{q} \in \mathbb{R}^6$  is given on one hand by the 3 Cartesian positions  $x, y, z$  of the origin of the submarine frame ( $\Sigma_v$ ) with respect to a inertial frame ( $\Sigma_0$ ), and on the other hand by any set of attitude parameters that represent the rotation of the vehicle's frame with respect to the inertial one. Most common sets of attitude representation such a Euler angles, in particular roll-pitch-yaw ( $\phi, \theta, \psi$ ), use only 3 variables (which is the minimal number of orientation variables). Then, for a submarine, the generalized coordinates  $\mathbf{q} = (x_v, y_v, z_v, \phi_v, \theta_v, \psi_v)$  represents its 6 degrees of freedom. The vehicle velocity  $\boldsymbol{\nu} \in \mathbb{R}^6$  is the vector representing both linear and angular velocity of the submarine in the vehicle's frame. This vector is then defined as  $\boldsymbol{\nu} = (\mathbf{v}_v^{(v)T}, \boldsymbol{\omega}^{(v)T})^T$ . The relationship between this vector and the generalized coordinates is given by the kinematic equation. The linear operator  $J_\nu(\mathbf{q}) \in \mathbb{R}^{6 \times 6}$  in (2), is built by the concatenation of two transformations. The first is  $J_q(\mathbf{q}) \in \mathbb{R}^{6 \times 6}$  which converts time derivatives of attitude parameters in angular velocity. The second is  $J_R(\mathbf{q}) = \text{diag}\{R_0^v, R_0^v\} \in \mathbb{R}^{6 \times 6}$  which transforms a 6 dimension tensor from the inertial frame to vehicle's frame. Thus, the linear operator is defined as  $J_\nu(\mathbf{q}) = J_R^T(\mathbf{q})J_q(\mathbf{q})$ . A detailed discussion on the terms of (1) can be found in [1]. The disturbance  $\boldsymbol{\eta}_v(\boldsymbol{\nu}, \boldsymbol{\zeta}(t), \dot{\boldsymbol{\zeta}}(t))$  of the surrounding fluid depends mainly in the incidence velocity, i.e. the relative velocity of the vehicle velocity and the fluid velocity  $\boldsymbol{\zeta}(t)$ . The last is a non-autonomous function, but an external perturbation. This disturbance has the property of  $\boldsymbol{\eta}_v(\boldsymbol{\nu}, 0, 0) = 0$ . That is that all the disturbances are null when the fluid velocity and acceleration are null.

The dynamic model (1)-(2) can be rearranged by replacing (2) and its time derivative into (1). The result is one single equation model:

$$M_q(\mathbf{q})\ddot{\mathbf{q}} + C_q(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + D_q(\cdot)\dot{\mathbf{q}} + \mathbf{g}_q(\mathbf{q}) = \mathbf{u}_q + \boldsymbol{\tau}_c + \boldsymbol{\eta}_q(\dot{\mathbf{q}}, \boldsymbol{\zeta}(t), \dot{\boldsymbol{\zeta}}(t)); \quad (3)$$

which, whenever  $\zeta(t) = \dot{\zeta}(t) = 0$ , i.e.  $\eta_q(\cdot) = 0$ , has the form of any Lagrangian system. Its components fulfills all properties of such systems i.e. definite positiveness of inertia and damping matrices, skew symmetry of Coriolis matrix and appropriate bound of all components [15]. The contact effect is also obtained by the same transformation. However it can be expressed directly from the contact wrench in the inertial frame ( $\Sigma_0$ ) by the relationship  $\tau_c = J_\nu^T(\mathbf{q})\mathbf{F}_c^{(v)} = J_q^T(\mathbf{q})\mathbf{F}_c^{(0)}$ , where the contact force  $\mathbf{F}_c^{(0)}$  is the one expressed in the inertial frame. By simplicity it will be noted as  $\mathbf{F}_c$  from this point further. The relationship with the one expressed in the vehicle's frame is given by  $\mathbf{F}_c = J_R^T(\mathbf{q})\mathbf{F}_c^{(v)}$ . This wrench represents the contact forces/torques exerted by the environment to the submarine as if measured in a non moving frame. These forces/torques are given by the normal force of an holonomic constraint when in contact and the friction due to the same contact. For simplicity in this work, tangential friction is not considered. The equivalent of the disturbance is obtained also with the linear operator given as:  $\eta_q(\cdot) = J_\nu^T(\mathbf{q})\eta_v(\cdot)$ .

## 2.2 Contact Force Due to an Holonomic Constraint

A holonomic constraint (or infinitely rigid contact object) can be expressed as a function of the generalized coordinates of the submarine as

$$\varphi(\mathbf{q}) = 0, \quad (4)$$

with  $\varphi(\mathbf{q}) \in \mathbb{R}^r$ , where  $r$  stands for the number of independent contact points between the SRA and the motionless rigid object. Equation (4) means that stable contact appears while the SRA submarine does not deattach from the object  $\varphi(\mathbf{q}) = 0$ . Evidently all time derivatives of (4) are zero, which for  $r = 1$

$$J_\varphi(\mathbf{q})\dot{\mathbf{q}} = 0, \quad (5)$$

where  $J_\varphi(\mathbf{q}) = \frac{\partial \varphi(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{r \times n}$  is the constraint jacobian. Last equation means that velocities of the submarine in the directions of constraint jacobian are restricted to be zero. This directions are then normal to the constraint surface  $\varphi(\mathbf{q})$  at the contact point. As a consequence, the normal component of the contact force has exactly the same direction as those defined by  $J_\varphi(\mathbf{q})$ , consequently, the contact force wrench can be expressed as

$$\mathbf{F}_c = J_{\varphi+}^T(\mathbf{q})\lambda, \quad (6)$$

where  $J_{\varphi+}(\mathbf{q}) \triangleq \frac{J_\varphi}{\|J_\varphi\|}$  is a normalized version of the constraint jacobian;  $\lambda \in \mathbb{R}^r$  is the magnitude of the normal contact force at the origin of vehicle frame:  $\lambda = \|\mathbf{F}_c\|$ . The free moving model expressed by (1)-(2), when no fluid disturbance and in contact with the holonomic constraint can be rewritten as:

$$M_v \dot{\boldsymbol{\nu}} + \mathbf{h}_v(\mathbf{q}, \boldsymbol{\nu}, t) = \mathbf{u} + J_R^T(\mathbf{q})J_{\varphi+}^T(\mathbf{q})\lambda, \quad (7)$$

$$\boldsymbol{\nu} = J_\nu(\mathbf{q})\dot{\mathbf{q}}, \quad (8)$$

$$\varphi(\mathbf{q}) = 0, \quad (9)$$

where  $\mathbf{h}_v(\mathbf{q}, \boldsymbol{\nu}, t) = C_v(\boldsymbol{\nu})\boldsymbol{\nu} + D_v(\mathbf{q}, \boldsymbol{\nu}, t)\boldsymbol{\nu} + \mathbf{g}_v(\mathbf{q})$ . Equivalently, the model (3) is also expressed as

$$M_q(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}_q(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{u}_q + J_{\varphi}^T(\mathbf{q})\lambda, \quad (10)$$

$$\varphi(\mathbf{q}) = 0, \quad (11)$$

with  $\mathbf{h}_q(\mathbf{q}, \dot{\mathbf{q}}, t) = C_q(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + D_q(\mathbf{q}, \dot{\mathbf{q}}, t)\dot{\mathbf{q}} + \mathbf{g}_q(\mathbf{q})$  and  $J_{\varphi}(\mathbf{q}) = J_{\varphi+}(\mathbf{q})J_q(\mathbf{q})$ . Equations (10)-(11) are a set of Differential Algebraic Equations index 2 (DAE-2). To solve them numerically, a DAE solver is required. This last representation has the same structure and properties as those reported in [3].

### 3 Control Design

The introduction of a so called Orthogonalization Principle has been a key in solving, in a wide sense, the force control problem of a robot manipulators. This physical-based principle states that the orthogonal projection of contact forces and joint generalized velocities are complementary, and thus its dot product is zero. Relying on this fundamental observation, passivity arises from torque input to generalized velocities, in open-loop. To preserve passivity, then, the closed-loop system must satisfy the passivity inequality for controlled generalized error velocities. This is true for robot manipulators with fixed frame, and here we extend this approach for robots whose reference frame is not inertial, like SRA.

#### 3.1 Orthogonalization Principle and Linear Parametrization

Similar to [7], the orthogonal projection of  $J_{\varphi}(\mathbf{q})$ , which arises onto the tangent space at the contact point, is given by following operator

$$Q(\mathbf{q}) \triangleq I_n - J_{\varphi}^T(\mathbf{q})J_{\varphi}(\mathbf{q}) \in \mathbb{R}^{n \times n}, \quad (12)$$

where  $I_n \in \mathbb{R}^{r \times n}$  is an identity matrix. Notice that  $\text{rank}\{Q(\mathbf{q})\} = n - r$  since  $\text{rank}\{J_{\varphi}(\mathbf{q})\} = r$ . Also notice that  $Q\dot{\mathbf{q}} = \dot{\mathbf{q}}$  due to the definition of  $Q$  and (5). Therefore, according to the Orthogonalization Principle, the integral of  $(\tau, \dot{\mathbf{q}})$  is upper bounded by  $-\mathcal{H}(t_0)$ , for  $\mathcal{H}(t) = K + P$  whenever  $\boldsymbol{\eta}_v(\boldsymbol{\nu}, \zeta(t), \dot{\zeta}(t)) = 0$ , because  $\dot{\mathbf{q}}^T J_{\varphi}^T(\mathbf{q})\lambda \doteq 0$ . Then passivity arise for fully immersed submarines without inertial frame and no fluid disturbances. This conclusion gives a very useful and promising theoretical framework, similar to the approach of passivity-based control for fix-base robot arms. On the other hand, it is known that the dynamic equation (10) can be linearly parameterized as follows

$$M_q(\mathbf{q})\ddot{\mathbf{q}} + C_q(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + D_q(\cdot)\dot{\mathbf{q}} + \mathbf{g}_q(\mathbf{q}) = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\theta}, \quad (13)$$

where the regressor  $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \in \mathbb{R}^{n \times p}$  is composed of known nonlinear functions and  $\boldsymbol{\theta} \in \mathbb{R}^p$  by  $p$  unknown but constant parameters.

### 3.2 Change of Coordinates

In order to design the controller, we need to work out the open loop error equation using (13), in terms of nominal references ( $\dot{\mathbf{q}}_r$ , named  $\mathbf{Y}_r$ , as follows. Consider

$$M_q(\mathbf{q})\ddot{\mathbf{q}}_r + [C_q(\mathbf{q}, \dot{\mathbf{q}}) + D_q(\cdot)]\dot{\mathbf{q}}_r + \mathbf{g}_q(\mathbf{q}) = \mathbf{Y}_r(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\Theta, \quad (14)$$

where  $\dot{\mathbf{q}}_r$  is the time derivative of ( $\dot{\mathbf{q}}_r$ , to be defined. Then the open loop (10) can be written by adding and subtracting (14) as

$$M_q(\mathbf{q})\dot{\mathbf{s}} = -[C_q(\mathbf{q}, \dot{\mathbf{q}}) + D_q(\cdot)]\mathbf{s} - \mathbf{Y}_r(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\Theta + J_{\varphi}^T(\mathbf{q})\lambda + \mathbf{u}_q, \quad (15)$$

where  $\mathbf{s} \triangleq \dot{\mathbf{q}} - \dot{\mathbf{q}}_r$  is called the extended error. The problem of control design for the open loop (15) is to find  $\mathbf{u}_q$  such that exponential convergence arises when  $\mathbf{Y}_r\Theta$  is not available.

### 3.3 Orthogonal Nominal Reference

Consider that  $\mathbf{q}_d(t)$  and  $\lambda_d(t)$  are the desired smooth trajectories of position and contact force, with  $\tilde{\mathbf{q}} \triangleq \mathbf{q}(t) - \mathbf{q}_d(t)$  and  $\tilde{\lambda} \triangleq \lambda(t) - \lambda_d(t)$  as the position and force tracking errors, respectively. Then, let following reference  $\dot{\mathbf{q}}_r$  be:

$$\begin{aligned} \dot{\mathbf{q}}_r = Q & \left( \dot{\mathbf{q}}_d - \sigma\tilde{\mathbf{q}} + \mathbf{S}_{dp} - \gamma_1 \int_{t_0}^t \text{sgn}\{\mathbf{S}_{qp}(t)\}dt \right) \\ & + \beta J_{\varphi}^T \left( \mathbf{S}_F - \mathbf{S}_{dF} + \gamma_2 \int_{t_0}^t \text{sgn}\{\mathbf{S}_{qF}(t)\}dt \right), \end{aligned} \quad (16)$$

where the parameters  $\beta$ ,  $\sigma$ ,  $\gamma_1$  and  $\gamma_2$  are constant matrices of appropriate dimensions; and  $\text{sgn}(y)$  stands for the entrywise signum function of vector  $y$ , and

$$\mathbf{S}_p \triangleq \dot{\tilde{\mathbf{q}}} + \sigma\tilde{\mathbf{q}}, \quad \mathbf{S}_{dp} \triangleq \mathbf{S}_p(t_0)e^{-\alpha(t-t_0)}, \quad \mathbf{S}_{qp} \triangleq \mathbf{S}_p - \mathbf{S}_{dp}, \quad (17)$$

$$\mathbf{S}_F \triangleq \int_{t_0}^t \tilde{\lambda} dt, \quad \mathbf{S}_{dF} \triangleq \mathbf{S}_F(t_0)e^{-\eta(t-t_0)}, \quad \mathbf{S}_{qF} \triangleq \mathbf{S}_F - \mathbf{S}_{dF}, \quad (18)$$

with  $\alpha > 0$ ,  $\eta > 0$ . Using (16) into  $\mathbf{s}$ , it arises

$$\mathbf{s} = Q(\mathbf{q})\mathbf{S}_{vp} - \beta J_{\varphi}^T(\mathbf{q})\mathbf{S}_{vF}, \quad (19)$$

where the orthogonal extended position and force manifolds  $\mathbf{S}_{vp}$  and  $\mathbf{S}_{vF}$ , respectively, are given by

$$\mathbf{S}_{vp} = \mathbf{S}_{qp} + \gamma_1 \int \text{sgn}(\mathbf{S}_{qp}(\zeta))d\zeta, \quad (20)$$

$$\mathbf{S}_{vF} = \mathbf{S}_{qF} + \gamma_2 \int \text{sgn}(\mathbf{S}_{qF}(\zeta))d\zeta. \quad (21)$$

### 3.4 Model-free Second Order Sliding Mode Control

Consider the following nominal continuous control law:

$$\mathbf{u}_q = -K_d \mathbf{S} + J_{\bar{\varphi}}^T(\mathbf{q}) \left( -\lambda^d + \dot{S}_{dF} + \gamma_2 \tanh(\mu S_{qF}) + \eta S_{qF} \right) \quad (22)$$

with  $\mu > 0$  and  $K_d = K_d^T > 0, \in \mathbf{R}^{n \times n}$ . This nominal control, designed in the  $q$ -space can be mapped to the original coordinates model, expressed by the set (1)-(2), using the next relationship  $\mathbf{u} = J_v^{-T}(\mathbf{q}) \mathbf{u}_q$ . Thus, the physical controller  $\mathbf{u}$  is implemented in terms of a key inverse mapping  $J_v^{-T}$ .

**Closed-loop System.** The open loop system (15) under the continuous model-free second order sliding mode control (22) yields to

$$M_q \dot{\mathbf{s}} = -[C_q + D_q + K_d] \mathbf{s} - \mathbf{Y}_r \Theta + J_{\bar{\varphi}}^T(\dot{S}_{vF} + \eta S_{qF}) + \gamma_2 J_{\bar{\varphi}}^T Z + \tau^* \quad (23)$$

where  $Z = \tanh(\mu S_{qF}) - \operatorname{sgn}(S_{qF})$ , and  $\tau^* \equiv 0$  is useful for the passivity analysis.

#### Stability Analysis

**Theorem 1.** Consider a constrained submarine (10)-(33) under the continuous model-free second order sliding mode control (22). The Underwater system yields a second order sliding mode regime with local exponential convergence for the position, and force tracking errors.

**Proof.** A passivity analysis  $\langle S, \tau^* \rangle$  indicates that the following candidate Lyapunov function  $V$  qualifies as a Lyapunov function

$$V = \frac{1}{2} (\mathbf{s}^T M_q \mathbf{s} + \beta S_{vF}^T S_{vF}), \quad (24)$$

for a scalar  $\beta > 0$ . The time derivative of the Lyapunov candidate equation immediately leads to

$$\begin{aligned} \dot{V} &= -\mathbf{s}^T (K_d + D_q) \mathbf{s} - \beta \eta S_{vF}^T S_{vF} - \mathbf{s}^T \mathbf{Y}_r \Theta + \mathbf{s}^T \gamma_2 J_{\bar{\varphi}}^T Z \\ &\leq -\mathbf{s}^T K_d \mathbf{s} - \beta \eta S_{vF}^T S_{vF} + \|\mathbf{s}\| \|\mathbf{Y}_r \Theta\| + \|\mathbf{s}\| \|\gamma_2\| \|J_{\bar{\varphi}}\| \|Z\| \\ &\leq -\mathbf{s}^T K_d \mathbf{s} - \beta \eta S_{vF}^T S_{vF} + \|\mathbf{s}\| \|\epsilon\|, \end{aligned} \quad (25)$$

where it has been used the skew symmetric property of  $\dot{M} - 2C(q, \dot{q})$ , the boundedness of both the feedback gains and submarine dynamic equation (there exists upper bounds for  $M, C(q, \dot{q}), g(q), \dot{q}_r, \ddot{q}_r$ ), the smoothness of  $\varphi(q)$  (so there exists upper bounds for  $J_{\bar{\varphi}}$  and  $Q(q)$ ), and finally the boundedness of  $Z$ . All these arguments establish the existence of the functional  $\epsilon$ . Then, if  $K_d, \beta$  and  $\eta$  are large enough such that  $\mathbf{s}$  converges into a neighborhood defined by  $\epsilon$  centered in the equilibrium  $\mathbf{s} = 0$ , namely

$$\mathbf{s} \rightarrow \epsilon \text{ as } t \rightarrow \infty. \quad (26)$$

This result stands for local stability of  $\mathbf{s}$  provided that the state is near the desired trajectories for any initial condition. This boundedness in the  $\mathcal{L}_\infty$  sense, leads to the existence of the constants  $\epsilon_3 > 0$  and  $\epsilon_4 > 0$  such that

$$\|\dot{\mathbf{S}}_{vp}\|_{\mathcal{L}_\infty} < \epsilon_3, \quad (27)$$

$$\|\dot{\mathbf{S}}_{vF}\|_{\mathcal{L}_\infty} < \epsilon_4. \quad (28)$$

An sketch of the local convergence of  $\mathbf{S}_{vp}$  is as follows<sup>1</sup>. Locally, in the  $n - r$  dimensional image of  $Q$ , we have that  $\mathbf{S}_{qp}^* = Q\mathbf{S}_{qp} \in \mathbb{R}^n$ . Consider now that under an abuse of notation that  $\mathbf{S}_{qp} = \mathbf{S}_{qp}^*$ , such that for small initial conditions, if we multiply the derivative of  $\mathbf{S}_{qp}$  in (20) by  $\mathbf{S}_{qp}^T$ , we obtain

$$\mathbf{S}_{qp}^T \dot{\mathbf{S}}_{qp} = -\gamma_1 \|\mathbf{S}_{qp}\| + \mathbf{S}_{qp}^T \dot{\mathbf{S}}_{vp} \leq -\gamma_1 \|\mathbf{S}_{qp}\| + \|\mathbf{S}_{qp}\| \|\dot{\mathbf{S}}_{vp}\| \leq -(\gamma_1 - \epsilon_3) \|\mathbf{S}_{qp}\|$$

which have used (27), and  $\gamma_1 > \epsilon_3$ , to guarantee the existence of a sliding mode at  $\mathbf{S}_{qp}(t) = 0$  at time  $t \leq \|\mathbf{S}_{qp}(t_0)\|/(\gamma_1 - \epsilon_3)$ , and according to the definition of  $\mathbf{S}_{qp}$  (below (20)),  $\mathbf{S}_{qp}(t_0) = 0$ , which simply means that  $\mathbf{S}_{qp}(t) = 0$  for all time. We see that if we multiply the derivative of (21) by  $\mathbf{S}_{vf}^T$ , we obtain

$$\mathbf{S}_{qF}^T \dot{\mathbf{S}}_{qF} = -\gamma_2 \|\mathbf{S}_{qF}\| + \mathbf{S}_{qF}^T \dot{\mathbf{S}}_{vF} \leq -\gamma_2 \|\mathbf{S}_{qF}\| + \|\mathbf{S}_{qF}\| \|\dot{\mathbf{S}}_{vF}\| \leq -(\gamma_2 - \epsilon_4) \|\mathbf{S}_{qF}\|$$

which have used (28), and  $\gamma_2 > \epsilon_4$ , to guarantee the existence of a sliding mode at  $\mathbf{S}_{qF}(t) = 0$  at time  $t \leq \|\mathbf{S}_{qF}(t_0)\|/(\gamma_2 - \epsilon_4)$  and, according to (21),  $\mathbf{S}_{qF}(t_0) = 0$ , which simply means that  $\mathbf{S}_{qf}(t) = 0$  for all time, which simply implies that  $\lambda \rightarrow \lambda_d$  exponentially fast.

## 4 Simulation Results

Simulations has been made on a simplified platform of a real submarine. Data has been obtained from the Vortex system of IFREMER. Simulator presents only vertical planar results (only in the x-z plane), so the generalized coordinates for this case of study are:

$$\mathbf{q} = \begin{pmatrix} x_v \\ z_v \\ \theta_v \end{pmatrix} \quad (29)$$

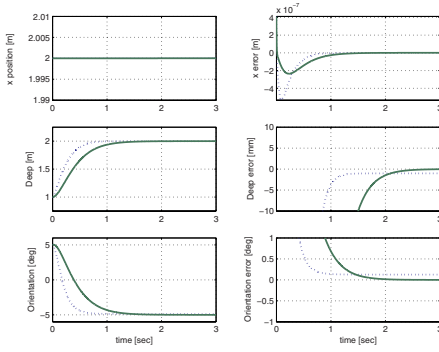
The vehicle velocities are

$$\mathbf{v} = \begin{pmatrix} u_v \\ w_v \\ q_v \end{pmatrix} \quad (30)$$

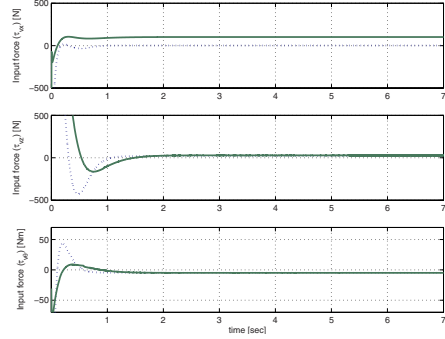
where  $u_v$  and  $w_v$  are linear velocities (surge and heave) and  $q_v$  is the angular velocity in the x-z plane. The holonomic constraint is given by a vertical surface located at 2 meters from the origin. This is expressed as:

$$\varphi(\mathbf{q}) \equiv x - 2 \quad (31)$$

<sup>1</sup> The strict analysis follows Liu, *et. al.*



**Fig. 1.** Generalized coordinates  $q$  and errors  $\tilde{q} = q(t) - q^d \nu$  for set-point disturbance-free case (continuous line for model-free second order sliding mode control, dotted line for PD control).



**Fig. 2.** Inputs controlled forces  $u$ , in vehicle's frame for set-point disturbance-free case (continuous line for model-free second order sliding mode control, dotted line for PD control).

Initial conditions were calculated to be at the contact surface with no force. Simulations with simple PD were also performed as a comparison tool. The model-free control parameters are as follows:

$K_d$	$\gamma_1$	$\gamma_2$	$\sigma$	$\alpha$	$\beta$	$\eta$	$\mu$
$200\hat{M}_v$	0.0025	$10^{-3}$	5	4	0.025	1000	1

where  $\hat{M}_v$  is made by the diagonal values of the constant Inertia matrix when expressed in the vehicle's frame. For de PD the gains were defined as  $K_p = 100\hat{M}_v$  and  $K_d = 20\hat{M}_v$ .

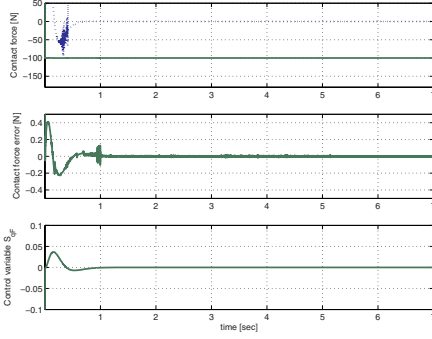
### 4.1 Numerical Considerations

To compute the value of  $\lambda$ , the constrained Lagrangian that fulfils the constrained movement, can be calculated using the second derivative of the holonomic restriction:  $\ddot{\varphi}(q) = 0$ . Then, using the dynamic equation (either of them) and after some algebra its expression becomes either:

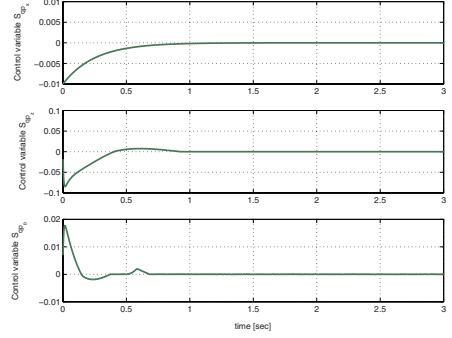
$$\lambda = \left[ J_\varphi J_\nu^{-1} M_v^{-1} J_R^T J_{\varphi+}^T \right]^{-1} \left( J_\varphi J_\nu^{-1} M_v^{-1} (\mathbf{h}_v - \mathbf{u}) - \frac{d}{dt} (J_\varphi J_\nu^{-1}) \nu \right), \quad (32)$$

$$= \left[ J_\varphi M_q^{-1} J_{\tilde{\varphi}}^T \right]^{-1} \left( J_\varphi M_q^{-1} (\mathbf{h}_q - \mathbf{u}_q) - \dot{J}_\varphi \dot{q} \right). \quad (33)$$

The set of eqns. 7)-(8)-(32) or the set (10)-(33) describes the constrained motion of the submarine when in contact to infinitely rigid surface described by (4). Numerical solutions of these sets can be obtained by simulation, however the numerical solution, using a DAE solver, can take too much effort to converge due to the fact that these sets of equation represent a highly stiff system. In order to minimize this numerical drawback,



**Fig. 3.** Contact force  $\lambda$ , force error  $\tilde{\lambda} = \lambda(t) - \lambda^d$  and control variables  $S_{qF}$  for the model-free second order sliding mode control; all for set-point disturbance-free case (*continuous line for model-free second order sliding mode control, dotted line for PD control*).



**Fig. 4.** Control variables  $S_{qp}$  for the model-free second order sliding mode control for set-point disturbance-free case.

the holonomic constraint has been treated as a numerically compliant surface which dynamic is represented by

$$\ddot{\varphi}(\mathbf{q}) + D\dot{\varphi}(\mathbf{q}) + P\varphi(\mathbf{q}) = 0. \quad (34)$$

This is known in the force control literature of robot manipulators as constrained stabilization method, which bounds the nonlinear numerical error of integration of the backward integration differentiation formula. With a appropriate choice of parameters  $P \gg 1$  and  $D \gg 1$ , the solution of  $\varphi(\mathbf{q}, t) \rightarrow 0$  is bounded. This dynamic is chosen to be fast enough to allow the numerical method to work properly. In this way, it is allowed very small deviation on the computation of  $\lambda$ , typically in the order of  $-10^6$  or less, which may produce, according to some experimental comparison, less than 0.001% numerical error. Then, the value of the normal contact force magnitude becomes:

$$\lambda = \left[ J_\varphi J_\nu^{-1} M_v^{-1} J_R^T J_{\varphi+}^T \right]^{-1} \left( J_\varphi J_\nu^{-1} M_v^{-1} (\mathbf{h}_v - \mathbf{u}) - \frac{d}{dt} (J_\varphi J_\nu^{-1}) \boldsymbol{\nu} - D J_\varphi J_\nu^{-1} \boldsymbol{\nu} - P \varphi(\mathbf{q}) \right), \quad (35)$$

$$= \left[ J_\varphi M_q^{-1} J_\varphi^T \right]^{-1} \left( J_\varphi M_q^{-1} (\mathbf{h}_q - \mathbf{u}_q) - \dot{J}_\varphi \dot{\mathbf{q}} - D J_\varphi \dot{\mathbf{q}} - P \varphi(\mathbf{q}) \right). \quad (36)$$

The numerical dynamic induced in the constraint surface were performed with  $P = 9x10^6$  and  $D = 36x10^3$ .



## 4.2 Disturbance-free

**Set-point Control.** A constant desired position/force task is developed, consisting on a change of deep, from 1m to 2m, and orientation from 5 degrees to -5, while the submarine remains in contact with a constant contact force of 100N.

Figures 1 and 2 shows respectively position and force inputs. The difference in stabilization velocity has been explicitly computed in order to visualize the qualitative differences. In any case, this velocity can be modified with appropriate tuning of gain parameters. Notice that there is some transient and variability in the position of the contact point in the the direction normal to that force (the x coordinate). The same transients are present in the force graphic 3 where a noticeable difference between simple PD and model-free second order sliding mode control is present. The big difference is that although PD can regulate with some practical relative accuracy it is not capable of track nor regulate force reference.

Figure 3 shows the contact force magnitude  $\lambda$ , the force error  $\tilde{\lambda} = \lambda(t) - \lambda^d$  and force manifold  $S_{qF}$  for the model-free second order sliding mode control. In this graphic is is clear that no force is induced with the PD scheme. In the case of the model-free 2nd order sliding mode, force regulation is achieved very fast indeed.

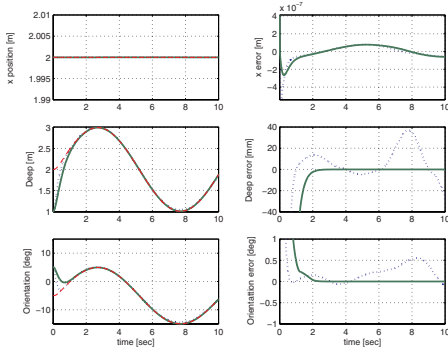
Figure 4 shows the convergence of the extended position manifolds  $S_{qp}$ . They do converge to zero and induce there after the sliding mode dynamics.

## 4.3 Disturbance of Fluid Dynamics

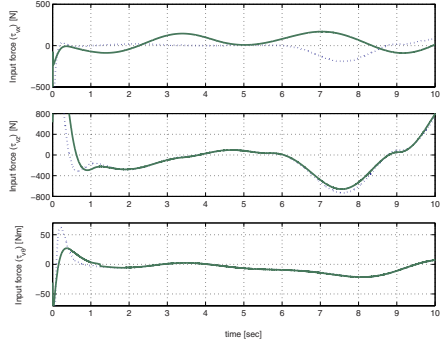
Disturbance forces where calculated using the function  $\eta_v(\nu, t)$  in (1), which is explicitly described in [6]. The velocity of the fluid were calculated considering a possible values of tides and current. So the horizontal fluid velocity (x component) is composed by a constant drift of 0.5 m/s (about 1 knot) and a periodic wave of 1 m/s amplitude (about 2 knots) over a period of 7 sec. The vertical fluid velocity (z component) is composed only by a periodic wave of 1 m/s amplitude (about 2 knots) over a period of 6 sec.

**Tracking Case.** For this case, the desired position/force desired trajectories are as follows: a variable deep, center at 2 meter with 1 meter amplitude (pick to pick) and a 10 sec period. The desired orientation trajectory is 10 degree amplitude with an offset of -5 degrees with period of 10 sec. And the contact desired force of 70N amplitude with offset of 100N, and a period of 4 sec.

Figures 5 and 6 shows respectively position and force inputs. The difference in stabilization velocity has been explicitly computed in order to visualize the qualitative differences. In any case, this velocity can be modified with appropriate tuning of gain parameters. Notice that there is some transient and variability in the position of the contact point in the the direction normal to that force (the x coordinate). The same transients are present in the force graphic 7 where a noticeable difference between simple PD and model-free second order sliding mode control is present. The big difference is that although PD can regulate with some practical relative accuracy it is not capable of track nor regulate force reference.



**Fig. 5.** Generalized coordinates  $q$  and errors  $\tilde{q} = q(t) - q^d \nu$  for tracking disturbance case (continuous line for model-free second order sliding mode control, dotted line for PD control).



**Fig. 6.** Inputs controlled forces  $u$ , in vehicle's frame, for tracking disturbance case (continuous line for model-free second order sliding mode control, dotted line for PD control).

Figure 7 shows the contact force magnitude  $\lambda$ , the force error  $\tilde{\lambda} = \lambda(t) - \lambda^d$  and force manifold  $S_{qF}$  for the model-free second order sliding mode control. In this graphic is clear that no force is induced with the PD scheme. In the case of the model-free 2nd order sliding mode, force regulation is achieved very fast indeed.

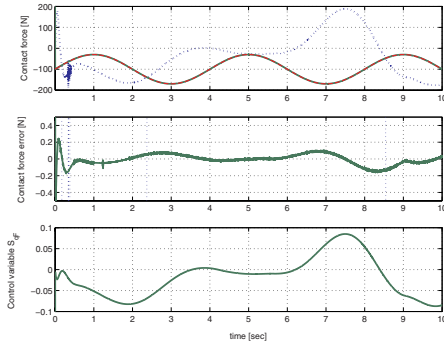
Figure 8 shows the convergence of the extended position manifolds  $S_{qp}$ . They do converge to zero and induce there after the sliding mode dynamics.

## 5 Some Discussions on Structural Properties

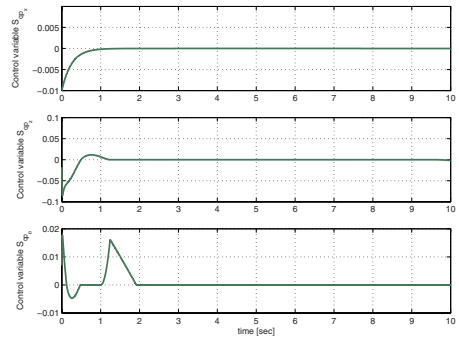
In the beginning we were expecting that a controller implemented for robot manipulator could be implemented also for submarine robots if some explicit additional terms were added, such as those control terms that compensate for hydrodynamic and buoyancy forces. However, surprisingly, no additional terms were required! It suffices only proper mapping of the gradient of contact forces. Some remarks are in order.

### 5.1 Properties of the Dynamics

As it as pointed out in section 2, the model of a submarine robot can be expressed in either a *self space* where the inertia matrix is constant for some conditions that in practice are not difficult to met or in a generalized coordinate space in which the inertial matrix is no longer constant but the model is expressed by only one equation likewise the kinematic lagrangian chains. Both representations has all known properties of lagrangian systems such skew symmetry for the Coriolis/Inertia matrix, boundness of all the components and passivity preserved properties for he hydrodynamics added effects (including buoyancy). The equivalences of these representation of the same model can be found by means of the kinematic equation. The last is a linear operator that maps



**Fig. 7.** Contact force  $\lambda$ , force error  $\tilde{\lambda} = \lambda(t) - \lambda^d$  and control variables  $S_{qF}$  for the model-free second order sliding mode control, all for tracking disturbance case (*continuous line for model-free second order sliding mode control, dotted line for PD control*).



**Fig. 8.** Control variables  $S_{qp}$  for the model-free second order sliding mode control for tracking disturbance case.

generalized coordinates time derivative with a generalized physical velocity. This relationship is specially important for the angular velocity of a free moving object due to the fact that time derivative of angular representations (such a roll-pitch-yaw) is not the angular velocity. However there is always a correspondence between these vectors. For external forces this mapping is indeed important. It relates a physical force/torque wrench to the generalized coordinates  $\mathbf{q} = (x_v, y_v, z_v, \phi_v, \theta_v, \psi_v)$  whose last 3 components does not represent a unique physical space. In this work such mapping is given by  $J_q$  and appears in the contact force mapping by the normalized operator  $J_{\tilde{\varphi}}$ .

### 5.2 The Controller

Notice that the controller exhibits a PD structure plus a nonlinear  $I$ -tame control action, with nonlinear time-varying feedback gains, for each orthogonal subspace. It is in fact a decentralized PID-like, coined as "Sliding PD" controller. It is indeed surprising that similar control structures can be implemented seemingly for a robot in the surface or below water, with similar stability properties, simultaneous tracking of contact force and posture. Of course, this is possible under a proper mapping of jacobians, key to implement this controller.

### 5.3 Friction at the Contact Point

When friction at the contact point arises, which is expected for submarine tasks wherein the contact object is expected to exhibit a rough surface, with high friction coefficients, a tangential friction model should be added in the right hand side. Particular care must be employed to map the velocities. Complex friction compensators can be designed, similar to the case of force control for robot manipulators, therefore details are omitted.

## 6 Conclusions

Although a PD controller, recently validated by Whitcomb, has a much simpler structure and very good performance for underwater vehicles in position control schemes, it does not deal with the task of contact force control.

Structural properties of the open-loop dynamics of submarine robots are key to design passivity-based controllers. In this paper, an advanced, yet simple, controller is proposed to achieve simultaneously tracking of time-varying contact force and posture, without any knowledge of its dynamic parameters. This is a significant feature of this controller, since in particular for submarine robots the dynamics are very difficult to compute, let alone its parameters. A simulation study provides additional insight into the closed-loop dynamic properties for regulation and tracking case.

## References

1. E. Olguín Díaz, V. Parra-Vega *On the Force/Posture Control of a Constrained Submarine Robot* 4th International Conference on Informatics in Control, Robotics and Automation, Conference Proceedings, May 2007
2. Smallwood, D.A.; Whitcomb, L.L. *Model-based dynamic positioning of underwater robotic vehicles: theory and experiment* Oceanic Engineering, IEEE Journal of Volume: 29 Issue: 1 Jan. 2004
3. V. Parra-Vega, *Second Order Sliding Mode Control for Robot Arms with Time Base Generators for Finite-time Tracking*, Dynamics and Control, 2001.
4. Smallwood, D.A.; Whitcomb, L.L. *Toward model based dynamic positioning of underwater robotic vehicles* OCEANS, 2001. MTS/IEEE Conference and Exhibition Volume: 2 2001
5. Villani, L.; Natale, C.; Siciliano, B.; Canudas de Wit, C.; *An experimental study of adaptive force/position control algorithms for an industrial robot* Control Systems Technology, IEEE Transactions on Volume 8, Issue 5, Sept. 2000
6. E. Olguín Díaz, *Modélisation et Commande d'un Système Véhicule/Manipulateur Sous-Marin* PhD Thesis. Laboratoire d'Automatique de Grenoble, January 1999
7. Y. H. Liu, S. Arimoto, V. Parra-Vega, and K. Kitagaki, *Decentralized Adaptive Control Of Multiple Manipulators in Cooperations*, International Journal of Control, Vol. 67, No. 5, pp. 649-673, 1997.
8. M. Perrier, C. Canudas de Wit, "Experimental Comparison of PID versus PID plus Nonlinear Controller for Subsea Robots" *Journal of Autonomous Robots, Special issue on Autonomous Underwater Robots, 1996*.
9. V. Parra-Vega and S. Arimoto, *A Passivity-based Adaptive Sliding Mode Position-Force Control for Robot Manipulators*, International Journal of Adaptive Control and Signal Processing, Vol. 10, pp. 365-377, 1996.
10. Arimoto, S. *Fundamental problems of robot control* Robotica (1995), volume 13, pp 19-27, 111-122, Cambridge University Press
11. Thor I. Fossen. *Guidance and Control of Ocean Vehicles*. John Wiley and Sons, Chichester 1994
12. I. Schjøberg, T. I. Fossen. "Modelling and Control of Underwater Vehicle-Manipulator Systems" *Proceedings the 3<sup>rd</sup> Conference on Marine Craft Maneuvering and Control (MCMC'94)*, Southampton, UK, 1994.
13. Chiaverini, S.; Sciavicco, L. *The parallel approach to force/position control of robotic manipulators*. IEEE Transactions on Robotics and Automation Volume 9, Issue 4, Aug. 1993

14. IFREMER, *Project VORTEX: Modélisation et simulation du comportement hydrodynamique d'un véhicule sous-marin asservi en position et vitesse*. 1992
15. Sagatun, S.I.; Fossen, T.I.; *Lagrangian formulation of underwater vehicles' dynamics* Decision Aiding for Complex Systems, Conference Proceedings., 1991 IEEE International Conference.
16. Spong M.W., Vidyasagar M. *Robot dynamics and control*. John Wiley, New York 1989
17. Yoerger, D.; Slotine, J. *Robust trajectory control of underwater vehicles* Oceanic Engineering, IEEE Journal of Volume: 10 Issue: 4 Oct 1985

# **PART III**

## **Signal Processing, Systems Modeling and Control**

# Modelling Robot Dynamics with Masses and Pulleys

Leo J. Stocco and Matt J. Yedlin

The Department of Electrical and Computer Engineering  
The University of British Columbia, 2332 Main Mall, Vancouver, V6T 1Z4, BC, Canada  
{leos, matty}@ece.ubc.ca

**Abstract.** The well-known electro-mechanical analogy that equates current, voltage, resistance, inductance and capacitance to force, velocity, damping, stiffness and mass has a shortcoming in that mass can only be used to simulate a capacitor which has one terminal connected to ground. A new model that was previously proposed by the authors that combines a mass with a pulley (MP) is shown to simulate a capacitor in the general case. This new MP model is used to model the off-diagonal elements of a mass matrix so that devices whose effective mass is coupled between more than one actuator can be represented by a mechanical system diagram that is topographically parallel to its equivalent electric circuit model. Specific examples of this technique are presented to demonstrate how a mechanical model can be derived for both a serial and a parallel robot with both two and three degrees of freedom. The technique, however, is extensible to any number of degrees of freedom.

**Keywords.** Mass matrix, inertia matrix, MP model, pulley, differential transmission, mechanical system representation, robot dynamics, impedance, equivalent electric circuit.

## 1 Introduction

The concept of impedance and its generalization reactance, has been used to define equivalent circuits of mechanical and electro-mechanical systems since the development of the Maxwell model of solids. The idea that driving point impedances could be decomposed into terms that parallel electrical elements was initiated by [5] who showed that the frequency response of any system is determined by the poles and zeros of its transfer function. The conditions for network synthesis are described by [1] and later applied by [8] who introduced bond graphs to distinguish and represent effort and flow variables in a graphical setting. Examples of electro-mechanical system simulations are numerous and include magnetic circuits [6], mechatronics and electromechanical transducers [11], [12], [9].

Mechanical block diagrams are routinely used to model robot dynamics although some [3] limit them to a single axis while others [13] rely entirely on equivalent electric circuits to avoid the inherent difficulties of creating mechanical models of multi-axis devices, transmission systems or other systems with coupled dynamics.

Section 2 of this paper describes the conventional electro-mechanical analogy and points out a limitation of the mass model. It goes on to describe a new mass/pulley (MP) model which overcomes the inherent deficiency in the conventional

mass model. In Section 3, it is shown how the new MP model can be used to model the dynamics of devices which have coupled effective masses. Examples are provided which include both 2-DOF and 3-DOF serial and parallel manipulators. Lastly, concluding remarks are made in Section 4.

## 2 Electro-Mechanical Analogies

The ability to define an electro-mechanical equivalent circuit stems from the parallelism in the differential equations that describe electrical and mechanical systems, each of which involve an across variable, a through variable and an impedance or admittance variable. In electrical circuits, voltage  $E(s)$  is the across variable and current  $I(s)$  is the through variable. In mechanical systems, velocity  $V(s)$  is the across variable and force  $F(s)$  is the through variable (i.e. flow variable[4]). This results in a correspondence between resistance  $R$  and damping  $B$ , inductance  $L$  and spring constant  $K$ , and capacitance  $C$  and mass  $M$  shown in (1-3). An alternate approach treats force as the across variable and velocity as the through variable but that approach is not used here. By (1-3), the electro-mechanical equivalents shown in Figure 1 can be substituted for one another to model a mechanical system as an electrical circuit and vice versa.

$$E(s) = I(s)R = I(s)\frac{1}{G} \quad V(s) = F(s)\frac{1}{B} \quad (1)$$

$$E(s) = I(s)sL \quad V(s) = F(s)\frac{s}{K} \quad (2)$$

$$E(s) = I(s)\frac{1}{sC} \quad V(s) = F(s)\frac{1}{sM} \quad (3)$$

### 2.1 Classical Mass Model Limitation

Each of the components in Figure 1 has two terminals except for the mass which has only one. This is due to the fact that the dynamic equation of a mass (3) does not accommodate an arbitrary reference. Acceleration is always taken with respect to the global reference, or ground. Consider the two systems in Figure 2 which are well known to be analogous.

In Figure 2, the voltage across the capacitor  $e_c$  corresponds to the velocity of the mass  $v_m$ . Both of these are relative measurements that only correspond to one another because both are taken with respect to ground. Consider, on the other hand, the circuit in Figure 3 which contains a capacitor with one terminal open circuited.

In Figure 3, the open circuit at  $n_2$  prevents any current from flowing through the capacitor. Since there is no current shunted into the capacitor at  $n_1$ , the voltage at  $n_1$  is unaffected by the capacitor. In the mechanical "equivalent", it is not possible to connect a non-zero mass  $M$  to node  $n_1$  without affecting the output velocity  $v_o$ . This is due to the implicit ground reference of the mass (shown by a dotted line) which is physically impossible to interrupt. Note that this same limitation does not apply to the



spring or damper since they both have two terminals which can be connected or left floating, as desired.






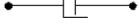




Current Source: $I(s)$ 	Force Source: $F(s)$ 	$F = I$
Voltage Source: $E(s)$ 	Velocity Source: $V(s)$ 	$V = E$
Resistor: $G$ 	Damper: $B$ 	$B = G$
Inductor: $1/sL$ 	Spring: $K/s$ 	$K = 1/L$
Capacitor: $sC$ 	Mass: $sM$ 	$M = C$

Fig. 1. Admittance of electro-mechanical equivalents.

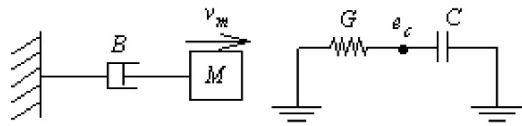


Fig.2. LC circuit and mechanical equivalent.

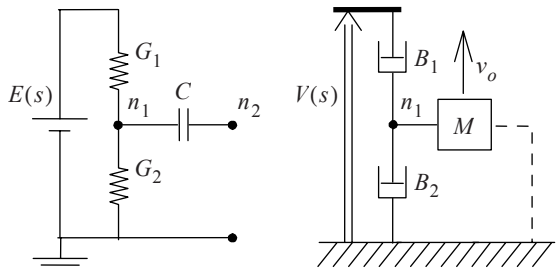


Fig. 3. RC circuit and mechanical equivalent.

## 2.2 The Mass/Pulley (MP) Model

Because of the above limitation, there are mechanical systems which can not be modelled using a mechanical system diagram. Elaborate transmission systems such as robotic manipulators may contain mass elements that are only present when relative motion occurs between individual motion stages. Currently, systems such as these can only be modelled using electric circuits since capacitors can be used to model this type of behaviour but masses cannot.

It would be useful to have a mechanical model which simulates the behaviour of a capacitor without an implicit ground connection so that any mechanism (or electric circuit) could be modelled by a mechanical system diagram. This new model should have two symmetric terminals (i.e. flipping the device over should not affect its response), obey Ohm’s Law, and be able to accommodate non-zero velocities at both terminals simultaneously. A model proposed by the authors [10] combines a mass with the pulley-based differential transmission shown in Figure 4. The pulley system obeys the differential position / velocity relationship shown in (4, 5).

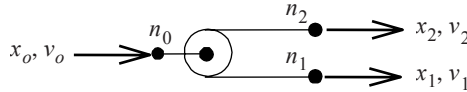


Fig. 4. Pulley based differential transmission.

$$\Delta x_o = \frac{1}{2}(\Delta x_1 + \Delta x_2) \tag{4}$$

$$v_o = \frac{1}{2}(v_1 + v_2) \tag{5}$$

Note from (5) that although the pulley provides the desired differential velocity input, it also introduces an undesired 2:1 reduction ratio. However, setting  $v_1$  to 0 (i.e. connecting  $n_1$  to ground) results in (6). Therefore, a similar pulley system with one input tied to ground could be used to scale up velocity by an equivalent ratio.

$$v_2 = 2v_o \tag{6}$$

The double pulley system shown in Figure 5 is a differential transmission with a unity gear ratio. The primary pulley provides the differential input while the secondary pulley cancels the reduction ratio to achieve unity gain. A mass connected to the secondary pulley is accelerated by a rate equal to the difference between the acceleration of the two inputs,  $n_1$  and  $n_2$ . This system simulates the behaviour of a capacitor that may or may not be connected to ground (Figure 5). Voltage  $E_1$  corresponds to velocity  $V_1$ , voltage  $E_2$  corresponds to velocity  $V_2$ , current  $I$  corresponds to tension  $F$  and capacitance  $C$  corresponds to mass  $M$  as shown by (7, 8). Note that the free-body diagram of the centre pulley shows that the tension  $F$  in the primary cable is equal to the tension  $F$  in the secondary cable. The system must be balanced because any net force on the massless centre pulley would result in infinite acceleration of the pulley and therefore, the mass as well.

$$E_2(s) - E_1(s) = I(s) \frac{1}{sC} \tag{7}$$

$$V_2(s) - V_1(s) = F(s) \frac{1}{sM} \tag{8}$$

The MP model uses ideal cables with zero mass and infinite length and stiffness. The ideal cables travel through the system of massless, frictionless pulleys without any loss of energy. The MP model operates in zero gravity so the mass is only accelerated as a result of cable tension and/or compression. Unlike practical cables, the ideal cables never become slack. When an attractive force is applied between  $n_1$  and  $n_2$ ,  $F < 0$  and the mass is accelerated downward. A block diagram of the MP model

is presented in Figure 6 where  $P$  has the same value as  $M$  in Figure 5. Note that, unlike a pure mass, the MP model has two terminals,  $n_1$  and  $n_2$  which correspond to the two ends of the primary cable.

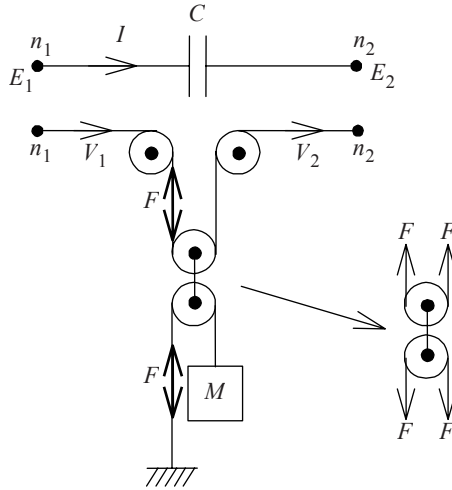


Fig. 5. Mass/ pulley equivalent of a capacitor.

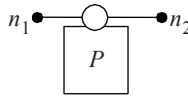


Fig. 6. Block diagram of MP model.

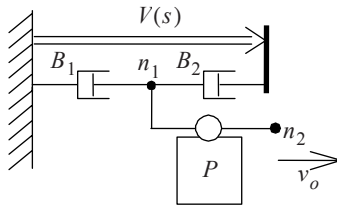


Fig. 7. Mechanical equivalent circuit using MP model.

Consider Figure 7 which is the mechanical system from Figure 3 with the mass replaced by an MP model. With terminal  $n_2$  left unconnected, the primary cable of the MP model travels freely through the primary pulley without accelerating the mass or consuming energy. The MP model behaves just like the capacitor in Figure 3. Also note the topological similarity between the electrical circuit in Figure 3 and its true mechanical equivalent in Figure 7. This is a direct result of the topological consistency between the capacitor and the MP model, both of which have two symmetric terminals. As pointed out in [10], this consistency allows one to analyze mechanical systems using electric circuit analysis techniques once all masses have been replaced by MP models.

### 3 Robot Mass Matrix

Consider the simplified dynamics of a 2-DOF robot (9) where  $M$  is the mass matrix,  $B$  is the damping matrix,  $F$  is a vector of joint forces/torques (10),  $R$  is a vector of joint rates  $r_1$  and  $r_2$  (10), and  $s$  is the Laplace operator. Spring constants, gravitational and coriolis effects are assumed to be negligible for the purpose of this example. If the damping in the system is dominated by the actuator damping coefficients,  $B$  is a diagonal matrix (10).  $M$ , on the other hand, represents the effective mass perceived by each joint and is not diagonal or otherwise easily simplified in general.

$$F = BR + MsR \tag{9}$$

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} b_1 & 0 \\ 0 & b_2 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} + Ms \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \tag{10}$$

For simple kinematic arrangements such as the redundant actuators shown in Figure 8 which only have a single axis of motion,  $M$  is shown in (11). The system responses are modeled by the mechanical system diagram shown in Figure 9 and the dynamic equation shown in (10). Using the electro-mechanical transformation described in Section 2, this system can also be represented by the electrical circuit analogy shown in Figure 9.

$$M = \begin{bmatrix} m_1 & m_2 \\ m_2 & m_2 \end{bmatrix} \tag{11}$$

Performing nodal analysis on the circuit in Figure 9 results in (12) by inspection. Note however, that (12) contains the term  $i_1-i_2$  as well as  $v_2$  which corresponds to the end-point velocity in the mechanical system or, in other words, the sum of the joint rates  $r_1+r_2$ . To obtain a correspondence between electrical and mechanical component values, the dynamic equation (10) is rearranged in (13) where the associated damping  $B'$  and mass  $M'$  matrices are shown in (14, 15). From (14), the resistor admittances  $g_1$  and  $g_2$  and capacitor values  $c_1$  and  $c_2$  correspond to the equivalent damping and mass values  $b'_1, b'_2, m'_1$  and  $m'_2$  (16) respectively.

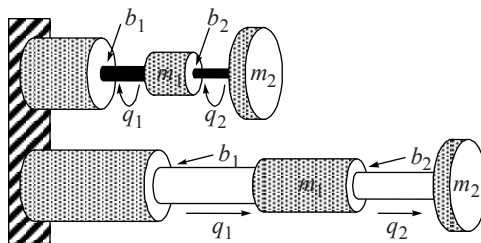
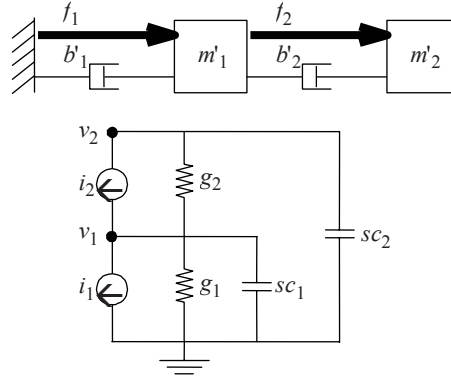


Fig. 8. Redundant rotary & prismatic actuators.



**Fig. 9.**System models of redundant actuators.

$$\begin{bmatrix} i_1 - i_2 \\ i_2 \end{bmatrix} = \begin{bmatrix} g_1 + g_2 & -g_2 \\ -g_2 & g_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} c_1 & 0 \\ 0 & c_2 \end{bmatrix} s \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (12)$$

$$\begin{bmatrix} f_1 - f_2 \\ f_2 \end{bmatrix} = B' \begin{bmatrix} r_1 \\ r_1 + r_2 \end{bmatrix} + M' s \begin{bmatrix} r_1 \\ r_1 + r_2 \end{bmatrix} \quad (13)$$

$$B' = \begin{bmatrix} b'_1 + b'_2 & -b'_2 \\ -b'_2 & b'_2 \end{bmatrix} = \begin{bmatrix} b_1 + b_2 & -b_2 \\ -b_2 & b_2 \end{bmatrix} \quad (14)$$

$$M' = \begin{bmatrix} m'_1 & 0 \\ 0 & m'_2 \end{bmatrix} = \begin{bmatrix} m_1 + m_2 & 0 \\ 0 & m_2 \end{bmatrix} \quad (15)$$

$$\begin{bmatrix} b'_1 \\ b'_2 \\ m'_1 \\ m'_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ m_1 + m_2 \\ m_2 \end{bmatrix} \quad (16)$$

In this simple example, masses are sufficient to model the system behaviour but only because the device has a single degree of freedom so  $M'$  is diagonal and there is no cross-coupling between actuators. In general, however, effective mass is not always decoupled and the off-diagonal elements of  $M'$  can be expected to be non-zero. When  $M'$  is not diagonal, conventional single-terminal masses are unable to model the entire effective mass of the system. They can not model the off-diagonal terms that describe inertial effects resulting from relative motion of the actuators.

### 3.1 Serial 2-DOF Robot

Consider the 2-DOF serial robot shown in Figure 10. The mass matrix for this mechanism is approximated in [2] by two point masses  $d_1$  and  $d_2$  placed at the distal actuator and end-effector as indicated below. The resulting mass matrix (17) has the terms shown in (18-20) where  $q_1$  and  $q_2$  are the joint angles and  $l_1$  and  $l_2$  are the link lengths. Just as in the previous example, actuator damping coefficients  $b_1$  and  $b_2$  are taken to dominate the total system damping.

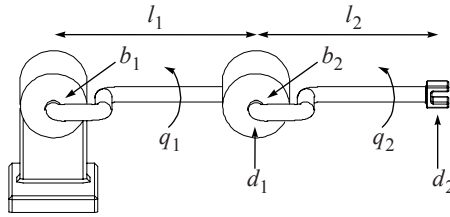


Fig. 10. 2-DOF serial robot.

$$M(q) = \begin{bmatrix} m_1(q) & m_3(q) \\ m_3(q) & m_2(q) \end{bmatrix} \tag{17}$$

$$m_1 = l_2^2 d_2 + 2l_1 l_2 d_2 \cos(q_2) + l_1^2 (d_1 + d_2) \tag{18}$$

$$m_2 = l_2^2 d_2 \tag{19}$$

$$m_3 = l_2^2 d_2 + l_1 l_2 d_2 \cos(q_2) \tag{20}$$

The equivalent circuit model of this system is shown in Figure 11. It is similar to Figure 9 except that the capacitor values are configuration dependent and a third capacitor  $c_{12}$  is included to model the coupled mass terms that are present. Performing nodal analysis results in (21) and the corresponding  $M'$  matrix in (22) which can be rearranged to solve for the mechanical model parameters in terms of the physical mass values in (23).  $B'$  is the same diagonal matrix as in (14).

Note from (22) that  $M'$  is diagonal (i.e.  $p'_{12}=0$ ) when  $m_2=m_3$ . From (19,20), this is merely the special case when  $q_2=\pm\pi/2$ . Therefore, it is not possible to model this system using only masses due to their implicit ground reference, as described in Section 2.1. The off-diagonal terms can, however, be modelled using the MP model proposed in Section 2.2. It results in a mechanical system model that is topologically identical to the equivalent circuit in Figure 11 where each grounded capacitor ( $c_1, c_2$ ) is replaced by a regular mass and each ungrounded capacitor ( $c_{12}$ ) is replaced by an MP model since the MP model is able to accommodate a non-zero reference acceleration. The resulting mechanical system is shown in Figure 12.

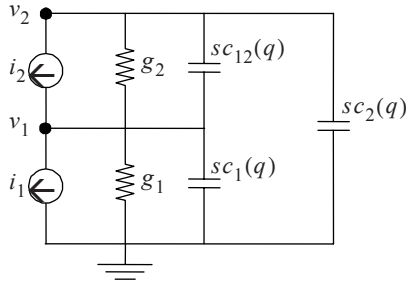


Fig. 11. Electrical model of 2-DOF serial robot.

$$\begin{bmatrix} i_1 - i_2 \\ i_2 \end{bmatrix} = \begin{bmatrix} g_1 + g_2 & -g_2 \\ -g_2 & g_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} c_1 + c_{12} & -c_{12} \\ -c_{12} & c_2 + c_{12} \end{bmatrix} s \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (21)$$

$$M'(q) = \begin{bmatrix} m'_1 + p'_{12} & -p'_{12} \\ -p'_{12} & m'_2 + p'_{12} \end{bmatrix} = \begin{bmatrix} m_1 + m_2 & m_3 - m_2 \\ m_3 - m_2 & m_2 \end{bmatrix} \quad (22)$$

$$\begin{bmatrix} m'_1 \\ m'_2 \\ p'_{12} \end{bmatrix} = \begin{bmatrix} m_1 + m_3 \\ m_3 \\ m_2 - m_3 \end{bmatrix} \quad (23)$$

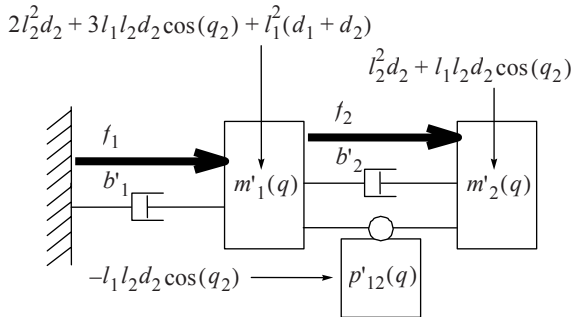


Fig. 12. Mechanical model of 2-DOF serial robot.

Although  $p'_{12}$  has a negative value when  $-\pi/2 < q_2 < \pi/2$ , the net mass perceived by each actuator is always positive because  $M$  is positive definite. When  $p'_{12}$  is negative, it simply means that the motion of actuator 1 reduces the net mass perceived by actuator 2, but the net mass perceived by actuator 2 is always greater than zero.

### 3.2 Parallel 2-DOF Robot

The same technique can be applied to parallel manipulators such as the 2-DOF 5-bar linkage used by (Hayward et al., 1994). In the case of parallel manipulators, each

actuator is referenced to ground but there remains a coupling between the effective mass perceived by each actuator which, like a serial manipulator, is configuration dependent. This coupling is modelled by  $c_{12}$  and  $p'_{12}$  in the equivalent electrical and mechanical models shown in Figure 13. Typically, parallel manipulators also have coupled damping terms due to their passive joints which would be modelled by a conductance  $g_{12}$  added between nodes 1 and 2 (i.e. in parallel with  $c_{12}$ ). However, for the sake of simplicity, the damping of the passive joints are neglected here.

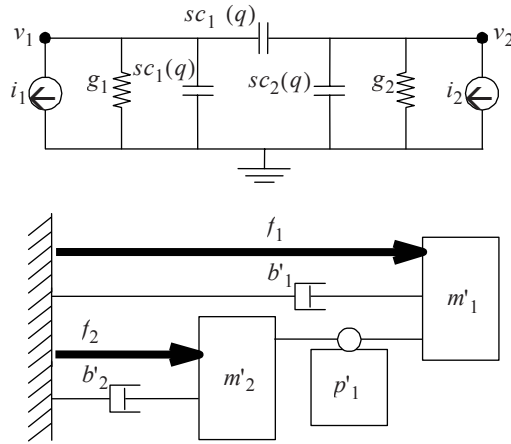


Fig. 13. Model of a 2-DOF parallel robot.

Performing nodal analysis on the circuit in Figure 13 results in (24) by inspection. For a parallel robot, currents and voltages correspond directly to joint forces and joint rates so  $B'=B$  and  $M'=M$ . For a mass matrix of the form shown in (17), the elements of the  $M'$  matrix, and therefore the parameter values associated with the masses and MP models of Figure 13, are shown in (26).

$$\begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} g_1 & 0 \\ 0 & g_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} c_1 + c_{12} & -c_{12} \\ -c_{12} & c_2 + c_{12} \end{bmatrix} s \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{24}$$

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} b'_1 & 0 \\ 0 & b'_2 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} + \begin{bmatrix} m'_1 + p'_{12} & -p'_{12} \\ -p'_{12} & m'_2 + p'_{12} \end{bmatrix} s \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \tag{25}$$

$$\begin{bmatrix} m'_1 \\ m'_2 \\ p'_{12} \end{bmatrix} = \begin{bmatrix} m_1 + m_3 \\ m_2 + m_3 \\ -m_3 \end{bmatrix} \tag{26}$$

### 3.3 Multiple Degree of Freedom Robots

This technique is easily extended to devices with any number  $n$  of degrees of freedom. With serial manipulators, the compliance and damping is often mainly in the



actuators and the damping  $B$  and spring  $K$  matrices are diagonal (27,28). With parallel manipulators, the  $B$  and  $K$  matrices typically contain off-diagonal terms but they are easily modelled using conventional techniques since springs and dampers are 2-terminal devices which can be placed at any two nodes in a system diagram.

$$B = \text{diag}\left(\left[b_1 \ b_2 \ \dots \ b_n\right]\right) \tag{27}$$

$$K = \text{diag}\left(\left[1/k_1 \ 1/k_2 \ \dots \ 1/k_n\right]\right) \tag{28}$$

To account for inertial cross-coupling, the model must contain a capacitor and/or MP model between every pair of actuators. For example, the electric circuit model and corresponding mechanical system model of a serial 3-DOF manipulator are shown in Figure 14. The capacitance  $C$  matrix resulting from the nodal analysis (29) of the circuit in Figure 14 is shown in (30).

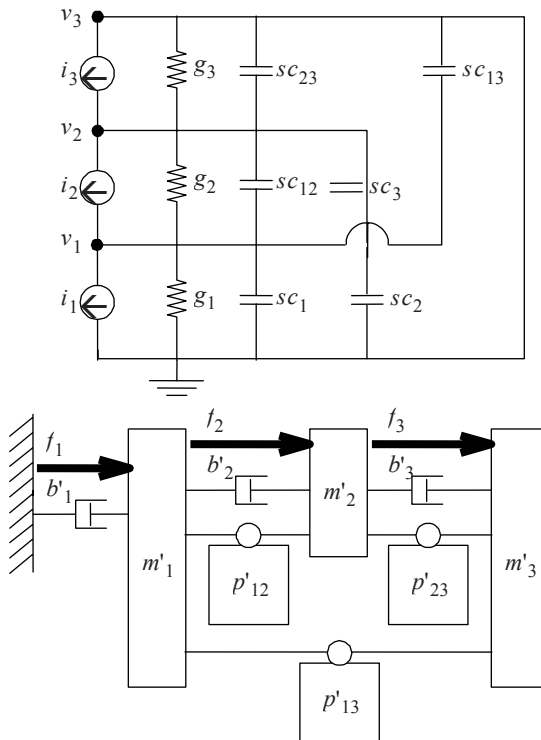


Fig. 14. Model of a 2-DOF parallel robot.

$$\begin{bmatrix} i_1 - i_2 \\ i_2 - i_3 \\ i_3 \end{bmatrix} = G(q) \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} + C(q)s \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (29)$$

$$C(q) = \begin{bmatrix} c_1 + c_{12} + c_{13} & -c_{12} & -c_{13} \\ -c_{12} & c_2 + c_{12} + c_{23} & -c_{23} \\ -c_{13} & -c_{23} & c_3 + c_{23} + c_{13} \end{bmatrix} \quad (30)$$

Just as in previous examples, the 3x3 mass matrix  $M'$  (32) is rearranged into the form shown in (31) to parallel the current/voltage relationship of (29). For a mass matrix  $M$  of the form shown in (33), the entries of the  $M'$  matrix are solved for in (34). Similarly, for a parallel 3-DOF robot, the electric circuit model and corresponding mechanical system model are shown in Figure 15. For a mass matrix of the form shown in (33), the elements of  $M'$  are shown in (35).

$$\begin{bmatrix} f_1 - f_2 \\ f_2 - f_3 \\ f_3 \end{bmatrix} = B' \begin{bmatrix} r_1 \\ r_1 + r_2 \\ r_1 + r_2 + r_3 \end{bmatrix} + M's \begin{bmatrix} r_1 \\ r_1 + r_2 \\ r_1 + r_2 + r_3 \end{bmatrix} \quad (31)$$

$$M'(q) = \begin{bmatrix} m'_1 + p'_{12} + p'_{13} & -p'_{12} & -p'_{13} \\ -p'_{12} & m'_2 + p'_{12} + p'_{23} & -p'_{23} \\ -p'_{13} & -p'_{23} & m'_3 + p'_{13} + p'_{23} \end{bmatrix} \quad (32)$$

$$M(q) = \begin{bmatrix} m_1(q) & m_4(q) & m_5(q) \\ m_4(q) & m_2(q) & m_6(q) \\ m_5(q) & m_6(q) & m_3(q) \end{bmatrix} \quad (33)$$

$$\begin{bmatrix} m'_1 \\ m'_2 \\ m'_3 \\ p'_{12} \\ p'_{23} \\ p'_{13} \end{bmatrix} = \begin{bmatrix} m_1 - m_4 \\ m_4 - m_5 \\ m_5 \\ m_2 + m_5 - m_4 - m_6 \\ m_3 - m_6 \\ m_6 - m_5 \end{bmatrix} \quad (34)$$

## 4 Conclusions

It is argued that a plain mass is not a complete and general model of a capacitor since a mass only has one terminal whereas a capacitor has two. The response of a mass corresponds to its acceleration with respect to ground and, therefore, can only be used to simulate a capacitor which has one terminal connected to ground. It cannot be used

to simulate a capacitor which has a non-zero reference voltage. A new model described here that consists of a mass and a pulley correctly simulates the response of a capacitor in the general case.

$$\begin{bmatrix} m'_1 \\ m'_2 \\ m'_3 \\ p'_{12} \\ p'_{23} \\ p'_{13} \end{bmatrix} = \begin{bmatrix} m_1 + m_4 + m_6 \\ m_2 + m_4 + m_6 \\ m_3 + m_5 + m_6 \\ -m_4 \\ -m_5 \\ -m_6 \end{bmatrix} \tag{35}$$

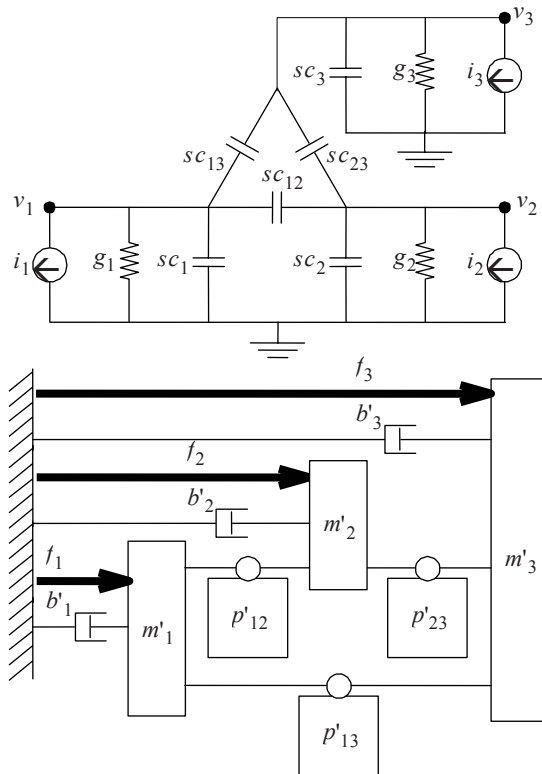


Fig. 15. Model of a 3-DOF parallel robot.

It is shown that the MP model can be used to model systems with cross-coupled effective masses which are otherwise, impossible to model with pure masses alone. This includes both serial and parallel manipulators with any number of degrees of freedom. The mechanical system model that is obtained fully describes the dynamic response of the system and is topologically identical to its electric circuit equivalent. As shown in (Stocco & Yedlin, 2007), this makes it possible to apply electric circuit analysis techniques to mechanical systems, directly.

**Acknowledgements.** The authors gratefully acknowledge Tim Salcudean for his valuable comments during the preparation of this manuscript.

## References

1. Brune, O., 1931. "Synthesis of a finite two-terminal network whose driving-point impedance is a prescribed function of frequency". *J. Math. Physics*, vol. 10, pp. 191-236.
2. Craig, J.J., 2005. "Introduction to Robotics Mechanics and Control". 3rd ed., *Pearson Prentice Hall*.
3. Eppinger, S., Seering, W., 1992. "Three Dynamic Problems in Robot Force Control". *IEEE Trans. Robotics & Auto.*, V. 8, No. 6, pp. 751-758.
4. Fairlie-Clarke, A.C., 1999. "Force as a Flow Variable". *Proc. Instn. Mech. Engrs.*, V. 213, Part I, pp. 77-81.
5. Foster, R. M., 1924. "A reactance theorem". *Bell System Tech. J.*, vol. 3, pp. 259-267.
6. Hamill, D.C., 1993. "Lumped Equivalent Circuits of Magnetic Components: The Gyrator-Capacitor Approach". *IEEE Transactions on Power Electronics*, vol. 8, pp. 97.
7. Hayward, V., Choksi, J., Lanvin, G., Ramstein, C., 1994. "Design and Multi-Objective Optimization of a Linkage for a Haptic Interface". *Proc. of ARK '94, 4th Int. Workshop on Advances in Robot Kinematics* (Ljubiana, Slovenia), pp. 352-359.
8. Paynter, H.M., 1961. *Analysis and Design of Engineering Systems*. MIT Press.
9. Sass, L., McPhee, J., Schmitke, C., Fisette, P., Grenier, D., 2004. "A Comparison of Different Methods for Modelling Electromechanical Multibody Systems". *Multibody System Dynamics*, vol. 12, pp. 209-250.
10. Stocco, L., Yedlin, M., Sept. 2007. "Toward Relative Mass with a Pulley-Based Differential Transmission". *Submitted to: IEEE Trans. Robotics*.
11. Tilmans, H.A.C., 1996. "Equivalent circuit representation of electromechanical transducers: I. Lumped-parameter systems". *J. Micromech. Microeng.*, vol. 6, pp. 157-176.
12. van Amerongen, J., Breedveld, P., 2003. "Modelling of physical systems for the design and control of mechatronic systems". *Annual Reviews in Control*, vol. 27, pp. 87-117.
13. Yamakita, M., Shibasaki, H., Furuta, K., 1992. "Tele-Virtual Reality of Dynamic Mechanical Model". *Proc. IEEE/RSJ Int. Conf. Intelligent Robots & Systems*, (Raleigh, NC), pp. 1103-1110.

# Stochastic Nonlinear Model Predictive Control based on Gaussian Mixture Approximations

Florian Weissel, Marco F. Huber and Uwe D. Hanebeck

Intelligent Sensor-Actuator-Systems Laboratory  
Institute of Computer Science and Engineering, Universität Karlsruhe (TH), Germany  
{weissel, marco.huber, uwe.hanebeck}@ieee.org

**Abstract.** In this paper, a framework for stochastic Nonlinear Model Predictive Control (NMPC) that explicitly incorporates the noise influence on systems with continuous state spaces is introduced. By the incorporation of noise, which results from uncertainties during model identification and measurement, the quality of control can be significantly increased. Since stochastic NMPC requires the prediction of system states over a certain horizon, an efficient state prediction technique for nonlinear noise-affected systems is required. This is achieved by using transition densities approximated by axis-aligned Gaussian mixtures together with methods to reduce the computational burden. A versatile cost function representation also employing Gaussian mixtures provides an increased freedom of modeling. Combining the prediction technique with this value function representation allows closed-form calculation of the necessary optimization problems arising from stochastic NMPC. The capabilities of the framework and especially the benefits that can be gained by considering the noise in the controller are illustrated by the example of a mobile robot following a given path.

## 1 Introduction

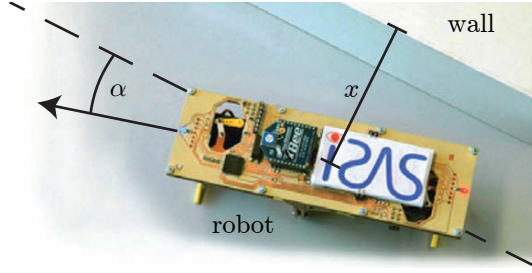
Model Predictive Control (MPC), which is also referred to as Receding or Rolling Horizon Control, has become more and more important for control applications from various fields. This is due to the fact that not only the current system state, but also a model-based prediction of future system states over a finite  $N$  stage prediction horizon is considered in the control law. For this prediction horizon, an open-loop optimal control problem with a corresponding cost function is solved. The resulting control input is then applied in an open-loop feedback fashion to the system.

The well understood and widely used MPC for linear system models [1] together with linear or quadratic cost functions is not always sufficient when it is necessary to achieve even higher quality control, e.g., in high precision robot control or in the process industry. Steadily growing requirements on the control quality can be met by incorporating nonlinear system models and cost functions in the control. The typically significant increase in computational demand arising from the nonlinearities has been mitigated in the last years by the steadily increasing available computation power for control processes [2] and advances in the employed algorithms to solve the necessary optimizations [3].

Nevertheless, in most approaches, especially for the important case of continuous state spaces, the influence of noise on the system is not considered [4], which obviously leads to unsatisfactory solutions especially for highly nonlinear systems and cost functions. An approach considering uncertainty in time-continuous systems with linear control inputs is presented in [5]. In case of a cost function that depends quadratically on the control input and special assumptions on the structure of the noise process, the optimal open-loop control problem can be interpreted as a path integral, which can be solved approximately with Monte-Carlo methods. In [6], the stochastic system behavior is considered in the control of a discrete-time system, but the control is only determined in the vicinity of the deterministic solution, which just leads to a locally optimal solution. In [7], an approach for infinite horizon optimal control is presented, where a continuous state space is discretized by means of a radial-basis-function network. This approach leads to a consideration of the noise influence, but suffers as any discretization from the curse of dimensionality. A solution for systems with one single one-dimensional continuous state next to discrete states is given in [8]. In order for the approach to be applicable, the one-dimensional continuous state needs to be a continuously decreasing resource.

In technical applications, like robotics or sensor-actuator-networks, discrete-time controllers for systems with continuous-valued state spaces, e.g., the posture of a robot, but a finite set of control inputs, e.g., turn left / right or move straight, are of special importance. Therefore, in this paper a framework for discrete-time NMPC for continuous state spaces and a finite set of control inputs is presented that is based on the efficient state prediction of nonlinear stochastic models. Since an exact density representation in closed form and with constant complexity is preferable, a prediction method is applied that is founded on the approximation of the involved system transition densities by axis-aligned Gaussian mixture densities [9]. To lower the computational demands for approximating multi-dimensional transition densities, the so-called modularization for complexity reduction purposes is proposed. Thus, the Gaussian mixture representation of the predicted state can be evaluated efficiently with high approximation accuracy. As an additional part of this framework, an extremely flexible representation of the cost function, on which the optimization is based, is presented. Besides the commonly used quadratic deviation, a versatile Gaussian mixture representation of the cost function is introduced. This representation is very expressive thanks to the universal approximation property of Gaussian mixtures. Combining the efficient state prediction and the different cost function representations, an efficient integrated closed-form approach to NMPC for nonlinear noise affected systems with novel abilities is obtained.

This work is based on a publication entitled *A Closed-Form Model Predictive Control Framework for Nonlinear Noise-Corrupted Systems* [10]. The remainder of this paper is structured as follows: In the next section, the considered NMPC problem is described together with an example from the field of mobile robot control. In Section 3, the efficient closed-form prediction approach for nonlinear systems based on transition density approximation and complexity reduction is derived. Different techniques for modeling the cost function are introduced in Section 4. In Section 5, three different kinds of NMPC controllers are compared based on simulations employing the exam-



**Fig. 1.** Miniature walking robot [11].

ple system, which has been introduced in previous sections. Concluding remarks and perspectives on future work are given in Section 6.

## 2 Problem Formulation

The considered discrete-time system is given by

$$\underline{\boldsymbol{x}}_{k+1} = \underline{a}(\underline{\boldsymbol{x}}_k, \underline{u}_k, \underline{\boldsymbol{w}}_k), \quad (1)$$

where  $\underline{\boldsymbol{x}}_k$  denotes the vector-valued random variable of the system state,  $\underline{u}_k$  is the applied control input, and  $\underline{a}(\cdot)$  a nonlinear, time-invariant function.  $\underline{\boldsymbol{w}}_k$  denotes the white stationary noise affecting the system additively *element-wise*, i.e., the elements of  $\underline{\boldsymbol{w}}_k$  are processed in  $\underline{a}(\cdot)$  just additively. For details see Section 3.3. Please note that random variables are denoted by lower case bold face letters, an underscore denotes a vector-valued quantity.

### Example System

A mobile miniature walking robot (Fig. 1) is supposed to move along a given trajectory, e.g. along a wall, with constant velocity. This robot is able to superimpose left and right turns onto the forward motion. The robot's motion can be modeled similar to the motion of a two-wheeled differential-drive robot, where the system state  $\underline{\boldsymbol{x}}_k = [\boldsymbol{x}_k, \boldsymbol{\alpha}_k]^T$  comprises the distance to the wall  $\boldsymbol{x}_k$  and the robot's orientation relative to the wall  $\boldsymbol{\alpha}_k$ . This leads to the nonlinear discrete-time system equation

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + s \cdot \sin(\boldsymbol{\alpha}_k) + \boldsymbol{w}_k^x, \\ \boldsymbol{\alpha}_{k+1} &= \boldsymbol{\alpha}_k + u_k + \boldsymbol{w}_k^\alpha, \end{aligned} \quad (2)$$

where  $s$  is the robots constant step width and  $\boldsymbol{w}_k^x$  as well as  $\boldsymbol{w}_k^\alpha$  denote the noise influence on the system. The control input  $u_k$  is a steering action, i.e., a change of direction of the robot. Furthermore, the robot is equipped with sensors to measure distance  $\boldsymbol{y}_k^x$  and orientation  $\boldsymbol{y}_k^\alpha$  with respect to the wall according to

$$\begin{aligned} \boldsymbol{y}_k^x &= \boldsymbol{x}_k + \boldsymbol{v}_k^x, \\ \boldsymbol{y}_k^\alpha &= \boldsymbol{\alpha}_k + \boldsymbol{v}_k^\alpha, \end{aligned} \quad (3)$$

where  $\boldsymbol{v}_k^x$  and  $\boldsymbol{v}_k^\alpha$  describe the measurement noise. ■

At any time step  $k$ , the system state is predicted over a finite  $N$ -step prediction horizon. Within this horizon, an open-loop optimal control problem is solved, i.e., the optimal input  $\underline{u}_k^*$  is determined according to

$$\underline{u}_k^*(\underline{x}_k) = \arg \min_{\underline{u}_k} V_k(\underline{x}_k, \underline{u}_k),$$

where

$$V_k(\underline{x}_k, \underline{u}_k) = \min_{\underline{u}_{k,1:N-1}} \mathbb{E}_{\underline{x}_{k,1:N}} \left\{ g_N(\underline{x}_{k,N}) + \sum_{n=0}^{N-1} g_n(\underline{x}_{k,n}, \underline{u}_{k,n}) \right\}, \quad (4)$$

with  $\underline{x}_k = \underline{x}_{k,0}$ .  $V_k(\underline{x}_k, \underline{u}_k)$  comprises the step costs  $g_n(\underline{x}_{k,n}, \underline{u}_{k,n})$  depending on the predicted system states  $\underline{x}_{k,n}$  and the corresponding control inputs  $\underline{u}_{k,n}$ , as well as a terminal cost  $g_N(\underline{x}_{k,N})$ . This open-loop optimal control input  $\underline{u}_k^*$  is then applied to the system at time step  $k$ . In the next time step  $k + 1$ , the whole procedure is repeated, which leads to an open-loop feedback control scheme.

For most nonlinear systems, the analytical evaluation of (2) is not possible. One reason for this is the required prediction of system states for a noise-affected nonlinear system. The other one is the necessity for calculating expected values, which also cannot be performed in closed form. In the next sections an integrated approach to overcome these two problems is presented.

### 3 State Prediction

Predicting the system state is an important part in stochastic NMPC for noise-affected systems. The probability density  $\tilde{f}_{k+1}^x(\underline{x}_{k+1})$  of the system state  $\underline{x}_{k+1}$  for the next time step  $k + 1$  has to be computed utilizing the so-called Chapman-Kolmogorov equation [12]

$$\tilde{f}_{k+1}^x(\underline{x}_{k+1}) = \int_{\mathbb{R}^d} \tilde{f}_{\underline{u}_k}^T(\underline{x}_{k+1}|\underline{x}_k) \tilde{f}_k^x(\underline{x}_k) d\underline{x}_k. \quad (5)$$

The transition density  $\tilde{f}_{\underline{u}_k}^T(\underline{x}_{k+1}|\underline{x}_k)$  depends on the system described by (1). For linear systems with Gaussian noise, the Kalman filter [13] provides an exact solution to (5), as (5) is reduced to the evaluation of an integral over a multiplication of two Gaussian densities, which is analytically solvable.

For nonlinear systems, an approximate description of the predicted density  $\tilde{f}_{k+1}^x(\underline{x}_{k+1})$  is inevitable, since an exact closed-form representation is generally impossible to obtain. One possible approach to stochastic NMPC is linearizing the system and then applying the Kalman filter [14]. The resulting single Gaussian density is typically not sufficient for approximating  $\tilde{f}_{k+1}^x(\underline{x}_{k+1})$ . Hence, we propose representing all densities involved in (5) by means of Gaussian mixtures, which can be done due to their universal approximation property [15].

To reduce the complexity of approximating all density functions corresponding to system (1) and to allow an efficient state prediction, the *concept of modularization*



is proposed, see Section 3.3. Here, (1) is decomposed into vector-valued subsystems. Approximations for these subsystems in turn can be reduced to the scalar case, as stated in Section 3.2. For that purpose, in the following section a short review on the closed-form prediction approach for scalar systems with additive noise is given. Combining these techniques enables state prediction for system (1) based on Gaussian mixture approximations of the transition density functions corresponding to scalar systems.

### 3.1 Scalar Systems

For the scalar system equation

$$\mathbf{x}_{k+1} = a(\mathbf{x}_k, \underline{u}_k) + \mathbf{w}_k \text{ ,}$$

the approach proposed in [9] allows to perform a closed-form prediction resulting in an approximate Gaussian mixture representation  $f_{k+1}^x(x_{k+1})$  of  $\tilde{f}_{k+1}^x(x_{k+1})$ ,

$$f_{k+1}^x(x_{k+1}) = \sum_{i=1}^L \omega_i \cdot \mathcal{N}(x_{k+1} - \mu_i; \sigma_i^2) \text{ ,} \tag{6}$$

where  $L$  is the number of Gaussian components,  $\mathcal{N}(x_{k+1} - \mu_i; \sigma_i^2)$  is a Gaussian density with mean  $\mu_i$ , standard deviation  $\sigma_i$ , and weighting coefficients  $\omega_i$  with  $\omega_i > 0$  as well as  $\sum_{i=1}^L \omega_i = 1$ . For obtaining this approximate representation of the true predicted density that provides high accuracy especially with respect to higher-order moments and a multimodalities, the corresponding transition density  $\tilde{f}_{\underline{u}_k}^T(x_{k+1}|x_k)$  from (5) is approximated off-line by the Gaussian mixture

$$f_{\underline{u}_k}^T(x_{k+1}, x_k, \underline{\eta}) = \sum_{i=1}^L \omega_i \cdot \mathcal{N}(x_{k+1} - \mu_{i,1}; \sigma_{i,1}^2) \cdot \mathcal{N}(x_k - \mu_{i,2}; \sigma_{i,2}^2)$$

with parameter vector

$$\underline{\eta} = [\eta_1^T, \dots, \eta_L^T]^T \text{ .}$$

This Gaussian mixture comprises  $L$  axis-aligned Gaussian components (short: axis-aligned Gaussian mixture), i.e., the covariance matrices of the Gaussian components are diagonal, with parameters

$$\eta_i^T = [\omega_i, \mu_{i,1}, \sigma_{i,1}, \mu_{i,2}, \sigma_{i,2}] \text{ .}$$

The axis-aligned structure of the approximate transition density allows performing repeated prediction steps with constant complexity, i.e., a constant number  $L$  of mixture components for  $f_{k+1}^x(x_{k+1})$ .

This efficient prediction approach can be directly applied to vector-valued systems, like (1). However, off-line approximation of the multi-dimensional transition density corresponding to such a system is computationally demanding. Therefore, in the next two sections techniques to lower the computational burden are introduced.

### 3.2 Vector-Valued Systems

Now we consider the vector-valued system

$$\underline{\mathbf{x}}_{k+1} = \underline{\mathbf{a}}(\underline{\mathbf{x}}_k, \underline{\mathbf{u}}_k) + \underline{\mathbf{w}}_k, \quad (7)$$

with state vector  $\underline{\mathbf{x}}_{k+1} = [\mathbf{x}_{k+1,1}, \mathbf{x}_{k+1,2}, \dots, \mathbf{x}_{k+1,d}]^T \in \mathbb{R}^d$  and noise  $\underline{\mathbf{w}}_k = [\mathbf{w}_{k,1}, \mathbf{w}_{k,2}, \dots, \mathbf{w}_{k,d}]^T \in \mathbb{R}^d$ . Under the assumption that  $\underline{\mathbf{w}}_k$  is white and stationary (but not necessarily Gaussian or zero-mean), with *mutual independent* elements  $\mathbf{w}_{k,j}$ , approximating the corresponding transition density  $\tilde{f}_{\underline{\mathbf{w}}_k}^T(\underline{\mathbf{x}}_{k+1}|\underline{\mathbf{x}}_k) = \tilde{f}^w(\underline{\mathbf{x}}_{k+1} - \underline{\mathbf{a}}(\underline{\mathbf{x}}_k, \underline{\mathbf{u}}_k))$  can be reduced to the scalar system case.

#### Theorem 1 (Composed Transition Density)

The transition density  $\tilde{f}_{\underline{\mathbf{w}}_k}^T(\underline{\mathbf{x}}_{k+1}|\underline{\mathbf{x}}_k)$  of system (7) can be decomposed into separate transition densities of scalar systems  $a_j(\cdot)$ ,  $j = 1, 2, \dots, d$ , where

$$\underline{\mathbf{a}}(\cdot) = [a_1(\cdot), a_2(\cdot), \dots, a_d(\cdot)]^T.$$

*Proof.* Marginalizing  $\tilde{f}_{k+1}^x(\underline{\mathbf{x}}_{k+1})$  from the joint density function  $\tilde{f}_k(\underline{\mathbf{x}}_{k+1}, \underline{\mathbf{x}}_k, \underline{\mathbf{w}}_k)$  and separating the elements of  $\underline{\mathbf{w}}_k$  leads to

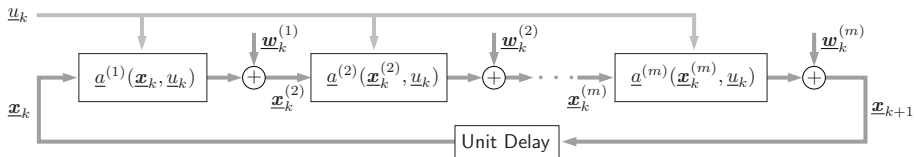
$$\begin{aligned} \tilde{f}_{k+1}^x(\underline{\mathbf{x}}_{k+1}) &= \int_{\mathbb{R}^{2d}} \delta(\underline{\mathbf{x}}_{k+1} - \underline{\mathbf{a}}(\underline{\mathbf{x}}_k, \underline{\mathbf{u}}_k) - \underline{\mathbf{w}}_k) \cdot \tilde{f}_k^x(\underline{\mathbf{x}}_k) \tilde{f}^w(\underline{\mathbf{w}}_k) d\underline{\mathbf{x}}_k d\underline{\mathbf{w}}_k \\ &= \int_{\mathbb{R}^{2d}} \prod_{j=1}^d \delta(x_{k+1,j} - a_j(\underline{\mathbf{x}}_k, \underline{\mathbf{u}}_k) - w_{k,j}) \cdot \tilde{f}_k^x(\underline{\mathbf{x}}_k) \prod_{j=1}^d \tilde{f}^{w_j}(w_{k,j}) d\underline{\mathbf{x}}_k d\underline{\mathbf{w}}_k \\ &= \int_{\mathbb{R}^d} \left( \prod_{j=1}^d \underbrace{\tilde{f}^{w_j}(x_{k+1,j} - a_j(\underline{\mathbf{x}}_k, \underline{\mathbf{u}}_k))}_{\text{separate transition densities}} \right) \tilde{f}_k^x(\underline{\mathbf{x}}_k) d\underline{\mathbf{x}}_k. \end{aligned}$$

As a result of the mutual independence of the elements in  $\underline{\mathbf{w}}_k$ , the transition density of the vector-valued system (7) is separated into  $d$  transition densities of  $d$  scalar systems. Approximating these lower-dimensional transition densities is possible with decreased computational demand [9].  $\square$

The concept of modularization, introduced in the following section, benefits strongly from the result obtained in Theorem 1.

### 3.3 Concept of Modularization

For our proposed stochastic NMPC framework, we assume that the nonlinear system is corrupted by element-wise additive noise. By incorporating this specific noise structure, the previously stated closed-form prediction step can indirectly be utilized for system (1). Similar to Rao-Blackwellized particle filters [16], we can reduce the system in (1)



**Fig. 2.** Modularization of the vector-valued system  $\underline{x}_{k+1} = \underline{a}(\underline{x}_k, \underline{u}_k, \underline{w}_k)$ .

to a set of less complex subsystems. These subsystems are of a form according to (7),

$$\begin{aligned} \underline{x}_{k+1} &= \underline{a}(\underline{x}_k, \underline{u}_k, \underline{w}_k) = \underline{a}^{(m)}(\underline{x}_k^{(m)}, \underline{u}_k) + \underline{w}_k^{(m)} \\ \underline{x}_k^{(m)} &= \underline{a}^{(m-1)}(\underline{x}_k^{(m-1)}, \underline{u}_k) + \underline{w}_k^{(m-1)} \\ &\vdots \\ \underline{x}_k^{(2)} &= \underline{a}^{(1)}(\underline{x}_k^{(1)}, \underline{u}_k) + \underline{w}_k^{(1)}. \end{aligned}$$

We name this approach *modularization*, where the subsystems

$$\underline{x}_k^{(i+1)} = \underline{a}^{(i)}(\underline{x}_k^{(i)}, \underline{u}_k) + \underline{w}_k^{(i)}, \text{ for } i = 1, \dots, m$$

correspond to transition densities that can be approximated according to Section 3.1 and 3.2. Since these subsystems are less complex than the overall system (1), approximating transition densities is also less complex. Furthermore, a nested prediction can be performed to obtain the predicted density  $f_{k+1}^x(\underline{x}_{k+1})$ , which is illustrated in Fig. 2. Starting with  $\underline{x}_k^{(1)} = \underline{x}_k$ , each subsystem  $\underline{a}^{(i)}(\cdot)$  receives an *auxiliary system state*  $\underline{x}_k^{(i)}$  and generates an auxiliary predicted system state  $\underline{x}_k^{(i+1)}$ .

The noise  $\underline{w}_k$  can be separated into its subvectors  $\underline{w}_k^{(i)}$  according to

$$\underline{w}_k = [\underline{w}_k^{(1)}, \underline{w}_k^{(2)}, \dots, \underline{w}_k^{(m)}]^T,$$

in case the noise subvectors  $\underline{w}_k^{(i)}$  are mutually independent.

**Example System: Modularization**

The system model (2) describing the mobile robot can be modularized into the subsystems

$$\underline{x}_k^{(2)} = s \cdot \sin(\alpha_k) + \underline{w}_k^x$$

and

$$\begin{aligned} \underline{x}_{k+1} &= \underline{x}_k + \underline{x}_k^{(2)}, \\ \alpha_{k+1} &= \alpha_k + u_k + \underline{w}_k^\alpha. \end{aligned}$$

The auxiliary system state  $\underline{x}_k^{(2)}$  is stochastically dependent on  $\alpha_k$ . We omit this dependence in further investigations of the example system for simplicity. ■

Please note that there are typically stochastic dependencies between several auxiliary system states. To consider this fact, the relevant auxiliary system states have to be augmented to conserve the dependencies. Thus, the dimensions of these auxiliary states need not all to be equal.

## 4 Cost Functions

In this section, two possibilities to model cost functions, the well-known quadratic deviation and a novel approach employing Gaussian mixture cost functions, are presented. Exploiting the fact that the predicted state variables are, as explained in the previous section, described by Gaussian mixture densities, the necessary evaluation of the expected values in (4) can be performed efficiently in closed-form for both options.

In the following, cumulative cost functions according to (4) are considered, where  $g_n(\underline{\mathbf{x}}_{k,n}, \underline{\mathbf{u}}_{k,n})$  denotes a step cost within the horizon and  $g_N(\underline{\mathbf{x}}_{k,N})$  a cost depending on the terminal state at the end of the horizon.

For simplicity, step costs that are additively decomposable according to

$$g_n(\underline{\mathbf{x}}_n, \underline{\mathbf{u}}_n) = g_n^x(\underline{\mathbf{x}}_n) + g_n^u(\underline{\mathbf{u}}_n)$$

are considered, although the proposed framework is not limited to this case.

### 4.1 Quadratic Cost

One of the most popular cost functions is the quadratic deviation from a target value  $\check{\underline{\mathbf{x}}}$  or  $\check{\underline{\mathbf{u}}}$  according to

$$g_n^x(\underline{\mathbf{x}}_n) = (\underline{\mathbf{x}}_n - \check{\underline{\mathbf{x}}}_n)^\top (\underline{\mathbf{x}}_n - \check{\underline{\mathbf{x}}}_n).$$

As in our framework the probability density function of the state  $\underline{\mathbf{x}}_n$  is given by an axis-aligned Gaussian mixture  $f_n^x(\underline{\mathbf{x}}_n)$  with  $L$  components, the calculation of  $\mathbb{E}_{\underline{\mathbf{x}}_n}\{g_n^x(\underline{\mathbf{x}}_n)\}$ , which is necessary to compute (4), can be performed analytically as it can be interpreted as the sum over shifted and dimension-wise calculated second-order moments

$$\begin{aligned} \mathbb{E}_{\underline{\mathbf{x}}_n}\{g_n^x(\underline{\mathbf{x}}_n)\} &= \mathbb{E}_{\underline{\mathbf{x}}_n}\{(\underline{\mathbf{x}}_n - \check{\underline{\mathbf{x}}}_n)^\top (\underline{\mathbf{x}}_n - \check{\underline{\mathbf{x}}}_n)\} \\ &= \text{trace} \mathbb{E}_{\underline{\mathbf{x}}_n}^2\{(\underline{\mathbf{x}}_n - \check{\underline{\mathbf{x}}}_n)\} \\ &= \text{trace} \sum_{i=1}^L \omega_i \left( (\underline{\boldsymbol{\mu}}_i - \check{\underline{\mathbf{x}}}_n)(\underline{\boldsymbol{\mu}}_i - \check{\underline{\mathbf{x}}}_n)^\top + \text{diag}(\underline{\boldsymbol{\sigma}}_i^2) \right), \end{aligned}$$

employing  $\mathbb{E}_{\mathbf{x}}^2\{\mathbf{x}\} = \sum_{i=1}^L \omega_i (\boldsymbol{\mu}_i^2 + \boldsymbol{\sigma}_i^2)$ .

#### Example System: Quadratic Cost

If the robot is intended to move parallel along the wall, the negative quadratic deviation of the angle  $\alpha_k$  with respect to the wall, i.e.,  $g_n^\alpha(\alpha_n) = (\alpha_n - \alpha_n^{Wall})^2$ , is a suitable cost function. ■

## 4.2 Gaussian Mixture Cost

A very versatile description of the cost function can be realized if Gaussian mixtures are employed. In this case, arbitrary cost functions can be realized due to the Gaussian mixtures' universal approximation property [15]. Obviously, in this case the Gaussian mixtures may have arbitrary parameters, e.g., negative weights  $\omega$ .

### Example System: Gaussian Mixture Cost Function

In case the robot is intended to move at a certain optimal distance to the wall (e.g.,  $\tilde{x}_n = 2$ , with  $x_n^{Wall} = 0$ ), where being closer to the wall is considered less desirable than being farther away, this can, e.g., be modeled with a cost function as depicted in Fig. 3 (a). If two different distances are considered equally optimal, this can be modeled with a cost function as depicted in Fig. 3 (b). ■

Here, the calculation of the expected value  $E_{\underline{x}_n}\{g_n^x(\underline{x}_n)\}$ , which is necessary for the calculation of (4), can also be performed analytically

$$\begin{aligned}
 E_{\underline{x}_n}\{g_n^x(\underline{x}_n)\} &= \int_{\mathbb{R}^d} f_n^x(\underline{x}_n) \cdot g_n^x(\underline{x}_n) d\underline{x}_n \\
 &= \int_{\mathbb{R}^d} \sum_{i=1}^L \omega_i \mathcal{N}(\underline{x}_n - \underline{\mu}_i; \text{diag}(\underline{\sigma}_i)^2) \\
 &\quad \cdot \sum_{j=1}^M \omega_j \mathcal{N}(\underline{x}_n - \underline{\mu}_j; \text{diag}(\underline{\sigma}_j)^2) d\underline{x}_n \\
 &= \sum_{i=1}^L \sum_{j=1}^M \omega_{ij} \underbrace{\int_{\mathbb{R}^d} \mathcal{N}(\underline{x}_n - \underline{\mu}_{ij}; \text{diag}(\underline{\sigma}_{ij})^2) d\underline{x}_n}_{=1}, \quad (8)
 \end{aligned}$$

with

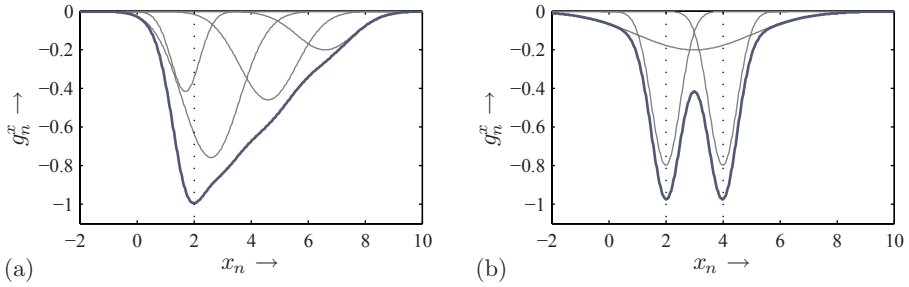
$$\omega_{ij} = \omega_i \omega_j \cdot \mathcal{N}(\underline{\mu}_i - \underline{\mu}_j; \text{diag}(\underline{\sigma}_i)^2 + \text{diag}(\underline{\sigma}_j)^2),$$

where  $f_n^x(\underline{x}_n)$  denotes the  $L$ -component Gaussian mixture probability density function of the system state (6) and  $g_n^x(\underline{x}_n)$  the cost function, which is a Gaussian mixture with  $M$  components.

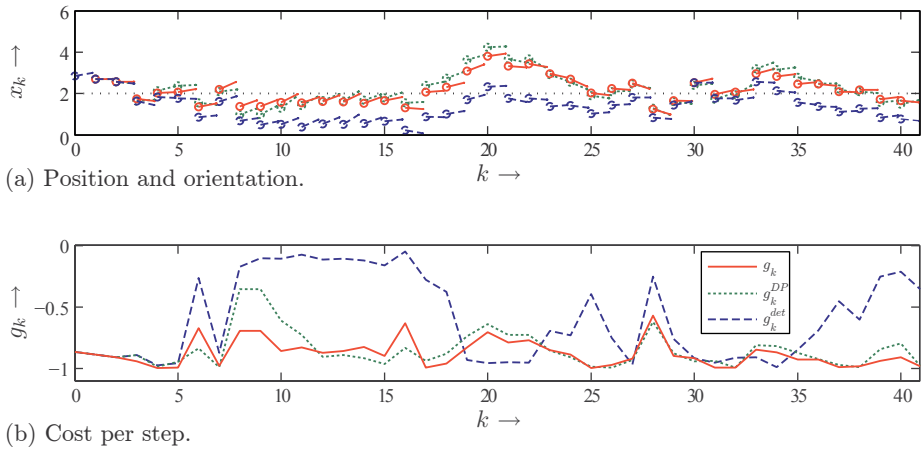
The versatility of the cost function representation can even be increased if also Dirac mixtures, i.e., weighted sums of Dirac delta distributions, are employed. This allows to penalize (or reward) individual discrete configurations of the continuous-valued state space. The calculation of the expected value  $E_{\underline{x}_n}\{g_n^x(\underline{x}_n)\}$  can be carried out similarly to (8) for Dirac mixtures as well as for the sum of Gaussian and Dirac mixtures [17].

## 4.3 Input Dependent Part

The input dependent part of the cost function  $g_n^u(\underline{u}_n)$  can either be modeled similar to the procedures described above or with a lookup-table since there is just a finite number of discrete  $\underline{u}_n$ .



**Fig. 3.** Asymmetric and multimodal cost functions consisting of four and three components (gray), respectively.



**Fig. 4.** First 40 steps of a simulation (red solid line: stochastic NMPC, green dotted line: stochastic NMPC with DP, blue dashed line: deterministic NMPC).

Using the efficient state prediction presented in Section 3 together with the value function representations presented above, (2) can be solved analytically for a finite set of control inputs. Thus, an efficient closed-form solution for the optimal control problem within stochastic NMPC is available. Its capabilities will be illustrated by simulations in the next section.

## 5 Simulations

Based on the above example scenario, several simulations are conducted to illustrate the modeling capabilities of the proposed framework as well as to illustrate the benefits that can be gained by the direct consideration of noise in the control. The considered system is given by (2) and (3), with  $s = 1$  and  $u_k \in \{-0.2, -0.1, 0, 0.1, 0.2\}$ . The considered noise influences on the system  $w_k^x$  and  $w_k^\alpha$  are zero-mean white Gaussian noise with standard deviation  $\sigma_w^x = 0.5$  and  $\sigma_w^\alpha = 0.05 \approx 2.9^\circ$  respectively. The

measurement noise is also zero-mean white Gaussian noise with standard deviation  $\sigma_v^x = 0.5$  and  $\sigma_v^\alpha = 0.1 \approx 5.7^\circ$ . All simulations are performed for an  $N = 4$  step prediction horizon, with a cumulative cost function according to (4), where  $g_N(\mathbf{x}_{k,N})$  is the function depicted in Fig. 3 (a) and  $g_n(\mathbf{x}_{k,n}, \mathbf{u}_{k,n}) = g_N(\mathbf{x}_{k,N}) \forall n$ . In addition, the modularization is employed as described above.

To evaluate the benefits of the proposed NMPC framework, three different kinds of simulations are performed:

*Calculation of the input without noise consideration (deterministic NMPC):*

The deterministic or certainty equivalence control is calculated as a benchmark neglecting the noise influence.

*Direct calculation of the optimal input considering all noise influences (stochastic NMPC):*

The direct calculation of the open-loop feedback control input with consideration of the noise is performed using the techniques presented in the previous sections. Thus, it is possible to execute all calculations analytically without the need for any numerical method. Still, this approach has the drawback that the computational demand for the optimal control problem increases exponentially with the length of the horizon  $N$ , which makes it only suitable for short horizons.

*Calculation of the optimal input with a value function approximation scheme and Dynamic Programming (stochastic NMPC with Dynamic Programming):*

In order to be able to use the framework efficiently also for long prediction horizons as well as to consider state information within the prediction horizon (closed-loop feedback or optimal control), it is necessary to employ Dynamic Programming (DP). Unfortunately, this is not directly possible, as no closed-form solution for the value function  $J_n$  is available. One easy but either not very accurate or computationally demanding solution would be to discretize the state space. More advanced solutions can be found by value function approximation [18]. For the simulations, a value function approximation as described in [7] is employed that is well-suited with regard to closed-form calculations. Here, the state space is discretized by covering it with a finite set of Gaussians with fixed means and covariances. Then weights, i.e., scaling factors, are selected in such a way that the approximate and the true value function coincide at the means of every Gaussian. Using these approximate value functions together with the techniques described above, again all calculations can be executed analytically. In contrast to the direct calculation, now the computational demand increases only linearly with the length of the prediction horizon but quadratically in the number of Gaussians used to approximate the value function. Here, the value functions are approximated by a total of 833 Gaussians equally spaced over the state space within  $(\hat{x}_n, \hat{\alpha}_n) \in \Omega := [-2, 10] \times [-2, 2]$ .

For each simulation run, a particular noise realization is used that is applied to the different controllers. In Fig. 4(a), the first 40 steps of a simulation run are shown. The distance to the wall  $x_k$  is depicted by the position of the circles, the orientation  $\alpha_k$  by the orientation of the arrows. It can be clearly seen that the system is heavily influenced

**Table 1.** Simulation Results.

controller	average cost
deterministic NMPC	-0.6595 (100.00%)
stochastic NMPC	-0.7299 (110.66%)
stochastic NMPC with DP	-0.6824 (103.48%)

by noise and that the robot under deterministic control behaves very differently from the other two. The deterministic controller just tries to move the robot to the minimum of the cost function at  $\tilde{x}_k = 2$  and totally neglects the asymmetry of the cost function. The stochastic controllers lead to a larger distance to the wall, as they consider the noise affecting the system in conjunction with the non-symmetric cost function.

In Fig. 4(b), the evaluation of the cost function for each step is shown. As expected, both stochastic controllers perform much better, i.e., they generate less cost, than the deterministic one. This finding has been validated by a series of 100 Monte Carlo simulations with different noise realizations and initial values. The uniformly distributed initial values are sampled from the interval  $x_0 \in [0, 8]$  and  $\alpha_0 \in [-\pi/4, \pi/4]$ . In Table 1, the average step costs of the 100 simulations with 40 steps each are shown. To facilitate the comparison, also normalized average step costs are given. Here, it can be seen that the stochastic controller outperforms the deterministic one by over 10% in terms of cost. In 82% of the runs, the stochastic controller gives overall better results than the deterministic one. By employing dynamic programming together with value function approximation the benefits are reduced. Here, the deterministic controller is only outperformed by approximately 3.5%. The analysis of the individual simulations leads to the conclusion that the control quality significantly degrades in case the robot attains a state which is less well approximated by the value function approximation as it lies outside  $\Omega$ . Still, the dynamic programming approach produced better results than the deterministic approach in 69% of the runs.

These findings illustrate the need for advanced value function approximation techniques in order to gain good control performance. One approach that seamlessly integrates into the presented SNMPC framework and that outperforms the employed value function approximation significantly is presented in [19]. This is possible by abandoning the grid approximation approach and using Gaussian kernels that are placed in an optimized fashion.

## 6 Conclusions

A novel framework for closed-form stochastic Nonlinear Model Predictive Control for a continuous state space and a finite set of control inputs has been presented that directly incorporates the noise influence in the corresponding optimal control problem. By using the proposed state prediction methods, which are based on transition density approximation by Gaussian mixture densities and complexity reduction techniques, the otherwise not analytically solvable state prediction of nonlinear noise affected systems can be performed in an efficient closed-form manner. Another very important aspect of NMPC is the modeling of the cost function. The proposed methods also use Gaussian mixtures, which leads to a level of flexibility far beyond the traditional representations.



By employing the same representation for both the predicted probability density functions and the cost functions, stochastic NMPC is solvable in closed-form for nonlinear systems with consideration of noise influences. The effectiveness of the presented framework and the importance of the consideration of noise in the controller have been shown in simulations of a walking robot following a specified trajectory.

One interesting future extension will be the incorporation of the state estimation in the control, which is important for nonlinear systems and nonquadratic cost functions, as here the separation principle does not hold. An additional interesting aspect will be the consideration of effects of inhomogeneous noise, i.e., noise with state and/or input dependent noise levels. Here, the consideration of the stochastic behavior of the system is expected to have an even greater impact on the control quality. Also the extension to new application fields is intended. Of special interest is the extension to the related emerging field of Model Predictive Sensor Scheduling [20, 21], which is of special importance, e.g. in sensor-actuator-networks.

## References

1. Qin, S.J., Badgwell, T.A.: An Overview of Industrial Model Predictive Control Technology. *Chemical Process Control* 93(316) (1997) 232–256
2. Findeisen, R., Allgwer, F.: An Introduction to Nonlinear Model Predictive Control. In: 21st Benelux Meeting on Systems and Control. (March 2002) 119–141
3. Ohtsuka, T.: A Continuation/GMRES Method for Fast Computation of Nonlinear Receding Horizon Control. *Automatica* 40(4) (April 2004) 563–574
4. Camacho, E.F., Bordons, C.: *Model Predictive Control*. 2 edn. Springer-Verlag London Ltd. (June 2004)
5. Kappen, H.J.: Path integrals and symmetry breaking for optimal control theory. *Journal of Statistical Mechanics: Theory and Experiments* 2005(11) (November 2005) P11011
6. Deisenroth, M.P., Weissel, F., Ohtsuka, T., Hanebeck, U.D.: Online-Computation Approach to Optimal Control of Noise-Affected Nonlinear Systems with Continuous State and Control Spaces. In: *Proceedings of the European Control Conference (ECC 2007)*, Kos, Greece (July 2007)
7. Nikovski, D., Brand, M.: Non-Linear Stochastic Control in Continuous State Spaces by Exact Integration in Bellman's Equations. In: *Proceedings of the 2003 International Conference on Automated Planning and Scheduling*. (June 2003) 91–95
8. Marecki, J., Koenig, S., Tambe, M.: A Fast Analytical Algorithm for Solving Markov Decision Processes with Real-Valued Resources. In: *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*. (January 2007)
9. Huber, M., Brunn, D., Hanebeck, U.D.: Closed-Form Prediction of Nonlinear Dynamic Systems by Means of Gaussian Mixture Approximation of the Transition Density. In: *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*. (September 2006) 98–103
10. Weissel, F., Huber, M.F., Hanebeck, U.D.: A Closed-Form Model Predictive Control Framework for Nonlinear Noise-Corrupted Systems. In: *4th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2007)*. Volume SPSMC., Angers, France (May 2007) 62–69
11. Weissel, F., Huber, M.F., Hanebeck, U.D.: Test-Environment based on a Team of Miniature Walking Robots for Evaluation of Collaborative Control Methods. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*. (November 2007)

12. Schweppe, F.C.: *Uncertain Dynamic Systems*. Prentice-Hall (1973)
13. Kalman, R.E.: A new Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME, Journal of Basic Engineering* 82 (March 1960) 35–45
14. Lee, J.H., Ricker, N.L.: Extended Kalman Filter Based Nonlinear Model Predictive Control. In: *Industrial & Engineering Chemistry Research*. Volume 33., ACS (1994) 1530–1541
15. Maz'ya, V., Schmidt, G.: On Approximate Approximations using Gaussian Kernels. *IMA Journal of Numerical Analysis* 16(1) (1996) 13–29
16. de Freitas, N.: Rao-Blackwellised Particle Filtering for Fault Diagnosis. In: *IEEE Aerospace Conference Proceedings*. Volume 4. (2002) 1767–1772
17. Weissel, F., Huber, M.F., Hanebeck, U.D.: A Nonlinear Model Predictive Control Framework Approximating Noise Corrupted Systems with Hybrid Transition Densities. In: *IEEE Conference on Decision and Control (CDC 2007)*, New Orleans, USA (December 2007)
18. Bertsekas, D.P.: *Dynamic Programming and Optimal Control*. 2nd edn. Athena Scientific, Belmont, Massachusetts, U.S.A. (2000)
19. Weissel, F., Huber, M.F., Hanebeck, U.D.: Efficient Control of Nonlinear Noise-Corrupted Systems Using a Novel Model Predictive Control Framework. In: *Proceedings of the 2007 American Control Conference (ACC 2007)*, New York City, USA (July 2007)
20. He, Y., Chong, E.K.P.: Sensor Scheduling for Target Tracking in Sensor Networks. In: *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC 2004)*. Volume 1. (December 2004) 743–748
21. Savkin, A.V., Evans, R.J., Skafidas, E.: The Problem of Optimal Robust Sensor Scheduling. In: *Proceedings of the 39th IEEE Conference on Decision and Control (CDC 2000)*. Volume 4. (December 2000) 3791–3796

# The Conjugate Gradient Partitioned Block Frequency-Domain for Multichannel Adaptive Filtering

Lino García Morales

Universidad Europea de Madrid  
Departamento Electrónica y Comunicaciones, Spain  
lino.garcia@uem.es

**Abstract.** The conjugate gradient is the most popular optimization method for solving large systems of linear equations. In a system identification problem, for example, where very large impulse response is involved, it is necessary to apply a particular strategy which diminishes the delay, while improving the convergence time. In this paper we propose a new scheme which combines frequency-domain adaptive filtering with a conjugate gradient technique in order to solve a high order multichannel adaptive filter, while being delayless and guaranteeing a very short convergence time.

## 1 Introduction

The multichannel adaptive filtering problem's solution depends on the correlation between the channels, the number of channels and the order and nature of the impulse responses involved in the system. The multichannel acoustic echo cancellation (MAEC) application, for example, can be seen as a system identification problem with extremely large impulse responses (depending on the environment and its reverberation time, the echo paths can be characterized by FIR filters with thousands of taps).

In these cases a multirate adaptive scheme such a partitioned block frequency-domain adaptive filter (PBFDAF) [8] is a good alternative and is widely used in commercial systems nowadays. However, the convergence speed may not be fast enough under certain circumstances.

Figure 1 shows the working framework, where  $x_p$  represents the  $p$  channel input signal,  $d$  the desired signal,  $y$  the output of adaptive filter and  $e$  the error signal we try to minimize. In typical scenarios, the filter input signals  $x_p$ ,  $p = 1, \dots, P$  (where  $P$  is a number of channels), are highly correlated which further reduces the overall convergence of the adaptive filter coefficients  $w_{pm}$ ,  $m = 1, \dots, L$  ( $L$  is the filter length),

$$y(n) = \sum_{p=1}^P \sum_{m=1}^L x_p(n-m) w_{pm} . \quad (1)$$

The mean square error (MSE) minimization of the multichannel signal with respect to the filter coefficients is equivalent to the Wiener-Hopf equation

$$Rw = r . \quad (2)$$

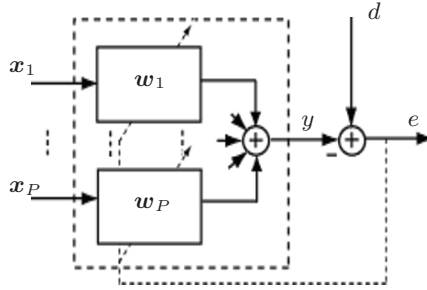


Fig. 1. Multichannel Adaptive Filtering.

$\mathbf{R}$  represents the autocorrelation matrix and  $\mathbf{r}$  the cross-correlation vector between the input and the desired signals. Both are a priori time-domain statistical unknown variables, although can be estimated iteratively from  $\mathbf{x}$  and  $d$ .

$\mathbf{R} = E\{\mathbf{x}\mathbf{x}^H\}$  and  $\mathbf{r} = E\{\mathbf{x}d^*\}$ , with  $\mathbf{x} = [x_1^T \cdots x_P^T]^T$ ;  $\mathbf{w} = [w_1^T \cdots w_P^T]^T$  and  $w_p = [w_{p1} \cdots w_{pL}]^T$ . In the notation we are using  $a$  for scalar,  $\mathbf{a}$  for vector and  $\mathbf{A}$  for matrix;  $\mathbf{a}, \mathbf{A}$  denotes vector and matrix respectively in a frequency-domain:  $\mathbf{a} = \mathbf{F}\mathbf{a}$ ,  $\mathbf{A} = \mathbf{F}\mathbf{A}$ .  $\mathbf{F}$  represents the discrete Fourier transform (DFT) matrix defined as  $F_{kl} = \exp^{-j2\pi kl/M}$ , with  $k, l = 0, \dots, M - 1, j = \sqrt{-1}$  and  $\mathbf{F}^{-1}$  as its inverse. Of course, in the final implementation, the DFT matrix is substituted by much more efficient fast Fourier transforms (FFT). Here  $(\cdot)^T$  denotes transpose operator and  $(\cdot)^H = ((\cdot)^T)^*$  the Hermitian operator (conjugate transpose).

The conjugate gradient (CG) method is efficient to obtain the solution to (2), however, a big delay is introduced (noted that the system order is  $LP \times LP$ ). In order to reduce this convergence speed problem we propose a new algorithm which employs much more powerful CG optimization techniques, but keeping the frequency block partition strategy to allow computationally realistic low latency situations.

The chapter is organized as follows: Section 2 reviews the Multichannel PBFDAF approach and its implementation. Section 3 develops the Multichannel Conjugate Gradient Partitioned Frequency Domain Adaptive Filter algorithm (PBFDAF-CG). Results of the new approach are presented in Section 4 and 5 followed by conclusions.

## 2 PBFDAF

The PBFDAF was developed to deal efficiently with such situations. The PBFDAF is a more efficient implementation of Least Mean Square (LMS) algorithm in the frequency-domain. It reduces the computational burden and user-delay bounded. In general, the PBFDAF is widely used due to be good trade-off between speed, computational complexity and overall latency. However, when working with long impulse response, as the acoustic impulse responses (AIR) used in MAEC, the convergence properties provided by the algorithm may not be enough. Besides, the multichannel adaptive filter is structurally more difficult, in general, than the single channel case [4].

This technique makes a sequential partition of the impulse response in the time-domain prior to a frequency-domain implementation of the filtering operation. This

time segmentation allows setting up individual coefficient updating strategies concerning different sections of the adaptive canceller, thus avoiding the need for disabling the adaptation in the complete filter. The adaptive algorithm is based on the frequency-domain adaptive filter (FDAF) for every section of the filter [7].

The main idea of frequency-domain adaptive filter is to frequency transform the input signal in order to work with matrix multiplications instead of dealing with slow convolutions. The frequency-domain transform employs one or more DFTs and can be seen as a pre-processing block that generates decorrelated output signals.

In the more general FDAF case, the output of the filter in the time domain (1) can be seen as a direct frequency-domain translation of the block LMS (BLMS) algorithm. In the PBFDAF case, the filter is partitioned transversally in an equivalent structure. Partitioning  $w_p$  in  $Q$  segments ( $K$  length) we obtain

$$y(n) = \sum_{p=1}^P \sum_{q=1}^Q \sum_{m=0}^{K-1} x_p(n - qK - m) w_{p(qK+m)} . \tag{3}$$

Where the total filter length  $L$ , for each channel, is a multiple of the length of each segment  $L = QK$ ,  $K \leq L$ . Thus, using the appropriate data sectioning procedure, the  $Q$  linear convolutions (per channel) of the filter can be independently carried out in the frequency-domain with a total delay of  $K$  samples instead of the  $QK$  samples needed in standard FDAF implementations.

Figure 2 shows the block diagram of the algorithm using the overlap-save method. In the frequency domain with matrix notation, equation (3) can be expressed as

$$\mathbf{Y} = \mathbf{X} \otimes \mathbf{W} . \tag{4}$$

Where  $\mathbf{X} = \mathbf{F}\mathbf{X}$  represents a matrix of dimensions  $M \times Q \times P$  which contains the Fourier transform of the  $Q$  partitions and  $P$  channels of the input signal matrix  $\mathbf{X}$ . Being  $\mathbf{X}$ ,  $2K \times P$ -dimensional (supposing 50% overlapping between the new block and the previous one). It should be taken into account that the algorithm adapts every  $K$  samples.  $\mathbf{W}$  represents the filter coefficient matrix adapted in the frequency-domain (also  $M \times Q \times P$ -dimensional) while the  $\otimes$  operator multiplies each of the elements one by one; which in (4) represents a circular convolution.

The output vector  $\mathbf{y}$  can be obtained as the double sum (rows) of the  $\mathbf{Y}$  matrix. First we obtain a  $M \times P$  matrix which contains the output of each channel in the frequency-domain  $\mathbf{y}_p$ ,  $p = 1, \dots, P$ , and secondly, adding all the outputs we obtain the output of the whole system  $\mathbf{y}$ . Finally, the output in the time-domain is obtained by using

$$\mathbf{y} = \text{last } K \text{ components of } \mathbf{F}^{-1}\mathbf{y} . \tag{5}$$

Notice that the sums are performed prior to the time-domain translation. In this way we reduce  $(P - 1)(Q - 1)$  FFTs in the complete filtering process. As in any adaptive system the error can be obtained as

$$\mathbf{e} = \mathbf{d} - \mathbf{y} , \tag{6}$$

with  $\mathbf{d} = [d(mK) \dots d((m + 1)K - 1)]^T$ .

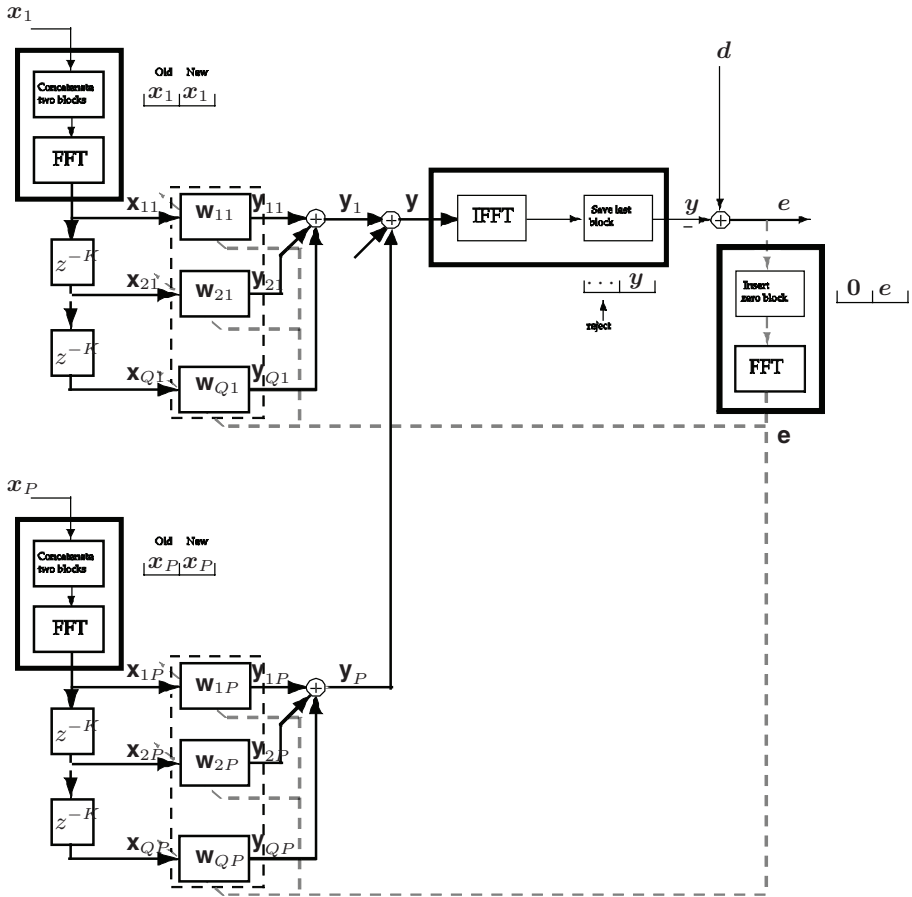


Fig. 2. Multichannel PBFDAF (Overlap-Save method).

The error in the frequency-domain (for the actualization of the filter coefficients) can be obtained as

$$\mathbf{e} = \mathbf{F} \begin{bmatrix} \mathbf{0}_{K \times 1} \\ e \end{bmatrix} . \tag{7}$$

As we can see, a block of  $K$  zeros is added to ensure a correct linear convolution implementation. In the same way, for the block gradient estimation, it is necessary to employ the same error vector in the frequency-domain for each partition  $q$  and channel  $p$ .

This can be achieved by generating an error matrix  $\mathbf{E}$  with dimensions  $M \times Q \times P$  which contains replicas of the error vector, defined in (7), of dimensions  $P$  and  $Q$  ( $\mathbf{E} \leftarrow \mathbf{e}$  in the notation). The actualization of the weights is performed as

$$\mathbf{W}(m+1) = \mathbf{W}(m) + 2\mu(m) \otimes \mathbf{G}(m) . \tag{8}$$

The instantaneous gradient is estimated as

$$\mathbf{G} = -\mathbf{X}^* \otimes \mathbf{E} . \tag{9}$$

This is the unconstrained version of the algorithm which saves two FFTs from the computational burden at the cost of decreasing the convergence speed. As we are trying to improve specifically this parameter we have implemented the constrained version which basically makes a gradient projection. The gradient matrix is transformed into the time-domain and is transformed back into the frequency-domain using only the first  $K$  elements of  $\mathbf{G}$  as

$$\mathbf{G} = \mathbf{F} \begin{bmatrix} \mathbf{G} \\ \mathbf{0}_{K \times Q \times P} \end{bmatrix} . \tag{10}$$

### 3 PBFDAF-CG

CG algorithm is a technique originally developed to minimize quadratic functions, as (2), which was later adapted for the general case [6]. Its main advantage is its speed as it converges in a finite number of steps. In the first iteration it starts estimating the gradient, as in the steepest descent (SD) method, and from there it builds successive directions that create a set of mutually conjugate vectors with respect to the positively defined Hessian (in our case, the auto-correlation matrix  $\mathbf{R}$  in the frequency-domain).

In each  $m$ -block iteration the conjugate gradient algorithm will iterate  $k = 1, \dots, \min(N, K)$  times; where  $N$  represent the memory of the gradient estimation,  $N \leq K$ . In a practical system the algorithm is stopped when it reaches a user-determined MSE level. To apply this conjugate gradient approach to the PBFDAF algorithm the weight actualization equation (8) must be modified as

$$\mathbf{w}(m+1) = \mathbf{w}(m) + \alpha \mathbf{v}(m) . \tag{11}$$

Where  $\mathbf{w}$  is the coefficient vector of dimension  $MQP \times 1$  which results from rearranging matrix  $\mathbf{W}$  (in the notation  $\mathbf{w} \leftarrow \mathbf{W}$ ).  $\mathbf{v}$  is a finite  $\mathbf{R}$ -conjugated vector set which satisfies  $\mathbf{v}_i^H \mathbf{R} \mathbf{v}_j = 0, \forall i \neq j$ . The  $\mathbf{R}$ -conjugacy property is useful as the linear independency of the conjugate vector set allows expanding the  $\mathbf{w}^*$  solution as

$$\mathbf{w}^* = \alpha_0 \mathbf{v}_0 + \dots + \alpha_{K-1} \mathbf{v}_{K-1} = \sum_{k=0}^{K-1} \alpha_k \mathbf{v}_k . \tag{12}$$

Starting at any point  $\mathbf{w}_0$  of the weighting space, we can define  $\mathbf{v}_0 = -\mathbf{g}_0$  being  $\mathbf{g}_0 \leftarrow \bar{\mathbf{G}}_0, \bar{\mathbf{G}}_0 = \nabla(\mathbf{W}_0), \mathbf{p}_0 \leftarrow \bar{\mathbf{P}}_0, \bar{\mathbf{P}}_0 = \nabla(\mathbf{W}_0 - \bar{\mathbf{G}}_0)$ .

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \mathbf{v}_k , \tag{13}$$

$$\alpha_k = \frac{\mathbf{g}_k^H \mathbf{v}_k}{\mathbf{v}_k^H (\mathbf{g}_k - \mathbf{p}_k)} , \tag{14}$$

$$\mathbf{g}_{k+1} \leftarrow \bar{\mathbf{G}}_{k+1}, \bar{\mathbf{G}}_{k+1} = \nabla(\mathbf{W}_{k+1}) , \tag{15}$$

$$\mathbf{p}_{k+1} \leftarrow \bar{\mathbf{P}}_{k+1}, \bar{\mathbf{P}}_{k+1} = \nabla(\mathbf{W}_{k+1} - \bar{\mathbf{G}}_{k+1}) ,$$

$$\mathbf{v}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{v}_k , \tag{16}$$

$$\beta_k^{HS} = \frac{\mathbf{g}_{k+1}^H (\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{v}_k^H (\mathbf{g}_{k+1} - \mathbf{g}_k)} . \tag{17}$$

Where  $\mathbf{p}_k$  represents the gradient estimated in  $\mathbf{w}_k - \mathbf{g}_k$ . For that, it is necessary to evaluate  $\mathbf{Y} = \mathbf{X} \otimes (\mathbf{W} - \mathbf{G})$ , (5), (6), (7) and (9). In order to be able to generate nonzero direction vectors which are conjugate to the initial negative gradient vector, a gradient estimation is necessary [5]. This gradient estimation is obtained by averaging the instantaneous gradient estimates over  $N$  past values. The  $\nabla$  operator is an averaging gradient estimation with the current weights and  $N$  past inputs  $\mathbf{X}$  and  $\mathbf{d}$ ,

$$\bar{\mathbf{G}}_k = \nabla (\mathbf{W}_k) = \frac{2}{N} \sum_{n=0}^{N-1} \mathbf{G}_{k-n} \Big|_{\mathbf{W}_k, \mathbf{X}_{k-n}, \mathbf{d}_{k-n}} \quad (18)$$

This alternative approach does not require knowing neither the Hessian nor the employment of a linear search. Notice that all the operations (13-17) are vector operations that keep the computational complexity low. The equation (17) is known as the Hestenes-Stiefel method but there are different approaches for calculating  $\beta_k$ : Fletcher-Reeves (19), Polar-Ribire (20) and Dai-Yuan (21) methods.

$$\beta_k^{FR} = \frac{\mathbf{g}_{k+1}^H \mathbf{g}_{k+1}}{\mathbf{g}_k^H \mathbf{g}_k} \quad (19)$$

$$\beta_k^{PR} = \frac{\mathbf{g}_{k+1}^H (\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{g}_k^H \mathbf{g}_k} \quad (20)$$

$$\beta_k^{DY} = \frac{\mathbf{g}_{k+1}^H \mathbf{g}_{k+1}}{\mathbf{v}_k^H (\mathbf{g}_{k+1} - \mathbf{g}_k)} \quad (21)$$

The constant  $\beta_k$  is chosen to provide  $\mathbf{R}$ -conjugacy for the vector  $\mathbf{v}_k$  with respect to the previous direction vectors  $\mathbf{v}_{k-1}, \dots, \mathbf{v}_0$ . Instability occurs whenever  $\beta_k$  exceeds unity.

In this approach, the successive directions are not guaranteed to be conjugate to each other, even when one uses the exact value of the gradient at each iteration. To ensure the algorithm stability the gradient can be initialized forcing  $\beta_k = 1$  in (16) when  $\beta_k > 1$ .

### 4 Computational Cost

Table 1 shows a comparative analysis for both algorithms in terms of operations number (multiplications, sums) clustered by functionality. Note that constants  $A$ ,  $B$  and  $C$ , in the PBFDAF computational burden estimation, are used as reference for the number of operations in PBFDAF-CG. For one iteration ( $k = 1$ ), the computational cost of the PBFDAF-CG is 40 times higher than the PBFDAF.

**Table 1.** Computational Cost Comparative ( $O = PQM$ ).

Alg./Op.	Gradient Estimation and Convolution	Updating	Constrained Version
PBFDAF	$A = (P + 2) O \log_2 O + P(Q(M + 1) + 1) + K + O$	$B = 9O$	$C = 2O \log_2 O$
PBFDAF-CG	$((N(A + 1) + 1) + 1) 2 + 1 (k + 1)$	$(13O + 2) k$	$2CN(k + 1)$



### 5 Simulation Examples

MAEC application is a good example of complex system identification because has to deal with very long adaptive filters in order to achieve good results. The scenario employed in our tests simulates two small chambers imitating a typical teleconference environment depicted in Fig. 3. Following an acoustic opening approach, both chambers can be acoustically connected by means of linear arrays of microphones and loudspeakers. Details of this configuration follow. Room dimensions are [2000 2440 2700] mm.

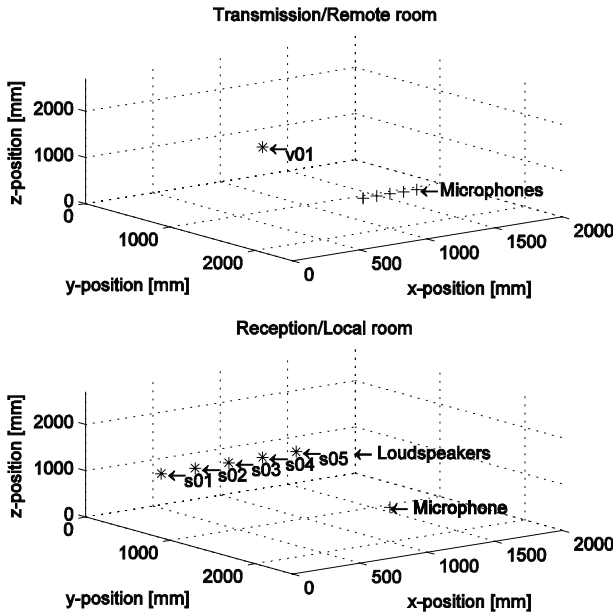


Fig. 3. Working environment for the tests.

The impulse responses are calculated using the image method [2] with an expected reverberation time of 70ms (reflection coefficients [0.8 0.8; 0.5 0.5; 0.6 0.6]). The speech source, microphones and loudspeakers are situated as in Fig. 3. In the emitting room, the source is located in [1000 500 1000] and the microphones in [{800 900 1000 1100 1200} 2000 750]. Notice that the microphone separation is only 10 cm, which would be a worse case scenario that provides with highly correlated signals. In the reception room the loudspeakers are situated in [{500 750 1000 1250 1500} 100 750] and the microphone in [1000 2000 750].

The directivity patterns of the loudspeakers ([elevation 0°, azimuth -90°, aperture beam 180°]) and the microphones ([0° 90° 180°]) are modified so that they are face to face. We are considering  $P = 5$  channels as it is a realistic situation for home applications; enough for obtaining good spatial localization and significantly more complex than the stereo case.

The source is a male speech recorded in an anechoic chamber at a sampling rate of 16 kHz and the background noise in the local room has a power of -40 dB of SNR.

Figure 4 shows the constrained PBFDAF algorithm behaviour. For equation (8) we are using a power normalizing expression as

$$\mu(m) = \frac{\mu}{\mathbf{U}(m)+\delta} , \tag{22}$$

$$\mathbf{U}(m) = (1 - \lambda) \mathbf{U}(m - 1) + \lambda |\mathbf{X}|^2 . \tag{23}$$

Where  $\mu(m)$  is a matrix of dimensions  $M \times Q \times P$ ,  $\mu$  is the step size,  $\lambda$  is an averaging factor, and  $\delta$  is a constant to avoid stability problems. In our case  $\mu = 0.025$ ,  $\lambda = 0.25$  and  $\delta = 0.5$ .

Figure 5 shows the result of using the PBFDAF-CG algorithm with the Hestenes-Stiefel method where the difference in convergence can be observed. A maximum of  $N = \lceil \sqrt{K} \rceil$  or when MSE below -45 dB is employed.

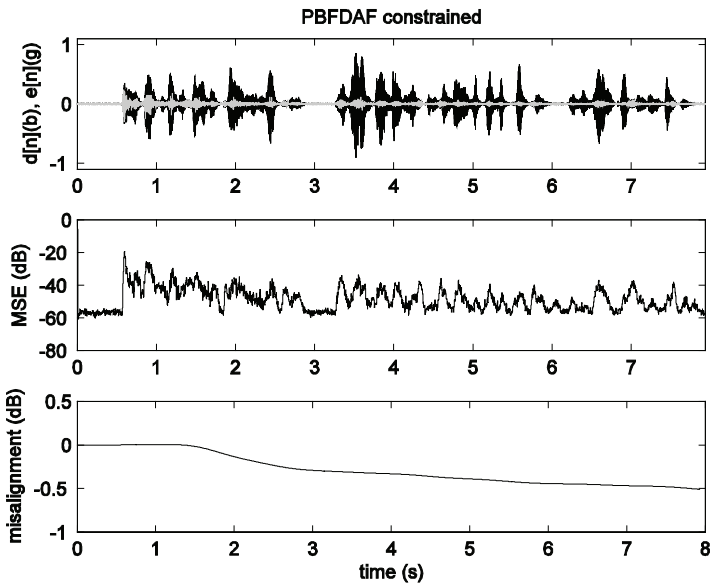


Fig. 4. PBFDAF Constrained.

For both algorithms we use  $Q = 8$  partitions,  $L = 1024$  taps,  $K = L/Q = 128$  taps for each partition. The length of the FFTs is  $M = 2K = 256$ . Working with sample rate of 16 kHz means 8 ms of latency (although a delayless approach already has been studied) [3]. Again in both cases the algorithm uses the overlap-save method (50% overlapping).

The upper part of the figures show the echo signal  $d$  (black) and the residual error  $e$  (grey). The centre shows the MSE (dB) and the lower picture the misalignment (also in dB) obtained as  $\epsilon = \|\mathbf{h} - \mathbf{w}\| / \|\mathbf{h}\|$ , being  $\mathbf{h}$  the unknown impulse response and  $\mathbf{w} = [w_1^T \cdots w_P^T]^T$  the estimation.

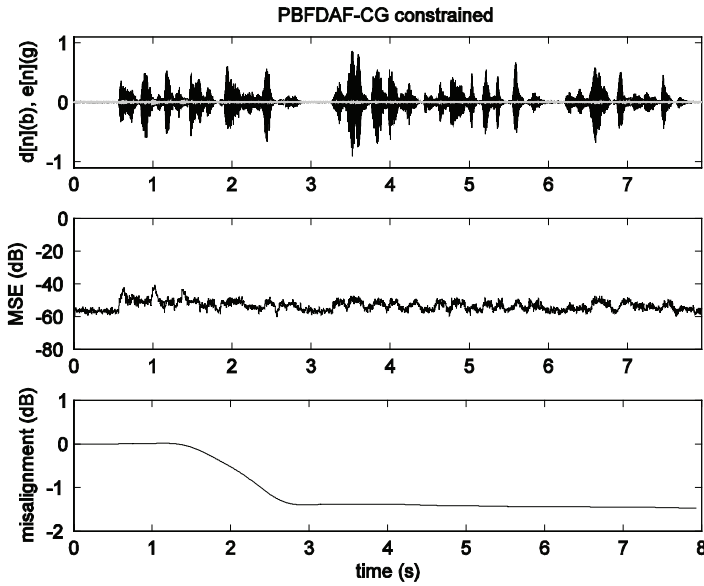


Fig. 5. PBFDAF-CG Constrained.

The speech input signal to MAEC application is an inappropriate perturbation signal due to a nonstationary character. The speech waveform contains segments of voiced (quasi-periodic) sounds, such as “e”, unvoiced or fricative (noiselike) sounds, such as “g”, and silence.

Besides it is possible a double-talk situations (when the speech of the at least two talkers arrives simultaneously at the canceller) that made identification much more problematic than it might appear at first glance.

A much more conditioned application is an adaptive multichannel measure of impulse response. In this case, it is possible to select the best perturbation signal, with the appropriate SNR, for system identification and adapt until the error signal falls below a MSE setting threshold.

The maximum length sequences (MLS) are pseudorandom binary signals which autocorrelation function is approximately an impulse.

In an industrial case it is probably the most convenient method to use because it is simple and allows system identification without perturbing the system operation or stopping the plant [1]. In this case it is necessary superimpose the perturbation signal to the input system with a power enough to identify the system while guaranty the optimal functioning.

## 6 Conclusions

The PBFDAF algorithm is widely used in multichannel adaptive filtering applications such as MAEC commercial systems with good results (in general for stereo case).

However, especially when working in multichannel, high reverberation environments (like teleconference) its convergence may not be fast enough. In this article we

have presented a novel algorithm: PFDFAF-CG; based on the same structure, but using much more powerful CG techniques to speed up the convergence time and improve the MSE and misalignment performance.

As shown in the results, the proposed algorithm improves a MSE and misalignment performance, and converges a lot faster than its counterpart while keeping the computational convergence relatively low, because all the operations are performed between vectors in the frequency-domain.

We are working on better gradient estimation methods in order to reduce computational cost. Besides, it is possible to arrive to a compromise between complexity and speed modifying the maximum number of iterations.

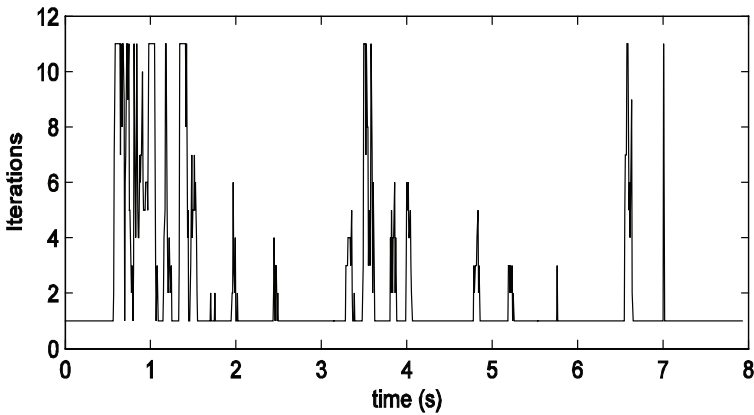


Fig. 6. PBFDAF-CG iterations versus time.

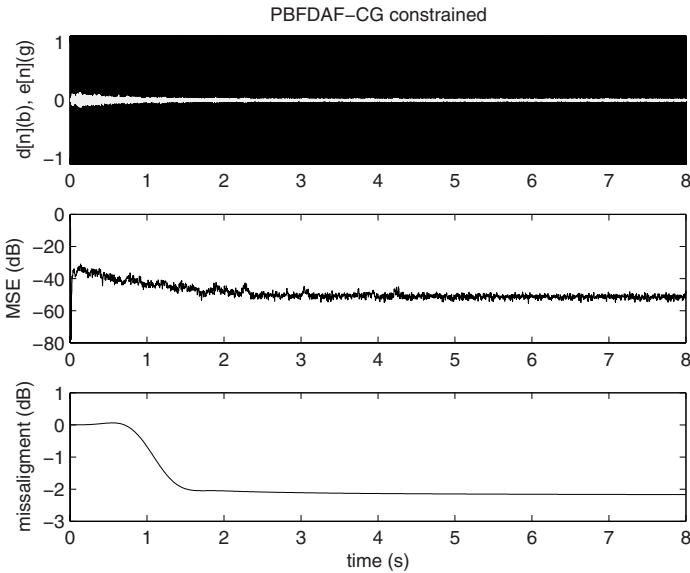
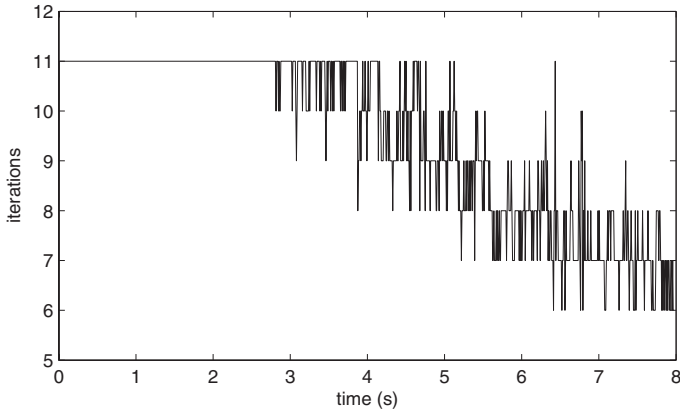


Fig. 7. PBFDAF-CG Constrained (MLS).



**Fig. 8.** PBFDAF-CG iterations versus time (MLS).

Figure 6 shows the PBFDAF-CG iterations versus time. The total number of iterations for this experiment is 992 for PBFDAF and 1927 for PBFDAF-CG (80 times higher computational cost).

Figure 7 shows the result of PBFDAF-CG with MLS source (identical settings) and Fig. 8 the iterations versus time. Notice that more uniform MSE convergence and best misalignment. The computational cost decrease while time the increases. A better performance is possible increasing the SNR and diminishing the MSE level threshold.

## References

1. Aguado, A., Martnez, M.: *Identificacin y Control Adaptativo*. Prentice Hall (2003).
2. Allen, J.B., Berkley, D.A.: Image method for efficiently simulating small-room acoustics. In *J.A.S.A.* 65 (1979) 943–950.
3. Bendel, Y., Burshtein, D.: Delayless Frequency Domain Acoustic Echo Cancellation. In *IEEE Transactions on Speech and Audio Processing*, 9:5 (2001) 589–587.
4. Benesty, J., Huang, Y. (Eds.): *Adaptive Signal Processing: Applications to Real-World Problems*, Springer (2003).
5. Boray, G., Srinath, M.D.: Conjugate Gradient Techniques for Adaptive Filtering. In *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Application*, 39:1 (1992) 1–10.
6. Luenberger, D.G.: *Introduction to Linear and Nonlinear Programming*, MA: Addison-Wesley, Reading, Mass (1984).
7. Shink, J.: Frequency-Domain and Multirate Adaptive Filtering. In *IEEE Signal Processing Magazine*, 9:1 (1992) 15–37.
8. Páez Borrallo, J., García Otero, M.: On the implementation of a partitioned block frequency-domain adaptive filter (PBFDAF) for long acoustic echo cancellation. In *Signal Processing*, 27 (1992) 301–315.

## Appendix

The “conjugacy” relation  $\mathbf{v}_i^H \mathbf{R} \mathbf{v}_j = 0, \forall i \neq j$  means that two vectors,  $\mathbf{v}_i$  and  $\mathbf{v}_j$ , are orthogonal with respect to any symmetric positive matrix  $\mathbf{R}$ . This can be looked upon as a generalization of the orthogonality, for which  $\mathbf{R}$  is the unity matrix. The best way to visualize the working of conjugate directions is by comparing the space we are working in with a “stretched” space.

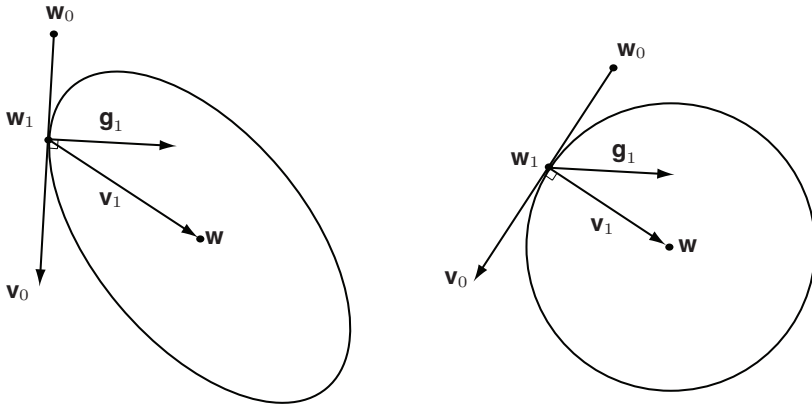


Fig. 9. Optimality of CG method.

The SD methods are slow due to the successive gradient orthogonality that results of minimize the recursive updating equation (8) respect to  $\mu(m)$ . The movement toward a minimum has the zigzag form. The left part in Fig. 9 shows the quadratic function contours in a real space (for  $r \neq \mathbf{0}$  in (2) are elliptical). Any pair of vectors that appear perpendicular in this space would be orthogonal. The right part shows the same drawing in a space that is stretched along the eigenvector axes so that the elliptical contours from the left part become circular. Any pair of vectors that appear to be perpendicular in this space is in fact  $\mathbf{R}$ -orthogonal. The search for a minimum of the quadratic function starts at  $\mathbf{w}_0$ , and takes a step in the direction  $\mathbf{v}_0$  and stops at the point  $\mathbf{w}_1$ . This is a minimum point along that direction, determined in the same way for SD method. While the SD method would search in the direction  $\mathbf{g}_1$ , the CG method would chose  $\mathbf{v}_1$ . In this stretched space, the direction  $\mathbf{v}_0$  appears to be a tangent to the now circular contours at the point  $\mathbf{w}_1$ . Since the next search direction  $\mathbf{v}_1$  is constrained to be  $\mathbf{R}$ -orthogonal to the previous, they will appear perpendicular in this modified space. Hence,  $\mathbf{v}_1$  will take us directly to the minimum point of the quadratic function ( $2^{\text{nd}}$  order in the example).

# Guaranteed Characterization of Capture Basins of Nonlinear State-Space Systems

Nicolas Delanoue<sup>1</sup>, Luc Jaulin<sup>2</sup>, Laurent Hardouin<sup>1</sup> and Mehdi Lhommeau<sup>1</sup>

<sup>1</sup> Laboratoire d'Ingénierie des Systèmes Automatisés  
Université d'Angers, 62 av. Notre Dame du Lac, 49000 Angers, France  
{nicolas.delanoue, mehdi.lhommeau,  
laurent.hardouin}@univ-angers.fr

<sup>2</sup> ENSIETA, 2 rue François Verny, 29806 Brest Cédex 09, France  
luc.jaulin@ensieta.fr

**Abstract.** This paper proposes a new approach to solve the problem of computing the capture basin  $\mathbb{C}$  of a target  $\mathbb{T}$ . The capture basin corresponds to the set of initial states such that the target is reached in finite time before possibly leaving of constrained set. We present an algorithm, based on interval analysis, able to characterize an inner and an outer approximation  $\mathbb{C}^- \subset \mathbb{C} \subset \mathbb{C}^+$  of the capture basin. The resulting algorithm is illustrated on the Zermelo problem.

## 1 Introduction

The purpose of this paper is to present an algorithm based on guaranteed numerical computation which, given the dynamics of the system, provides an inner and outer approximation of the capture basin. We recall some definitions and notations related to capture basin. In the sequel, we consider nonlinear continuous-time dynamical systems of the form

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{x}(0) = \mathbf{x}_0, \end{cases} \quad (1)$$

where  $\mathbf{x} \in \mathbb{R}^n$  is the state of the system with initial condition  $\mathbf{x}_0$  at  $t = 0$  and  $\mathbf{u} \in \mathbb{R}^m$  is the control vector. We shall assume that the function  $\mathbf{f}$  is sufficiently regular to guarantee that for all piecewise continuous function  $\mathbf{u}(\cdot)$  the solution of (1) is unique. The state vector  $\mathbf{x}(t)$  is not allowed to exit a given compact set  $\mathbb{K} \subset \mathbb{R}^n$  and the input  $\mathbf{u}(t)$  should belong to a given compact set  $\mathbb{U} \subset \mathbb{R}^m$ .

We define the flow (see [1])  $\phi^t(\mathbf{x}_0, \mathbf{u})$  as the solution of (1) for the initial vector  $\mathbf{x}_0$  and for the input function  $\mathbf{u}(\cdot)$ . The path from  $t_1$  to  $t_2$  is defined by

$$\phi^{[t_1, t_2]}(\mathbf{x}_0, \mathbf{u}) \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathbb{R}^n \mid \exists t \in [t_1, t_2], \mathbf{x}(t) = \phi^t(\mathbf{x}_0, \mathbf{u})\}. \quad (2)$$

Define a target set  $\mathbb{T} \subset \mathbb{K} \subset \mathbb{R}^n$  as a closed set we would like to reach for one  $t \geq 0$ . The *capture basin*  $\mathbb{C}$  of  $\mathbb{T}$  is the set

$$\mathbb{C} \triangleq \{\mathbf{x}_0 \in \mathbb{K} \mid \exists t \geq 0, \exists \mathbf{u}(\cdot) \in \mathcal{F}([0, t] \rightarrow \mathbb{U}), \phi^t(\mathbf{x}_0, \mathbf{u}) \in \mathbb{T} \\ \text{and } \phi^{[0, t]}(\mathbf{x}_0, \mathbf{u}) \subset \mathbb{K}\}, \quad (3)$$

where  $\mathcal{F}([0, t] \rightarrow \mathbb{U})$  represents the set of piecewise continuous functions from  $[0, t] \rightarrow \mathbb{U}$ . Then,  $\mathbb{C}$  is the set of initial states  $\mathbf{x} \in \mathbb{K}$  for which there exists an admissible control  $\mathbf{u}$ , and a finite time  $t \geq 0$  such that the trajectory  $\phi^{[0,t]}(\mathbf{x}_0, \mathbf{u})$  with the dynamic  $\mathbf{f}$  under the control  $\mathbf{u}$  lives in  $\mathbb{K}$  and reaches  $\mathbb{T}$  at time  $t$ .

The aim of the paper is to provide an algorithm able to compute an inner and an outer approximation of  $\mathbb{C}$ , i.e., to find two subsets  $\mathbb{C}^-$  and  $\mathbb{C}^+$  such that

$$\mathbb{C}^- \subset \mathbb{C} \subset \mathbb{C}^+.$$

Our contribution is twofold. First, in Section 2, we shall introduce interval analysis in the context of capture basin problem [2–4]. Second, in Section 3, we shall provide the first algorithm able to compute a guaranteed inner and an outer approximation for capture basins. The efficiency of our approach will be illustrated on the Zermelo problem in Section 4. Section 5 will then conclude the paper.

## 2 Interval Analysis

The interval theory was born in the 60’s aiming rigorous computations using finite precision computers (see [5]). Since its birth, it has been developed and it proposed today original algorithms for solving problems independently to the finite precision of computers computations, although reliable computations using finite precision remains one important advantage of the interval based algorithms [6].

An *interval*  $[x]$  is a closed and connected subset of  $\mathbb{R}$ . A *box*  $[\mathbf{x}]$  of  $\mathbb{R}^n$  is a Cartesian product of  $n$  intervals. The set of all boxes of  $\mathbb{R}^n$  is denoted by  $\mathbb{IR}^n$ . Note that  $\mathbb{R}^n = ]-\infty, \infty[ \times \dots \times ]-\infty, \infty[$  is an element of  $\mathbb{IR}^n$ . Basic operations on real numbers or vectors can be extended to intervals in a natural way.

*Example 1.* If  $[t] = [t_1, t_2]$  is an interval and  $[\mathbf{x}] = [x_1^-, x_1^+] \times [x_2^-, x_2^+]$  is a box, then the product  $[t] * [\mathbf{x}]$  is defined as follows

$$[t_1, t_2] * \begin{pmatrix} [x_1^-, x_1^+] \\ [x_2^-, x_2^+] \end{pmatrix} = \begin{pmatrix} [t_1, t_2] * [x_1^-, x_1^+] \\ [t_1, t_2] * [x_2^-, x_2^+] \end{pmatrix} = \begin{pmatrix} [\min(t_1x_1^-, t_1x_1^+, t_2x_1^-, t_2x_1^+), \max(t_1x_1^-, t_1x_1^+, t_2x_1^-, t_2x_1^+)] \\ [\min(t_1x_2^-, t_1x_2^+, t_2x_2^-, t_2x_2^+), \max(t_1x_2^-, t_1x_2^+, t_2x_2^-, t_2x_2^+)] \end{pmatrix}.$$

### 2.1 Inclusion Function

The function  $[\mathbf{f}](\cdot) : \mathbb{IR}^n \rightarrow \mathbb{IR}^p$  is an *inclusion function* of a function  $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^p$  if

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathbf{f}([\mathbf{x}]) \triangleq \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\} \subset [\mathbf{f}]([\mathbf{x}]).$$



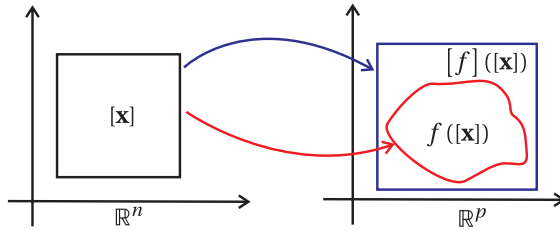


Fig. 1. Illustration of inclusion function.

Interval computation makes it possible to obtain inclusion functions of a large class of nonlinear functions, as illustrated by the following example.

Example 2. If  $f(x_1, x_2) \triangleq ((1 - 0.01x_2)x_1; (-1 + 0.02x_1)x_2)$ , a methodology to obtain an enclosure of the image set  $f([10, 20], [40, 50])$  is as follows:

$$f \left( \begin{matrix} [40, 50] \\ [10, 20] \end{matrix} \right) \subset \left( \begin{matrix} (1 - 0.01 * [40, 50]) * [10, 20] \\ (-1 + 0.02 * [10, 20]) * [40, 50] \end{matrix} \right) = \left( \begin{matrix} (1 - [0.4, 0.5]) * [10, 20] \\ (-1 + [0.2, 0.4]) * [40, 50] \end{matrix} \right) = \left( \begin{matrix} [0.5, 0.6] * [10, 20] \\ [-0.8, -0.6] * [40, 50] \end{matrix} \right) = \left( \begin{matrix} ([5, 12]) \\ ([-40, -24]) \end{matrix} \right).$$

This methodology can easily be applied for any box  $[x_1] \times [x_2]$  and the resulting algorithm corresponds to an inclusion function for  $f$ .

The interval union  $[x] \sqcup [y]$  of two boxes  $[x]$  and  $[y]$  is the smallest box which contains the union  $[x] \cup [y]$ . The width  $w([x])$  of a box  $[x]$  is the length of its largest side.

The  $\varepsilon$ -inflation of a box  $[x] = [x_1^-, x_1^+] \times \dots \times [x_n^-, x_n^+]$  is defined by

$$\text{inflate}([x], \varepsilon) \triangleq [x_1^- - \varepsilon, x_1^+ + \varepsilon] \times \dots \times [x_n^- - \varepsilon, x_n^+ + \varepsilon]. \quad (4)$$

### 2.2 Picard Theorem

Interval analysis for ordinary differential equations were introduced by Moore [5] (See [7] for a description and a bibliography on this topic). These methods provide numerically reliable enclosures of the exact solution of differential equations. These techniques are based on Picard Theorem.

**Theorem 1.** Let  $t_1$  be a positive real number. Assume that  $\mathbf{x}(0)$  is known to belong to the box  $[\mathbf{x}](0)$ . Assume that  $\mathbf{u}(t) \in [\mathbf{u}]$  for all  $t \in [0, t_1]$ . Let  $[\mathbf{w}]$  be a box (that is expected to enclose the path  $\mathbf{x}(\tau), \tau \in [0, t_1]$ ). If

$$[\mathbf{x}](0) + [0, t_1] * [\mathbf{f}]([\mathbf{w}], [\mathbf{u}]) \subset [\mathbf{w}], \quad (5)$$

where  $[\mathbf{f}]([\mathbf{x}], [\mathbf{u}])$  is an inclusion function of  $\mathbf{f}(\mathbf{x}, \mathbf{u})$ , then, for all  $t \in [0, t_1]$

$$\mathbf{x}([0, t_1]) \subset [\mathbf{x}](0) + [0, t_1] * [\mathbf{f}]([\mathbf{w}], [\mathbf{u}]). \quad (6)$$

### 2.3 Interval Flow

**Definition:** The inclusion function of the flow is a function

$$[\phi] : \begin{cases} \mathbb{I}\mathbb{R} \times \mathbb{I}\mathbb{R}^n \times \mathbb{I}\mathbb{R}^m \rightarrow \mathbb{I}\mathbb{R}^n \\ ([t], [\mathbf{x}], [\mathbf{u}]) \rightarrow [\phi]([t], [\mathbf{x}], [\mathbf{u}]) \end{cases}$$

such that

$$\forall t \in [t], \forall \mathbf{x} \in [\mathbf{x}], \forall \mathbf{u} \in \mathcal{F}([0, t] \rightarrow [\mathbf{u}]), \phi(t, \mathbf{x}, \mathbf{u}) \in [\phi]([t], [\mathbf{x}], [\mathbf{u}])$$

Using Theorem 1, one can build an algorithm computing an enclosure  $[\mathbf{x}](t_2)$  for the path  $\mathbf{x}([t]) = \{\mathbf{x}(t), t \in [t]\}$  from an enclosure  $[\mathbf{x}]$  for  $\mathbf{x}(0)$ . The principle of this algorithm is illustrated by Figure 2.

---

**Algorithm 1:** Inclusion function  $[\phi]$ .

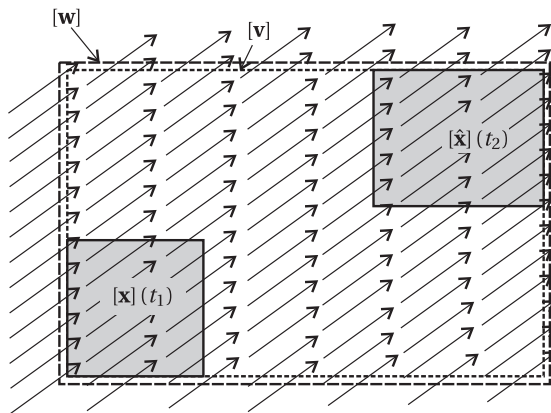
---

**Data:**  $[t] = [t_1, t_2], [\mathbf{x}](t_1), [\mathbf{u}]$   
**Result:**  $[\mathbf{x}](t_2), [\mathbf{w}]$

```

1 begin
2    $[\hat{\mathbf{x}}](t_2) := [\mathbf{x}](t_1) + (t_2 - t_1) * [\mathbf{f}]( [\mathbf{x}](t_1), [\mathbf{u}] );$ 
3    $[\mathbf{v}] := [\mathbf{x}](t_1) \sqcup [\hat{\mathbf{x}}](t_2);$ 
4    $[\mathbf{w}] := \text{inflate}([\mathbf{v}], \alpha.w([\mathbf{v}]) + \beta);$ 
5   if  $[\mathbf{x}](t_1) + [0, t_2 - t_1] * [\mathbf{f}]( [\mathbf{w}], [\mathbf{u}] ) \not\subseteq [\mathbf{w}]$  then
6      $[\mathbf{x}](t_2) := \mathbb{R}^n$ 
7     return
8    $[\mathbf{x}](t_2) := [\mathbf{x}](t_1) + (t_2 - t_1) * [\mathbf{f}]( [\mathbf{w}], [\mathbf{u}] );$ 
9 end
    
```

---



**Fig. 2.** Principle of algorithm  $[\phi]$ .

*Comments* : The interval  $[t] = [t_1, t_2]$  is such that  $t_1 \geq 0$ . Step 2 computes an estimation  $[\hat{\mathbf{x}}](t_2)$  for the domain of all  $\mathbf{x}(t_1)$  consistent with the fact that  $\mathbf{x}(0) \in [\mathbf{x}]$ . Note that, at this level, it is not certain that  $[\hat{\mathbf{x}}](t_2)$  contains  $\mathbf{x}(t_2)$ . Step 3 computes the smallest box  $[\mathbf{v}]$  containing  $[\mathbf{x}](t_1)$  and  $[\hat{\mathbf{x}}](t_2)$ . At Step 4,  $[\mathbf{v}]$  is inflated (see (4)) to provide a good candidate for  $[\mathbf{w}]$ .  $\alpha$  and  $\beta$  are small positive numbers. Step 5 checks the condition of Theorem 1. If the condition is not satisfied, no bounds can be computed for  $\mathbf{x}(t_2)$  and  $\mathbb{R}^n$  is returned. Otherwise, Step 8 computes a box containing  $\mathbf{x}(t_2)$  using theorem 1. ■

The algorithm to we gave to compute the interval flow is very conservative. The pessimism can drastically be reduced by using the Lohner method [8].

### 3 Algorithm

This section presents an algorithm to compute an inner and an outer approximation of the capture basin. It is based on Theorem 2.

**Theorem 2.** *If  $\mathbb{C}^-$  and  $\mathbb{C}^+$  are such that  $\mathbb{C}^- \subset \mathbb{C} \subset \mathbb{C}^+ \subset \mathbb{K}$ , if  $[\mathbf{x}]$  is a box and if  $\mathbf{u} \in \mathcal{F}([0, t] \rightarrow \mathbb{U})$ , then*

- (i)  $[\mathbf{x}] \subset \mathbb{T} \Rightarrow [\mathbf{x}] \subset \mathbb{C}$
- (ii)  $[\mathbf{x}] \cap \mathbb{K} = \emptyset \Rightarrow [\mathbf{x}] \cap \mathbb{C} = \emptyset$
- (iii)  $\phi(t, [\mathbf{x}], \mathbf{u}) \subset \mathbb{C}^- \wedge \phi([0, t], [\mathbf{x}], \mathbf{u}) \subset \mathbb{K} \Rightarrow [\mathbf{x}] \subset \mathbb{C}$
- (iv)  $\phi(t, [\mathbf{x}], \mathbb{U}) \cap \mathbb{C}^+ = \emptyset \wedge \phi(t, [\mathbf{x}], \mathbb{U}) \cap \mathbb{T} = \emptyset \Rightarrow [\mathbf{x}] \cap \mathbb{C} = \emptyset$

*Proof.* (i) and (ii) are due to the inclusion  $\mathbb{T} \subset \mathbb{C} \subset \mathbb{K}$ . Since  $\mathbb{T} \subset \mathbb{C}^- \subset \mathbb{C}$ , (iii) is a consequence of the definition of the capture basin (see (3)). The proof of (iv) is easily obtained by considering (3) and in view of fact that  $\mathbb{C} \subset \mathbb{C}^+ \subset \mathbb{K}$ .

Finally, a simple but efficient bisection algorithm is then easily constructed. It is summarized in Algorithm 2. The algorithm computes both an inner and outer approximation of the capture basin  $\mathbb{C}$ . In what follows, we shall assume that the set  $\mathbb{U}$  of feasible input vectors is a box  $[\mathbf{u}]$ . The box  $[\mathbf{x}]$  to be given as an input argument for ENCLOSE should contain set  $\mathbb{K}$ .

*Comments.* Steps 4 and 7 uses Theorem 2, (i)-(iii) to inflate  $\mathbb{C}^-$ . Steps 5 and 8 uses Theorem 2, (ii)-(iv) to deflate  $\mathbb{C}^+$ .

where

- $\varepsilon$  : ENCLOSE stops the bisecting procedure when the precision is reached ;
- $\mathbb{C}^-$  : Subpaving (list of nonoverlapping boxes) representing an inner approximation of the capture basin, that is the boxes inside the capture basin  $\mathbb{C}$  ;
- $\mathbb{C}^+$  : Subpaving representing the outer approximation of the capture basin, that is the boxes outside  $\mathbb{C}$  and the boxes for which no conclusion could be reached ;

These subpavings provide the following bracketing of the solution set :

$$\mathbb{C}^- \subset \mathbb{C} \subset \mathbb{C}^+.$$

**Algorithm 2:** ENCLOSE.

---

**Data:**  $\mathbb{K}, \mathbb{T}, [\mathbf{x}], [\mathbf{u}]$   
**Result:**  $\mathbb{C}^-, \mathbb{C}^+$

```

begin
1   $\mathbb{C}^- \leftarrow \emptyset; \mathbb{C}^+ \leftarrow [\mathbf{x}]; \mathcal{L} \leftarrow \{[\mathbf{x}]\};$ 
2  while  $\mathcal{L} \neq \emptyset$  do
3      pop the largest box  $[\mathbf{x}]$  from  $\mathcal{L}$ ;
4      if  $[\mathbf{x}] \subset \mathbb{T}$  then
5           $\mathbb{C}^- \leftarrow \mathbb{C}^- \cup [\mathbf{x}];$ 
6      else if  $[\mathbf{x}] \cap \mathbb{K} = \emptyset$  then
7           $\mathbb{C}^+ \leftarrow \mathbb{C}^+ \setminus [\mathbf{x}];$ 
8      take  $t \geq 0$  and  $\mathbf{u} \in [\mathbf{u}]$ 
9      if  $[\phi](t, [\mathbf{x}], \mathbf{u}) \subset \mathbb{C}^-$  and  $[\phi]([0, t], [\mathbf{x}], \mathbf{u}) \subset \mathbb{K}$  then
10          $\mathbb{C}^- \leftarrow \mathbb{C}^- \cup [\mathbf{x}];$ 
11     else if  $[\phi](t, [\mathbf{x}], \mathbf{u}) \cap \mathbb{C}^+ = \emptyset$  and  $[\phi](t, [\mathbf{x}], \mathbf{u}) \cap \mathbb{T} = \emptyset$  then
12          $\mathbb{C}^+ \leftarrow \mathbb{C}^+ \setminus [\mathbf{x}];$ 
13     else if  $w([\mathbf{x}]) \geq \varepsilon$  then
14         bisect  $[\mathbf{x}]$  and store the two resulting boxes into  $\mathcal{L}$ ;
end

```

---

## 4 Experimentations

This section presents an application of Algorithm 2. The algorithm has been implemented in C++ using Profil/BIAS interval library and executed on a PentiumM 1.4Ghz processor. As an illustration of the algorithm we consider the Zermelo problem [9, 10]. In control theory, Zermelo has described the problem of a boat which wants to reach an island from the bank of a river with strong currents. The magnitude and direction of the currents are known as a function of position. Let  $f(x_1, x_2)$  be the water current of the river at position  $(x_1, x_2)$ . The method for computing the expression of the speed vector field of two dimensional flows can be found in [11]. In our example the dynamic is nonlinear,

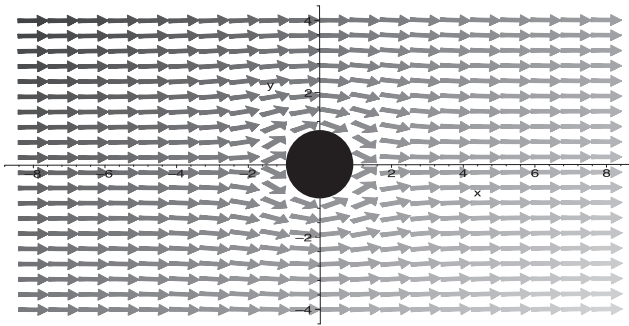
$$f(x_1, x_2) \triangleq \left( 1 + \frac{x_2^2 - x_1^2}{(x_1^2 + x_2^2)^2}, \frac{-2x_1x_2}{(x_1^2 + x_2^2)^2} \right).$$

The speed vector field associated to the dynamic of the currents is represented on Figure 3.

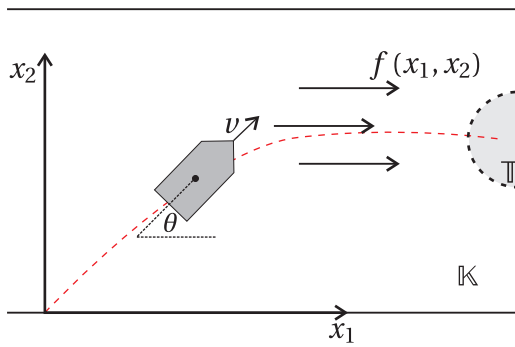
Let  $\mathbb{T} \triangleq \mathcal{B}(0, r)$  with  $r = 1$  be the island and we set  $\mathbb{K} = [-8, 8] \times [-4, 4]$ , where  $\mathbb{K}$  represents the river. The boat has his own dynamic. He can sail in any direction at a speed  $v$ . Figure 4 presents the two-dimensional boat. Then, the global dynamic is given by

$$\begin{cases} x_1'(t) = 1 + \frac{x_2^2 - x_1^2}{(x_1^2 + x_2^2)^2} + v \cos(\theta) \\ x_2'(t) = \frac{-2x_1x_2}{(x_1^2 + x_2^2)^2} + v \sin(\theta) \end{cases},$$

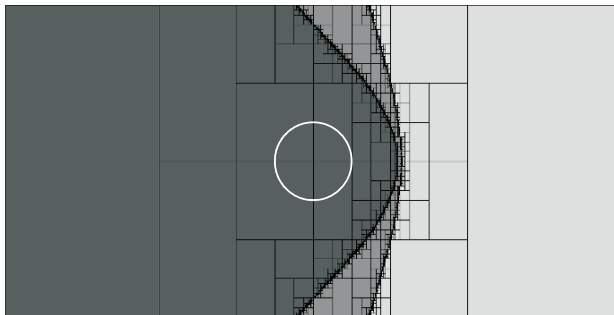
where the controls  $0 \leq v \leq 0.8$  and  $\theta \in [-\pi, \pi]$ .



**Fig. 3.** Vector field of the currents.



**Fig. 4.** Zermelo's problem.



**Fig. 5.** Two dimensional example of ENCLOSE algorithm.

Figure 5 shows the result of the ENCLOSE algorithm, where the circle delimits the border of the target  $\mathbb{T}$ . Then,  $\mathbb{C}^-$  corresponds to the union of all dark grey boxes and  $\mathbb{C}^+$  corresponds to the union of both grey and light grey boxes. Thus, we have the following inclusion relation :

$$\mathbb{C}^- \subset \mathbb{C} \subset \mathbb{C}^+.$$

## 5 Conclusions

In this paper, a new approach to deal with capture basin problems is presented. This approach uses interval analysis to compute an inner and an outer approximation of the capture basin for a given target. To fill out this work, different perspectives appear. It could be interesting to tackle problems in significantly larger dimensions. The limitation is mainly due to the bisections involved in the interval algorithms that makes the complexity exponential with respect to the number of variables. Constraint propagation techniques [12] make it possible to push back this frontier and to deal with high dimensional problems (with more than 1000 variables for instance). In the future, we plan to combine our algorithm with graph theory and guaranteed numerical integration [7, 13] to compute a guaranteed control  $\mathbf{u}$ .

## References

1. Hirsch, M. W., Smale, S.: *Differential Equations, Dynamical Systems, and Linear Algebra*. ap, San Diego (1974)
2. Aubin, J.: *Viability theory*. Birkhuser, Boston (1991)
3. Saint-Pierre, P.: Approximation of the viability kernel. *Applied Mathematics & Optimization* 29 (1994) 187-209
4. Cruck, E., Moitie, R., Seube, N.: Estimation of basins of attraction for uncertain systems with affine and lipschitz dynamics. *Dynamics and Control* 11(3) (2001) 211-227
5. Moore, R.E.: *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ (1966)
6. Kearfott, R. B., Kreinovich, V., eds.: *Applications of Interval Computations*. Kluwer, Dordrecht, the Netherlands (1996)
7. Nedialkov, N. S., Jackson, K. R., Corliss, G. F.: Validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation* 105 (1999) 21-68
8. Lohner, R.: Enclosing the solutions of ordinary initial and boundary value problems. In Kaucher, E., Kulisch, U., Ullrich, C., eds.: *Computer Arithmetic: Scientific Computation and Programming Languages*. BG Teubner, Stuttgart, Germany (1987) 255-86
9. Bryson, A.E., Ho, Y.C.: *Applied optimal control: optimization, estimation, and control*. Halsted Press (1975)
10. Cardaliaguet, P., Quincampoix, M., Saint-Pierre, P.: Optimal times for constrained nonlinear control problems without local controllability. *Applied Mathematics and Optimization* 36 (1997) 21-42
11. Batchelor, G.K.: *An introduction to fluid dynamics*. Cambridge university press(2000)
12. L. Jaulin, M. Kieffer, O. Didrit, E. Walter: *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London (2001)
13. Delanoue, N.: *Algorithmes numériques pour l'analyse topologique*. PhD dissertation, Université d'Angers, ISTIA, France (decembre 2006) Available at [www.istia.univ-angers.fr/~delanoue/](http://www.istia.univ-angers.fr/~delanoue/).

# In Situ Two-Thermocouple Sensor Characterisation using Cross-Relation Blind Deconvolution with Signal Conditioning for Improved Robustness

Peter Hung<sup>1</sup>, Seán McLoone<sup>1</sup>, George Irwin<sup>2</sup>, Robert Kee<sup>2</sup> and Colin Brown<sup>2</sup>

<sup>1</sup> Department of Electronic Engineering, National University of Ireland Maynooth  
Maynooth, Co. Kildare, Ireland

{phung, sean.mcloone}@eeng.nuim.ie

<sup>2</sup> Virtual Engineering Centre, Queen's University Belfast  
Belfast, BT9 5HN, Northern Ireland

{g.irwin, r.kee, cbrown17}@qub.ac.uk

**Abstract.** Thermocouples are one of the most widely used temperature measurement devices due to their low cost, ease of manufacture and robustness. However, their robustness is obtained at the expense of limited sensor bandwidth. Consequently, in many applications signal compensation techniques are needed to recover the true temperature from the attenuated measurements. This, in turn, necessitates *in situ* thermocouple characterisation. Recently the authors proposed a novel characterisation technique based on the cross-relation method of blind deconvolution applied to the output of two thermocouples simultaneously measuring the same temperature. This offers a number of advantages over competing methods including low estimation variance and no need for *a priori* knowledge of the time constant ratio. A weakness of the proposed method is that it yields biased estimates in the presence of measurement noise. In this paper we propose the inclusion of a signal conditioning step in the characterisation algorithm to improve the robustness to noise. The enhanced performance of the resulting algorithm is demonstrated using both simulated and experimental data.

**Keywords.** Sensor, system identification, thermocouple, blind deconvolution.

## 1 Introduction

In order to achieve high quality, low cost production with low environmental impact, modern industry is turning more and more to extensive sensing of processes and machinery, both for diagnostic purposes and as inputs to advanced control systems. Of particular interest in many applications is the accurate measurement of temperature transients in gas or liquid flows. For example, in an internal combustion engine the dynamics of the exhaust gas temperature is a key indicator of its performance as well as a valuable analytical input for on-board diagnosis of catalyst malfunction, while in the pharmaceutical industry precise control of transient temperatures is sometimes necessary in lyophilisers used in drug manufacture to ensure the quality and consistency of the final product. These and many other applications, thus require the availability of fast response temperature sensors.

Fast response temperature measurement can be performed using techniques such as Coherent Anti-Stokes Spectroscopy, Laser-Induced Fluorescence and Infrared Pyrometry [1], [2]. However, these are expensive, difficult to calibrate and maintain and are therefore impractical for wide-scale deployment outside the laboratory [2].

Thermocouples are widely used for temperature measurement due to their high permissible working limit and good linear temperature dependence. In addition, their low cost, robustness, ease of installation and reliability means that there are many situations in which thermocouples are indeed the only suitable choice. Unfortunately, their design involves a compromise between robustness and speed of response which poses major problems when measuring temperature fluctuations with high frequency signal components.

To remove the effect of the sensor on the measured quantity in such conditions, compensation of the thermocouple measurement is desirable. Usually, this compensation involves two stages: thermocouple characterisation followed by temperature reconstruction. Reconstruction is a process of restoring the unknown gas or fluid temperature from thermocouple outputs using either software techniques or hardware. This paper focuses on the first stage, since effective and reliable characterisation is essential for achieving satisfactory temperature reconstruction.

In an attempt to improve on existing thermocouple characterisation methods, the authors recently proposed a novel characterisation technique based on the cross-relation (CR) method of blind deconvolution [3] applied to the output of two thermocouples simultaneously measuring the same temperature [3], [4]. This offers a number of advantages over competing methods [2], [5], [6], [7] including low estimation variance and no need for *a priori* knowledge of the time constant ratio. However, a weakness of the proposed CR method is that it yields biased estimates in the presence of measurement noise [8]. This contrasts with its leading competitor, the Generalised Total Least Squares (GTLS) based difference equation characterisation algorithm [2], [9], which is an unbiased estimator but suffers from high estimation variance.

In this paper we propose a modification of the CR method that involves the inclusion of a signal conditioning step prior to the application of the CR algorithm, leading to improved robustness to measurement noise. The algorithm is validated using Monte Carlo simulations and data from an experimental test rig [10].

The remainder of the paper is organised as follows. The two-thermocouple characterisation methodology and the GTLS difference equation algorithm are introduced in Section 2. Section 3 provides an overview of the CR characterisation method and its principal characteristics. In Section 4 the CR implementation that incorporates signal conditioning filters is developed. The performance of this new algorithm is compared with the conventional CR implementation and the GTLS algorithm for both simulated and experimental data in Section 5. Finally, conclusions are presented in Section 6.



## 2 Difference Equation Sensor Characterisation

### 2.1 Thermocouple Modelling

Provided a number of criteria regarding thermocouple construction are satisfied [5] [6], a first-order lag model with time constant  $\tau$  and unity gain can represent the frequency response of a fine-wire thermocouple [11]. This simplified model can be written mathematically as

$$T_f(t) = T(t) + \tau \dot{T}(t). \quad (1)$$

Here the original liquid or gas flow temperature  $T_f$  can be reconstructed if  $\tau$ , the thermocouple output  $T(t)$  and its derivative are available. In practice, this direct approach is infeasible as  $T(t)$  contains noise and its derivative is difficult to estimate accurately. More importantly, it is generally not possible to obtain a reliable *a priori* estimate of  $\tau$ , related to their thermocouple bandwidth  $\omega_B$

$$\tau = \frac{1}{\omega_B}, \quad (2)$$

which, in turn, is a function of thermocouple wire diameter  $d$  and fluid velocity  $v$

$$\omega_B \propto \sqrt{\frac{v}{d^3}}. \quad (3)$$

Hence,  $\tau$  varies as a function of operating conditions. Clearly, a single thermocouple does not provide sufficient information for *in situ* estimation.

Equation (3) highlights the fundamental trade-off that exists when using thermocouples. Large wire diameters are usually employed to withstand harsh environments such as engine combustion systems, but these result in thermocouples with low bandwidth, typically  $\omega_B < 1$  Hz. In these situations high frequency temperature transients are lost with the thermocouple output significantly attenuated and phase-shifted compared to  $T_f$ . Consequently, appropriate compensation of the thermocouple measurement is needed to restore the high frequency fluctuations.

### 2.2 Two-Thermocouple Sensor Characterisation

In 1936 Pfried [12] suggested using two thermocouples with different time constants to obtain *in situ* sensor characterisation. Since then, various thermocouple compensation techniques incorporating this idea have been proposed in an attempt to achieve accurate and robust temperature compensation [2] [5] [6] [7] [13]. However, the performance of all these algorithms deteriorates rapidly with increasing noise power, and many are susceptible to singularities and sensitive to offsets [14].

Some of these two-thermocouple methods rely on the restrictive assumption that the ratio of the thermocouple time constants  $\alpha$  ( $\alpha < 1$  by definition) is known *a priori*. Hung *et al.* [2] [13] developed difference equation methods that do not require any *a priori* assumption about the time constant ratio. The equivalent discrete time representation for the thermocouple model (1) is:

$$T(k) = aT(k-1) + bT_f(k-1), \quad (4)$$

where  $a$  and  $b$  are difference equation ARX parameters and  $k$  is the sample instant. Assuming ZOHs and sampling interval  $\tau_s$ , the parameters of the discrete and continuous time thermocouple models are related by

$$a = \exp(-\tau_s/\tau), \quad b = 1 - a. \quad (5)$$

For two thermocouples we have

$$T_1(k) = a_1T_1(k-1) + b_1T_f(k-1) \quad \text{and} \quad (6)$$

$$T_2(k) = a_2T_2(k-1) + b_2T_f(k-1), \quad (7)$$

where subscripts 1 and 2 are used to distinguish between signals from different thermocouples. The discrete time equivalent of the time constant ratio  $\alpha$  is then defined as

$$\beta = b_2/b_1, \quad \beta < 1. \quad (8)$$

The unknown temperature  $T_f$  can be eliminated from the thermocouple models (6) and (7) to give an expression in terms of  $\beta$ ,  $b_2$  and the thermocouple outputs only [2] [15], that is:

$$\Delta T_2^k = \beta \Delta T_1^k + b_2 \Delta T_{12}^{k-1}, \quad (9)$$

where the pseudo-sensor output  $\Delta T_2^k$  and inputs  $\Delta T_1^k$  and  $\Delta T_{12}^{k-1}$  are defined as

$$\begin{aligned} \Delta T_1^k &= T_1(k) - T_1(k-1) \\ \Delta T_2^k &= T_2(k) - T_2(k-1) \\ \Delta T_{12}^{k-1} &= T_1(k-1) - T_2(k-1). \end{aligned} \quad (10)$$

For an  $M$ -sample data set (9) can be expressed in ARX vector form

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\theta}, \quad (11)$$

with  $\mathbf{Y} = \Delta \mathbf{T}_2^k$ ,  $\mathbf{X} = [\Delta \mathbf{T}_1^k \quad \Delta \mathbf{T}_{12}^{k-1}]$ , and  $\boldsymbol{\theta} = [\beta \quad b_2]^T$ . Here  $\Delta \mathbf{T}_1^k$ ,  $\Delta \mathbf{T}_2^k$  and  $\Delta \mathbf{T}_{12}^{k-1}$  are vectors containing  $M-1$  samples of the corresponding composite signals  $\Delta T_1^k$ ,  $\Delta T_2^k$  and  $\Delta T_{12}^{k-1}$ .

This characterisation model, referred to as the  $\beta$ -formulation, can be identified using least squares techniques. Due to the form of the composite input and output signals, the noise terms in the  $\mathbf{X}$  and  $\mathbf{Y}$  data blocks are not independent with the result that conventional least squares and total least squares both generate biased parameter estimates even when the measurement noise on the thermocouples is independent. However, by formulating identification as a generalised total least squares (GTLS) problem, unbiased parameter estimates can be obtained. The resulting  $\beta$ -GTLS algorithm is more robust than other difference equation formulations [15] and

provides superior performance to other two-thermocouple probe characterisation methods at low and medium noise levels [2].

Unfortunately, the variance of  $\beta$ -GTLS estimates grows rapidly with increasing noise level, particularly when compared with conventional least squares [2]. In addition, the approach occasionally returns unreasonable time constant estimates at high noise levels, as noted in [4], due to ill-conditioning [16] and the sensitivity of the relationship between time constants and discrete model parameters (5) in the vicinity of the singularity at  $b=0$ . The cross-relation blind deconvolution approach proposed in [3] [4] avoids these issues.

### 3 Blind Sensor Characterisation

One of the best known deterministic blind deconvolution approaches is the method of cross-relation (CR) proposed by Liu *et al.* [17]. Such techniques exploit the information provided by output measurements from multiple systems of known structure but unknown parameters, for the same input signal.

This new approach to characterisation of thermocouples is completely different from those in Section 2. As commutation is a fundamental assumption for the method of cross-relation, the thermocouple models are both assumed to be linear. This is reasonably realistic as long as the thermocouples concerned are used within well-defined temperature ranges. Nonetheless, linearisation can easily be carried out using either the data capture hardware or software, even if the thermocouple response is nonlinear. Further, the approach requires constant model parameters, therefore the fluid or gas flow velocity  $v$  is assumed to be constant, such that the two thermocouple time constants  $\tau_1$  and  $\tau_2$  are time-invariant.

#### 3.1 Two-Thermocouple Sensor Characterisation

By exploiting the commutative relationship between linear systems, a novel two-thermocouple characterisation scheme can be obtained as follows. Since the fluid temperature  $T_f$  is unknown, the two thermocouple output signals  $T_1$  and  $T_2$  are passed through two different synthetic thermocouples as shown in Fig. 1. These are also modelled by (1) and can be expressed in first-order transfer function form as:

$$\hat{H}_1(s) = \frac{1}{1 + s \hat{\tau}_1}, \quad \hat{H}_2(s) = \frac{1}{1 + s \hat{\tau}_2}, \quad (12)$$

where  $\hat{H}$  is the estimate of the thermocouple transfer function  $H$ . The unknown thermocouple time constant parameters can then be estimated as  $\hat{\tau}_1$  and  $\hat{\tau}_2$  using the cross-relation method, illustrated in Fig. 1. Here the cross-relation error signal,  $e = T_{12}(t) - T_{21}(t)$  is used to define a mean-square-error cost function

$$\begin{aligned} J_{\text{MSE}}(\hat{\tau}_1, \hat{\tau}_2) &= E[e^2] \\ &= E[(T_{12}(t) - T_{21}(t))^2], \quad \forall \hat{\tau}_1, \hat{\tau}_2. \end{aligned} \quad (13)$$

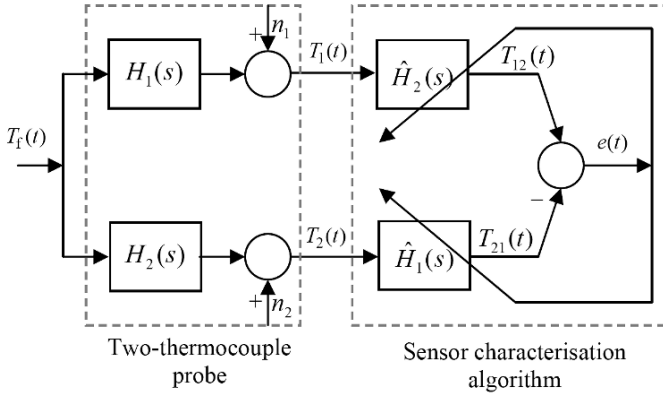


Fig. 1. Two-thermocouple cross-relation characterisation.

Equation (13) is then minimised with respect to  $\hat{\tau}_1$  and  $\hat{\tau}_2$  to yield the estimates of the unknown thermocouple time constants. Clearly, the cross-relation cost function  $J_{\text{MSE}}(\hat{\tau}_1, \hat{\tau}_2)$  is zero when  $\hat{\tau}_1 = \tau_1$  and  $\hat{\tau}_2 = \tau_2$ . In practice it will not be possible to obtain an exact match between  $T_{12}$  and  $T_{21}$  due to measurement noise and other factors such as thermocouple modelling inaccuracy and violations of the assumption that the two thermocouples are experiencing identical environmental conditions.

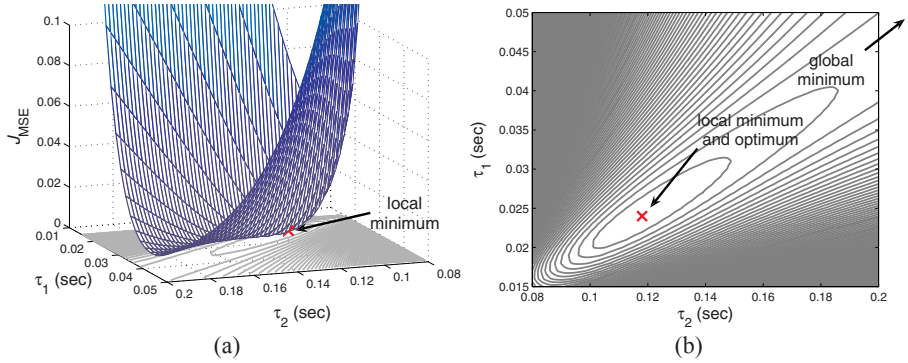
Xu *et al.* [18] suggest that one of the necessary conditions for multiple finite-impulse-response channels to be identifiable is that their transfer function polynomials do not share common roots. Applying this condition to the two-thermocouple characterisation problem corresponds to requiring that the time constants, and hence the diameters (3), of the thermocouples are different, that is

$$\tau_1 \neq \tau_2 \quad \Rightarrow \quad d_1 \neq d_2. \tag{14}$$

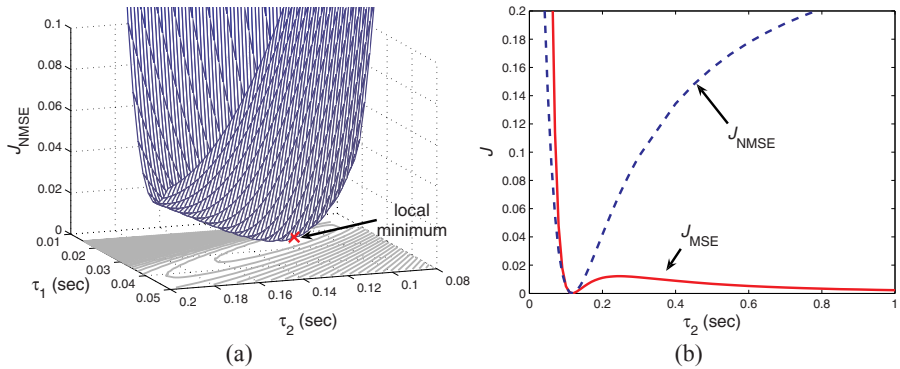
Not surprisingly, this requirement is consistent with all other two-thermocouple characterisation techniques mentioned in Section 2. Thus, cross-relation deconvolution converts the problem of sensor characterisation into an optimisation one.

### 3.2 Cost Function

A 3-D surface plot and a contour map of a typical  $J_{\text{MSE}}(\hat{\tau}_1, \hat{\tau}_2)$  cost function are shown in Fig. 2. Unfortunately,  $J_{\text{MSE}}(\hat{\tau}_1, \hat{\tau}_2)$  is not quadratic and cannot therefore be minimised using linear least squares. More importantly, the cost function has a second minimum when both time constant values approach infinity. Under these conditions, both low-pass filters (12) take infinite amounts of time to respond. In other words, they are effectively open-circuited and their differences will always be zero. The existence of this minimum applies regardless of the noise conditions or any violations of the modelling assumptions. The minimum at infinity is thus in fact the global minimum, while the true time constant value is located at a local minimum. In the absence of noise, it is noted that  $J_{\text{MSE}} = 0$  at both the global and local minima.



**Fig. 2.** A typical  $J_{MSE}$  cost function for noiseless thermocouple measurements: (a) 3-D plot of cost function; and (b) corresponding contour map.



**Fig. 3.** A typical  $J_{NMSE}$  cost function for noiseless thermocouple measurements: (a) 3-D plot; and (b) a comparison of 1-D cross sections of the MSE and NMSE CR cost functions.

The narrow basin of attraction of the desired local minimum coupled with the global minimum at infinity has serious implications for optimisation complexity since search bounds have to be carefully selected to avoid divergence of gradient search algorithms to the global minimum. Further, with increasing noise level the local minima becomes shallower and shallower, and eventually disappears causing the optimisation problem to become ill-posed.

As noted in [3] the ill-posed problem can be resolved by employing a normalised mean squared error (NMSE) cost function defined as

$$J_{NMSE}(\hat{\tau}_1, \hat{\tau}_2) = \frac{E[(T_{12}(t) - T_{21}(t))^2]}{0.5[\text{var}(T_{12}) + \text{var}(T_{21})]}. \quad (15)$$

A typical example of this cost function is plotted in Fig. 3(a). To highlight the effect of normalisation, the 1-D cross sections of both the MSE and NMSE cost functions along the line  $\hat{\tau}_1 = \alpha \hat{\tau}_2$  is also plotted in Fig. 3(b). Essentially, normalisation penalises large time constants, thereby eliminating the minimum at infinity giving a well conditioned convex cost function.

A weakness of the MSE and NMSE cross-relation algorithms is that they generate biased estimates. In fact, a statistical analysis of the algorithms [8] reveals that the MSE implementation yields postively biased estimates, while the NMSE implementation results in negatively biased estimates at high noise levels, though the latter is less significant when temperature variation is broadband.

### 4 Signal Conditioning

One approach to reducing the noise induced estimation bias is to introduce signal conditioning filters ( $F_c(s)$ ) prior to the CR characterisation algorithm as illustrated in Fig. 4. Provided the filters are identical, linear (thereby ensuring commutativity) and do not completely block the measured signals, the operation of the CR algorithm is unaffected. Within these constraints there is substantial freedom in the design of the filters.

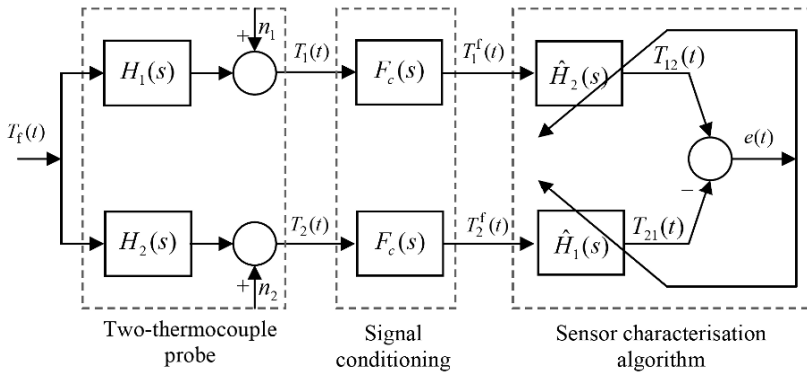


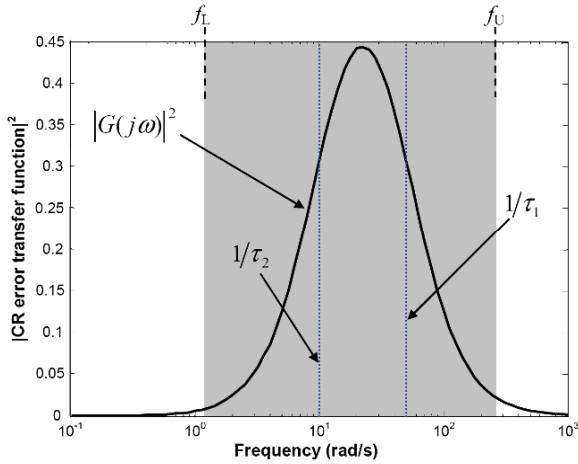
Fig. 4. Two-thermocouple cross-relation characterisation with signal conditioning

Assuming white measurement noise, which has a constant power spectrum profile across all frequencies, the obvious choice is to match the passband of the conditioning filters to the bandwidth of the temperature fluctuations. However, this is not the optimum choice, since it does not take into account the effect of the thermocouples. Consider the magnitude squared transfer function  $|G(j\omega)|^2$  from the input signal,  $T_i$ , to the cross-relation (CR) error signal,  $e$ , when  $\hat{H}_1 = \hat{H}_2 = 1$ , defined as

$$|G(j\omega)|^2 = \left| \frac{T_1(j\omega) - T_2(j\omega)}{T_i(j\omega)} \right|^2 = |H_1(j\omega) - H_2(j\omega)|^2. \tag{16}$$

This is plotted in Fig. 5 as a function frequency for a typical two-thermocouple sensor. As can be seen in Fig. 5,  $|G(j\omega)|^2$  has a peak between the thermocouple cut-off frequencies (i.e. between  $1/\tau_2$  and  $1/\tau_1$ ) and decays rapidly towards zero away from this peak. On the right hand side the decay is due to the increasing attenuation of the thermocouple signals at higher frequencies. On the left hand side, however, the

decay occurs because there is less and less difference between thermocouple signals while moving into the passband of the lowest bandwidth thermocouple (i.e.  $< 1/\tau_2$ ).



**Fig. 5.** Normalised Cross-relation error transfer function as a function of frequency for a two-thermocouple probe with time constants 0.02 and 0.1 seconds respectively.

The dynamic range of the CR error transfer function is approximately  $0.1/\tau_2$  to  $10/\tau_1$  rad/s. Thus, the effective CR error signal bandwidth will be limited to the intersection of the passband of  $G(j\omega)$  and the input signal bandwidth, and as such will, in general, be substantially less than the signal bandwidth. Consequently, for optimum signal-to-noise ratio performance the signal conditioning filters should be band-pass filters with a lower cut-off frequency,  $0 < f_L \leq \max(0.1/\tau_2, f_{\min})$  and an upper cut-off frequency,  $1/\tau_1 < f_U \leq \min(10/\tau_1, f_{\max})$ . Here,  $f_{\min}$  and  $f_{\max}$  are the minimum and maximum frequencies of the temperature fluctuations ( $T_f$ ). The maximum frequency  $f_{\max}$  is assumed to be greater than the bandwidth of the faster thermocouple ( $1/\tau_1$ ), otherwise signal compensation would not be required. In general, temperature fluctuations will be low-pass, in which case  $f_{\min}=0$  and  $f_{\max}$  corresponds to the signal bandwidth.

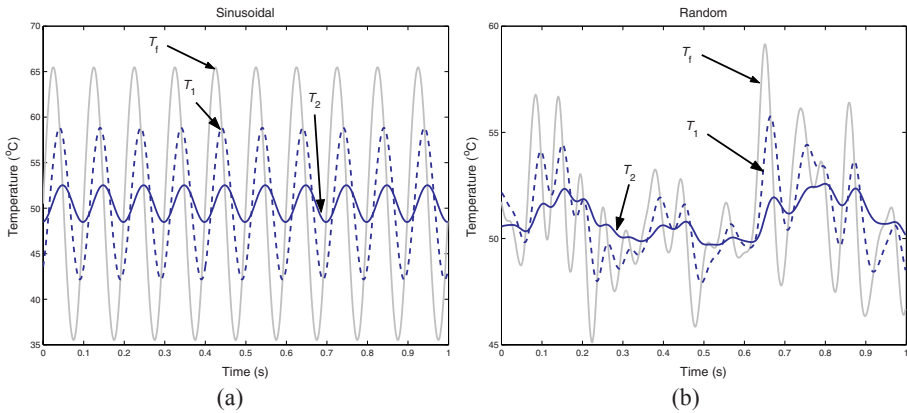
## 5 Performance Evaluation

To evaluate the performance of the proposed signal conditioned CR algorithm (SCCR) against conventional CR characterisation (CR) and the GTLS difference equation approach ( $\beta$ -GTLS), Monte Carlo simulations were performed using data from a two-thermocouple probe simulated in MATLAB® and an experimental test rig.

In the simulation the thermocouples were modelled as unity gain first-order low-pass filters with time constants  $\tau_1 = 23.8$  and  $\tau_2 = 116.8$  ms, respectively, and connected to a common input representing the fluctuating gas or liquid temperature signal. Data sets were generated for a sinusoidally varying temperature profile,

$$T_f(t) = 16.5 \sin(20\pi t) + 50.5, \quad (17)$$

and a band-limited white noise signal obtained by low-pass filtering the output of MATLAB's normally distributed random signal generator using a 125 rad/s bandwidth second-order Butterworth filter. Samples of each signal, along with the corresponding thermocouple measurements are given in Fig. 6. Each data set was recorded after initial condition transients had decayed and consisted of 5000 points at a sampling interval of 2 ms.



**Fig. 6.** Simulated temperature profiles: (a) sinusoidal; and (b) random band-limited to 20 Hz.

The test rig, depicted in Fig. 7, was specifically designed to produce periodic temperature fluctuations at constant fluid velocity [10]. It is supplied with air through a pressure regulator and a needle valve in order to obtain approximately constant mass flow rate. The flow is divided into two streams, one heated and the other at the supplied temperature. The streams are balanced using ball valves to ensure a uniform velocity profile across the air outlet. Both streams are then passed to isolated reservoirs before leaving their corresponding orifices. Finally, the warm and cool streams are combined in the mixing chamber before reaching the temperature probe. The frequency of periodic temperature fluctuations is controlled by the frequency of crank rotation that is connected to the rig via a linkage. The temperature probe consists of two thermocouples of unequal diameters (50 and 127  $\mu\text{m}$ ) and a constant-current thermal anemometer (3.8  $\mu\text{m}$ ) used to provide a reference temperature measurement. The gas velocity was measured using a pitot-static tube and a fast response pressure transducer which was fixed directly above the temperature measurement probe.

Using this test rig data was collected for periodic temperature fluctuations with a fundamental frequency of 38 rad/s at a sampling frequency of 1 kHz (Fig. 8(a)). Table 1 shows the time constant estimates obtained with each of the three characterisation methods. For comparison purposes, the best estimate of the time constants using the anemometer signal as an approximation to the true temperature is also included in the table. This essentially represents a lower bound on the true time constant values.



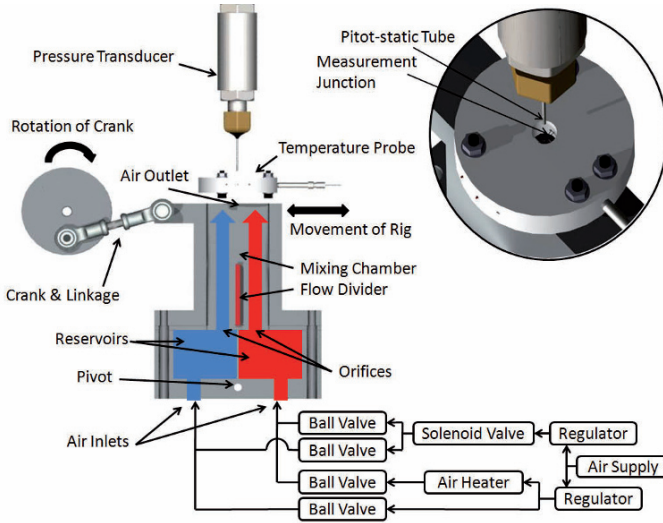


Fig. 7. Test rig schematic.

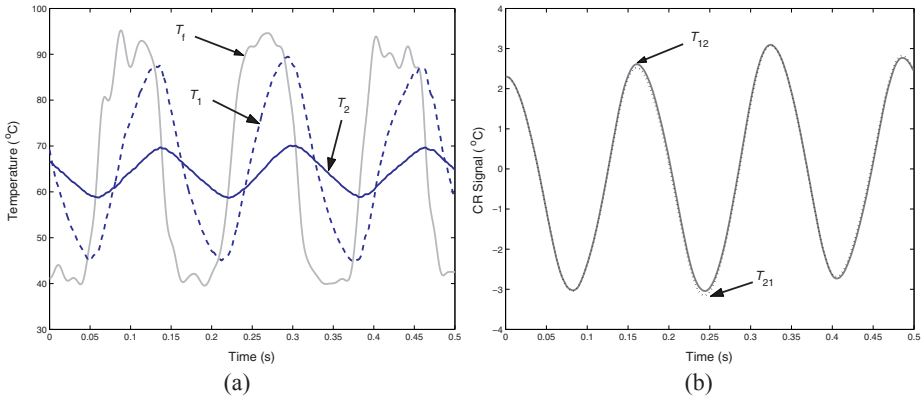


Fig. 8. Test rig data: (a) temperature profiles, and; (b) comparison of CR signals.

Table 1. Time constant estimates for test rig data.

Method	Ref. fit	$\beta$ -GTLS	CR	SCCR
$\tau_1$ estimate (ms)	37.9	37.4	38.0	37.7
$\tau_2$ estimate (ms)	187.9	185.3	187.4	185.8

As can be seen, the results are consistent across all methods, suggesting that the data has low noise contamination and is consistent with the two-thermocouple probe modelling assumptions. This is further confirmed by the close fit obtained between the CR signals ( $T_{12}$  and  $T_{21}$  from Fig. 1) as shown in Fig. 8(b). Consequently, for the purposes of the Monte Carlo simulations, the 500 point test rig data set is taken to be essentially noise free with  $\tau_1 \approx 38$  and  $\tau_2 \approx 188$  ms.

For each of the three data sets 100-run Monte Carlo simulations were performed for zero-mean white Gaussian measurement noise added to the noise free thermocouple outputs. The amount of noise added was quantified in terms of the noise level  $L_e$ , defined as

$$L_e = \sqrt{\frac{\text{var}(n_i)}{\text{var}(T_i)}} \cdot 100\%, \quad i = 1, 2, \quad (18)$$

where  $n_1$  and  $n_2$  are the noises added to the thermocouple measurements. For a given  $L_e$ , the performance of each characterisation algorithm was assessed in terms of the percentage error in estimating the time constants, that is:

$$e_{\tau_i} = \frac{\hat{\tau}_i - \tau_i}{\tau_i} \cdot 100\%, \quad i = 1, 2. \quad (19)$$

The means and standard deviations of this estimation error are recorded for a range of noise levels in Table 2 for each of the characterisation algorithms under consideration (SCCR, CR and  $\beta$ -GTLS). In SCCR the bandwidth of the conditioning filters was chosen as  $f_L=60, f_U=90$  rad/s for the sinusoidal data,  $f_L=5, f_U=120$  rad/s for the random data and  $f_L=25, f_U=125$  rad/s for the test rig data. Note that results for  $\hat{\tau}_2$  have been omitted as they show a similar pattern to those observed for  $\hat{\tau}_1$ .

## 6 Discussion and Conclusions

The results clearly show that the inclusion of signal conditioning filters has the desired effect. SCCR consistently has much lower bias than CR, particularly at higher noise levels. The picture for variance is less clear. SCCR estimates have slightly greater variance on average than the CR estimates for the simulated data, but substantially less variance in the case of the test rig data. This is currently the subject of further study.

While  $\beta$ -GTLS is theoretically unbiased, the variance in the estimates grows very rapidly with noise, and the algorithm essentially breaks down for  $L_e > 5$  in the simulated examples and  $L_e > 1$  for the test rig data. The substantially worse  $\beta$ -GTLS results in the last problem are due to the higher sample rate and fewer data points in this problem, both of which amplify the sensitivity of GTLS to noise. In contrast, CR and SCCR perform very well on this problem, though the estimation variance is significantly higher than in the simulated examples due to the smaller number of data points (500 compared to 5000). In practice, pre-filtering of the data can substantially improve the robustness of GTLS to noise at the expense of introducing some bias [2], but this has not been investigated here due to space constraints.

The cross-relation (CR) method of blind deconvolution provides an attractive framework for two-thermocouple sensor characterisation. It does not require *a priori* knowledge of the thermocouple time constant ratio  $\alpha$ , as required in many other characterisation algorithms, though this information can be exploited if available. CR is more noise-tolerant in the sense of reduced parameter estimation variance when compared to the alternatives such as  $\beta$ -GTLS. The standard CR implementation yields

biased estimates, but this is significantly reduced with the inclusion of signal conditioning filters. The resulting SCCR algorithm has been shown to be superior to other methods on both simulated and experimental data.

**Table 2.** Means and (standard deviations) of  $\hat{\tau}_1$  estimation errors (%) obtained with  $\beta$ -GTLS, CR and SCCR for each data set for a range of noise levels.

Noise Level ( $L_e$ )	1	3	5	7	10	15	20
<b>Sinusoidal simulation</b>							
$\beta$ -GTLS	-0.17 (0.69)	-0.77 (5.29)	-0.57 (13.90)	3.14 (26.93)	5.47 (60.83)	35.85 (318.50)	-6.43 (969.42)
CR	-0.13 (0.32)	-1.64 (1.01)	-3.95 (1.56)	-6.94 (2.39)	-12.10 (3.22)	-19.78 (4.29)	-26.3 (4.43)
SCCR	-0.07 (0.36)	-0.36 (1.02)	-0.57 (1.58)	-1.53 (2.41)	-2.57 (3.25)	-5.68 (4.70)	-9.53 (6.43)
<b>Random simulation</b>							
$\beta$ -GTLS	-0.01 (0.96)	0.33 (7.62)	1.27 (20.27)	5.23 (42.64)	17.12 (88.72)	49.57 (465.72)	98.31 (549.96)
CR	0.01 (0.21)	-0.34 (0.71)	0.73 (1.24)	1.55 (1.37)	2.87 (2.39)	5.27 (3.36)	10.18 (4.39)
SCCR	-0.04 (0.33)	0.22 (0.97)	-0.08 (1.54)	0.53 (2.28)	1.44 (3.07)	2.51 (4.80)	5.09 (6.60)
<b>Test rig</b>							
$\beta$ -GTLS	0.05 (9.76)	20.84 (89.11)	-127.97 (1237.54)	-180.31 (1028.05)	-281.23 (697.22)	-229.94 (604.96)	-168.53 (1000.33)
CR	-1.72 (1.29)	-1.08 (4.23)	-1.21 (6.12)	-1.83 (7.50)	-4.43 (11.50)	-3.35 (19.73)	-2.51 (31.30)
SCCR	-0.94 (1.03)	-0.98 (2.89)	-0.28 (4.51)	-0.59 (5.90)	-1.67 (9.68)	-1.87 (14.59)	1.57 (19.74)

## References

1. Kee, R. J., Blair, G. P.: Acceleration test method for a high performance two-stroke racing engine. In: SAE Motorsports Conference, Detroit, MI, Paper No. 942478 (1994)
2. Hung, P. C., McLoone, S., Irwin G., Kee, R.: A difference equation approach to two-thermocouple sensor characterisation in constant velocity flow environments. Rev. Sci. Instrum. 76, Paper No. 024902 (2005)
3. Hung, P. C., Kee, R. J., Irwin G. W., McLoone, S. F.: Blind Deconvolution for Two-Thermocouple Sensor Characterisation. ASME Dyn. Sys. Measure. Cont. 129, 194–202 (2007)
4. Hung, P. C., McLoone, S. F., Irwin, G. W., Kee, R. J.: Blind Two-Thermocouple Sensor Characterisation. In: International Conference on Informatics Control, Automation and Robotics (ICINCO 2007), Angers, France, pp.10–16 (2007)
5. Forney, L. J., Fralick G. C.: Two wire thermocouple: Frequency response in constant flow. Rev. Sci. Instrum. 65, 3252–3257 (1994)
6. Tagawa, M., Ohta, Y.: Two-Thermocouple Probe for Fluctuating Temperature Measurement in Combustion – Rational Estimation of Mean and Fluctuating Time Constants. Combust. and Flame. 109, 549–560 (1997)

7. Kee, R. J., O'Reilly, P. G., Fleck, R., McEntee, P. T.: Measurement of Exhaust Gas Temperature in a High Performance Two-Stroke Engine. SAE Trans. J. Engines. 107, Paper No. 983072 (1999)
8. McLoone, S. F., Hung, P. C., Irwin, G. W., Kee, R. J.: On the stability and biasedness of the cross-relation blind thermocouple characterisation method. In: IFAC World Congress 2008, Seoul, South Korea, accepted (2008)
9. McLoone, S., Hung, P., Irwin, G., Kee, R.: Difference equation sensor characterisation algorithms for two-thermocouple probes. Trans. InstMC, accepted (2008)
10. Brown, C., Kee, R. J., Irwin, G. W., McLoone, S. F., Hung, P.: Identification Applied to Dual Sensor Transient Temperature Measurement. In: UKACC Control 2008, Manchester, UK, submitted (2008)
11. Petit, C., Gajan, P., Lecordier, J. C., Paranthoen, P.: Frequency response of fine wire thermocouple. J. Phy. Part E. 15, 760–764 (1982)
12. Pfriem, H.: Zue messung verandelisher temperaturen von ogasen und flussigkeiten. Forsch. Geb. Ingenieurwes. 7, 85–92 (1936)
13. Hung, P., McLoone, S., Irwin G., Kee, R.: A Total Least Squares Approach to Sensor Characterisations. In: 13th IFAC Symposium on Sys. Id., Rotterdam, The Netherlands, pp. 337–342 (2003)
14. Kee, J. K., Hung, P., Fleck, B., Irwin, G., Kenny, R., Gaynor, J., McLoone, S.: Fast response exhaust gas temperature measurement in IC Engines. In: SAE 2006 World Congress, Detroit, MI, Paper No. 2006-01-1319 (2006)
15. McLoone, S., Hung, P., Irwin, G., Kee, R.: Exploiting A Priori Time Constant Ratio Information in Difference Equation Two-Thermocouple Sensor Characterisation. IEEE Sensors J. 6, 1627–1637 (2006)
16. Van Huffel S., Vandewalle, J.: The Total Least Squares Problem: Computational Aspects and Analysis, SIAM, Philadelphia, 1st edition (1991)
17. Liu, H., Xu, G., Tong, L.: A deterministic approach to blind identification of multichannel FIR systems. In: 27th Asilomar Conference on Signals, Systems and Computers, Asilomar, CA, pp. 581–584 (1993)
18. Xu, G., Liu, H., Tong, L., Kailath, T.: A least-squares approach to blind channel identification. IEEE Trans. Signal Proc. 43, 2982–2993 (1995)

# Dirac Mixture Approximation for Nonlinear Stochastic Filtering

Oliver C. Schrempf and Uwe D. Hanebeck

Intelligent Sensor-Actuator-Systems Laboratory  
Universität Karlsruhe (TH), Germany  
schrempf@ieee.org, uwe.hanebeck@ieee.org

**Abstract.** This work presents a filter for estimating the state of nonlinear dynamic systems. It is based on optimal recursive approximation the state densities by means of Dirac mixture functions in order to allow for a closed form solution of the prediction and filter step. The approximation approach is based on a systematic minimization of a distance measure and is hence optimal and deterministic. In contrast to non-deterministic methods we are able to determine the optimal number of components in the Dirac mixture. A further benefit of the proposed approach is the consideration of measurements during the approximation process in order to avoid parameter degradation.

## 1 Introduction

In this article, we present a novel stochastic state estimator for nonlinear dynamic systems suffering from system as well as measurement noise. The estimate is described by means of probability density functions. The problem that arises with the application of stochastic filters to nonlinear systems is that the complexity of the density representation increases and the exact densities cannot be calculated directly in general. Common solutions to this problem in order to build practical estimators can be divided into two classes. The approaches of the first class approximate or modify the system and measurement functions and apply a standard filter to this modified system. The idea of the second class is to approximate the resulting density functions themselves in order to calculate the filter steps in closed form.

A common representative of the first class is the extended Kalman filter (EKF). It is based on linearization of the system and measurement functions and applying a standard Kalman filter to this modified system. This approach is applicable to systems with negligible nonlinearities and additive noise, but fails in more general cases.

Another approach is to approximate the system together with its noise as a probabilistic model by means of a conditional density function. The application of adequate representations of the model like Gaussian mixtures with axis-aligned components [1], allows for efficient implementation of the filter steps.

Filters approximating the density functions instead of the system function can be divided into two main approaches found in the literature: i) sample-based density representations and ii) analytic density representations.

Sample-based filters like the popular particle filter [2] apply Monte Carlo methods for obtaining a sample representation. Since these samples are usually produced by a random number generator, the resulting estimate is not deterministic. Furthermore, Markov Chain Monte Carlo Methods (MCMC) are iterative algorithms that are unsuited for recursive estimation, hence, importance sampling like in [3] is often applied. The problem of sample degradation is usually tackled by bootstrap methods [4].

Other methods describe the probability density functions by means of their moments. A popular approach is the so called Unscented Kalman filter (UKF) [5] that uses the first moment and the second central moment for representing the densities. This allows for an efficient calculation of the update but fails in representing highly complex densities arising from nonlinear systems. Furthermore, the assumption of jointly Gaussian distributed states and measurements is made, which is not valid in general.

An approach that represents the state densities by means of Gaussian mixture density function is the so called Gaussian sum filter [6]. The Gaussian mixture representation allows for approximating arbitrary density functions, but finding the appropriate parameters is a tough problem. A more recent approach is the Progressive Bayes filter [7] which uses a distance measure for approximating the true densities. The key idea in this approach is to transform the approximation problem into an optimization problem. This is a major motivation for the approximation applied in the approach presented here.

The filter method we propose here follows the idea of approximating the density functions instead of the system itself, but the approximation is performed in a systematic manner. The general idea is to approximate the continuous density function by means of a Dirac mixture function that minimizes a certain distance measure to the true density. The approximation process itself is described in [8] and will therefore only be discussed briefly in this work. We will focus here on the complete filter consisting of approximation, prediction [9] and filter step.

Since we make use of a distance measure, we are able to quantify the quality of our approximation. Furthermore, it is possible to find an optimal number of components required for sufficient estimates. Following this idea we will extend our optimization method to a full estimation cycle by considering the measurement as well.

This work is based on a publication entitled *A State Estimator for Nonlinear Stochastic Systems Based on Dirac Mixture Approximations* [10] and is organized as follows: We will give a problem formulation in Section 2 followed by an overview of the complete filter in Section 3. The building blocks of the filter are described in Section 4 whereas Section 5 presents further optimization methods. Experimental results comparing the proposed filter to state-of-the-art filters are given in Section 6 followed by conclusions in Section 7.

## 2 Problem Formulation

We consider discrete-time nonlinear dynamic systems according to

$$x_{k+1} = a_k(x_k, u_k, w_k) .$$

The measurements of the system are given according to the nonlinear function

$$y_k = h_k(x_k, v_k) \ .$$

The state of the system is represented by  $x_k$ .  $u_k$  is a known input, and  $y_k$  is an observable output of the system.  $a_k(\cdot)$  is a time-varying nonlinear mapping describing the system's dynamic behavior.  $w_k$  represents both endogenous and exogenous noise sources acting upon the system and is described by means of a density function  $f_k^w(w_k)$ .  $h_k(\cdot)$  maps the system state to an output value which suffers from noise  $v_k$  modeled by means of a density function  $f_k^v(v_k)$ .

Starting with an initial state  $x_0$ , our goal is to keep track of the system's state over time while maintaining a full continuous stochastic representation of the uncertainty involved, caused by the system and measurement noise.

This corresponds to sequentially calculating the state densities  $f_k^x(x_k)$  for  $k = 1, \dots, N$  by means of a prediction and a filter step where the system and measurement functions are applied.

Exact computation of these densities, however, is not feasible, as the complexity of the density increases in every step. In addition, the resulting densities cannot be calculated in an analytic form in general.

The aim of this work is to provide a density representation that approximates the true density in order to allow for closed-form calculation of the prediction step while maintaining a predefined quality of the approximation with respect to a given distance measure.

For reasons of brevity, we omit the input  $u_k$  from now on. We further focus on additive noise, which results in the system equation

$$x_{k+1} = g_k(x_k) + w_k$$

and a measurement equation

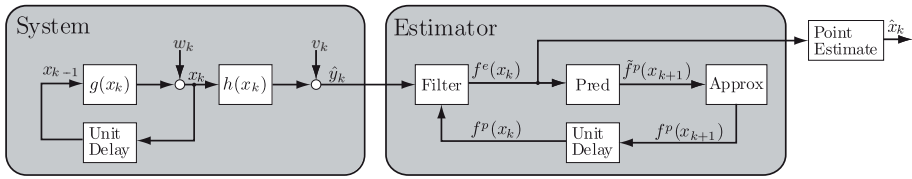
$$y_k = h_k(x_k) + v_k \ .$$

In addition, the time index  $k$  is omitted in some cases without notice.

### 3 Filter Outline

In this section, we will give a brief overview of the recursive filtering scheme depicted as a block diagram in Figure 1. The left part of the figure shows the nonlinear system suffering from additive noise as described in Sec. 2. The right part shows the estimator. The input of the estimator is a measurement  $\hat{y}_k$  coming from the system. The output of the estimator is a probability density function  $f^e(x_k)$  from which a point estimate  $\hat{x}_k$  can be derived. The estimator itself works recursively as can be seen from the loop in the diagram. Each recursion consists of a prediction step, an approximation step, and a filter step.

The prediction step receives a density  $f^e(x_k)$  from the previous filter step. This density is an approximation represented by means of a Dirac mixture allowing for an



**Fig. 1.** A block diagram of the recursive estimator. The left grey box shows the system given by the system and measurement equations. The estimator, shown in the grey box at the right, consists of a filter step, a prediction step and an approximation step. From the output of the estimator a point estimate can be derived.

analytically exact solution of the Bayesian prediction integral with respect to this approximation. The prediction yields a continuous mixture density representation (e.g., a Gaussian mixture)  $\tilde{f}^p(x_{k+1})$ . Details are given in Sec. 4.2.

The continuous mixture density  $\tilde{f}^p(x_{k+1})$  resulting from the prediction step serves as input to the approximation step. The density is systematically approximated by means of a Dirac mixture  $f^p(x_{k+1})$  minimizing a distance measure  $G(\tilde{f}^p(x_{k+1}))$ , ( $f^p(x_{k+1})$ ) as described in Sec. 4.1.

The approximated density  $f^p(x_{k+1})$  is then fed to the filter step, where it is fused with the likelihood function  $f^L(x, \hat{y})$ . This step is described in detail in Sec. 4.3.

## 4 Filter Components

### 4.1 Density Approximation

We will now introduce Dirac mixture functions and explain how they can be interpreted as parametric density functions. Subsequently, we will briefly describe the systematic approximation scheme.

**Dirac Mixture Density Representation.** Dirac mixtures are a sum of weighted Dirac delta functions according to

$$f(x, \underline{\eta}) = \sum_{i=1}^L w_i \delta(x - x_i) , \tag{1}$$

where

$$\underline{\eta} = [x_1, x_2, \dots, x_L, w_1, w_2, \dots, w_L]^T$$

is a parameter vector consisting of locations  $x_i, i = 1, \dots, L$  and weighting coefficients  $w_i, i = 1, \dots, L$ . The Dirac delta function is an impulse representation with the properties

$$\delta(x) = \begin{cases} 0, & x \neq 0 \\ \text{not defined,} & x = 0 \end{cases}$$

and

$$\int_{\mathbb{R}} \delta(x) dx = 1 .$$



The fundamental property of the Dirac delta function is given by

$$\int_{-\infty}^{\infty} f(x)\delta(x - x_i) dx = f(x_i) .$$

A mixture of Dirac delta functions as given in (1) can be used for representing arbitrary density functions if the following requirements are considered. Since the properties of a density function  $f(x)$  demand that  $f(x) \geq 0$  and  $\int_{\mathbb{R}} f(x) dx = 1$ , we have

$$w_i \geq 0, i = 1, \dots, L$$

and

$$\sum_{i=1}^L w_i = 1 .$$

Hence, we require  $2L$  parameters with  $2L - 1$  degrees of freedom.

A simplified density representation is given by equally weighted Dirac mixtures, as

$$f(x, \underline{\eta}) = \frac{1}{L} \sum_{i=1}^L \delta(x - x_i) ,$$

where only  $L$  parameters and  $L$  degrees of freedom are used. This results in a simpler, less memory consuming representation with less approximation capabilities.

Dirac mixtures are a generic density representation useful for approximating complicated densities arising in estimators for nonlinear dynamic systems.

**Approximation Approach.** A systematic approximation of a continuous density by means of another density requires a distance measure between the two densities

$$G \left( \tilde{f}^p(x_{k+1}), f^p(x_{k+1}, \underline{\eta}) \right) ,$$

where  $\tilde{f}^p(\cdot)$  is an arbitrary continuous density function and  $f^p(\cdot, \underline{\eta})$  is a Dirac mixture density. The approximation problem can then be reformulated as an optimization problem by finding a parameter vector  $\underline{\eta}$  that minimizes this distance measure.

Popular distance measures for comparing continuous densities are the Kullback–Leibler divergence [11] or the integral quadratic measure. For comparing a continuous density to a Dirac mixture, however, they are not very useful, since the Dirac mixture is undefined in the positions of the Dirac pulses and has zero values in between. Hence, instead of comparing the densities directly, the corresponding (cumulative) distribution functions are employed for that purpose. For the rest of this subsection we will omit the time index  $k$  and the  $p$  index in order to keep the formulae comprehensible.

The distribution function corresponding to the true density  $\tilde{f}(x)$  is given by

$$\tilde{F}(x) = \int_{-\infty}^x \tilde{f}(t) dt .$$

The distribution function corresponding to the Dirac mixture approximation can be written as

$$F(x, \underline{\eta}) = \int_{-\infty}^x f(t, \underline{\eta}) dt = \sum_{i=1}^L w_i H(x - x_i) \quad (2)$$

where  $H(\cdot)$  denotes the Heaviside function defined as

$$H(x) = \begin{cases} 0, & x < 0 \\ \frac{1}{2}, & x = 0 \\ 1, & x > 0 \end{cases} .$$

A suitable distance measure is given by the weighted Cramér–von Mises distance [12]

$$G(\underline{\eta}) = \int_{-\infty}^{\infty} r(x) \left( \tilde{F}(x) - F(x, \underline{\eta}) \right)^2 dx \quad (3)$$

where  $r(x)$  is a nonnegative weighting function.  $r(x)$  will be later in the filter step selected in such a way that only those portions of the predicted probability density function are approximated with high accuracy, where a certain support of the likelihood function is given. This avoids to put much approximation effort into irrelevant regions of the state space.

The goal is now to find a parameter vector  $\underline{\eta}$  that minimizes (3) according to  $\underline{\eta} = \arg \min_{\underline{\eta}} G(\underline{\eta})$ . Unfortunately, it is not possible to solve this optimization problem directly. Hence, we apply a progressive method introduced in [8]. For this approach, we introduce a so called progression parameter  $\gamma$  into  $\tilde{F}(x)$  that goes from  $0 \dots 1$ . The purpose of this parameter is to find a very simple and exact approximation of  $\tilde{F}(x, \gamma)$  for  $\gamma = 0$ . Further we must guarantee that  $\tilde{F}(x, \gamma = 1) = \tilde{F}(x)$ . By varying  $\gamma$  from 0 to 1 we track the parameter vector  $\underline{\eta}$  that minimizes the distance measure.

In order to find the minimum of the distance measure, we have to find the root of the partial derivative with respect to  $\underline{\eta}$  according to

$$\frac{\partial G(\underline{\eta}, \gamma)}{\partial \underline{\eta}} = \left[ \frac{\partial G(\underline{\eta}, \gamma)}{\partial \underline{x}} \right] \stackrel{!}{=} \underline{0} \quad (4)$$

Together with (2) and (3) this results in the system of equations

$$\begin{aligned} \tilde{F}(x_i, \gamma) &= \sum_{j=1}^L w_j H(x_i - x_j) \quad , \\ \int_{x_i}^{\infty} r(x) \tilde{F}(x, \gamma) dx &= \sum_{j=1}^L w_j \int_{x_i}^{\infty} r(x) H(x - x_j) dx \quad , \end{aligned}$$

where  $i = 1, \dots, L$ .

In order to track the minimum of the distance measure we have to take the derivative of (4) with respect to  $\gamma$ .

This results in a system of ordinary first order differential equations that can be written in a vector–matrix–form as

$$\underline{\dot{b}} = \mathbf{P} \underline{\dot{\eta}} \text{ ,} \tag{5}$$

where

$$\underline{b} = \begin{bmatrix} \frac{\partial \tilde{F}(x_1, \gamma)}{\partial \gamma} \\ \vdots \\ \frac{\partial \tilde{F}(x_L, \gamma)}{\partial \gamma} \\ \int_{x_0}^{\infty} \frac{\partial \tilde{F}(x, \gamma)}{\partial \gamma} dx \\ \int_{x_1}^{\infty} \frac{\partial \tilde{F}(x, \gamma)}{\partial \gamma} dx \\ \vdots \\ \int_{x_L}^{\infty} \frac{\partial \tilde{F}(x, \gamma)}{\partial \gamma} dx \end{bmatrix}$$

and

$$\underline{\dot{\eta}} = \frac{\partial \eta}{\partial \gamma} = [\dot{x}_1, \dots, \dot{x}_L, \dot{w}_0, \dot{w}_1, \dots, \dot{w}_L]^T \text{ .}$$

$\underline{\dot{\eta}}$  denotes the derivative of  $\underline{\eta}$  with respect to  $\gamma$ .

The  $\mathbf{P}$  matrix is given by

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix}$$

with

$$\mathbf{P}_{11} = \begin{bmatrix} -\tilde{f}(x_1, \gamma) & 0 & \cdots & 0 \\ 0 & -\tilde{f}(x_2, \gamma) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -\tilde{f}(x_L, \gamma) \end{bmatrix} \text{ ,}$$

$$\mathbf{P}_{12} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \cdots & 0 \\ 1 & \frac{1}{2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & \frac{1}{2} \end{bmatrix} \text{ ,}$$

$$\mathbf{P}_{21} = \begin{bmatrix} -\omega_1 & -\omega_2 & \cdots & -\omega_L \\ \tilde{F}(x_1, \gamma) - \omega_1 & -\omega_2 & \cdots & -\omega_L \\ 0 & \tilde{F}(x_2, \gamma) - \sum_{i=1}^2 \omega_i & \cdots & -\omega_L \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{F}(x_L, \gamma) - \sum_{i=1}^L \omega_i \end{bmatrix} \text{ ,}$$

and

$$\mathbf{P}_{22} = \begin{bmatrix} c - x_1 & c - x_2 & c - x_3 & \cdots & c - x_L \\ c - x_2 & c - x_2 & c - x_3 & \cdots & c - x_L \\ c - x_3 & c - x_3 & c - x_3 & \cdots & c - x_L \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c - x_L & c - x_L & c - x_L & \cdots & c - x_L \end{bmatrix} .$$

### 4.2 Prediction Step

We now explain the Bayesian prediction step and show how the approximation introduced in the last subsection can be used for closed-form calculations.

Calculation of the state densities  $f^p(x_{k+1})$ ,  $k = 1, \dots, N$ , is performed by evaluating the Bayesian forward step, which is given by

$$f^p(x_{k+1}) = \int_{-\infty}^{\infty} f(x_{k+1}|x_k) f^e(x_k) dx_k , \tag{6}$$

where the transition density  $f(x_{k+1}|x_k)$  of the considered nonlinear system with additive noise is given by

$$f(x_{k+1}|x_k) = f^w(x_{k+1} - g(x_k)) .$$

$f^w(\cdot)$  is the density of the system noise (e.g. Gaussian).

In general, the integral involved in (6) cannot be solved analytically for arbitrary prior densities  $f^e(x_k)$ . For a given input point  $\bar{x}_k$ , however, represented by the Dirac delta function  $f^e(x_k) = \delta(x_k - \bar{x}_k)$ , (6) can be solved in closed form according to

$$f_p(x_{k+1}) = f^w(x_{k+1} - g(\bar{x}_k)) .$$

In the case of zero mean Gaussian system noise with

$$f^w(w) = \mathcal{N}(w, 0, \sigma^w) ,$$

this yields

$$f_p(x_{k+1}) = \mathcal{N}(x_{k+1}, g(\bar{x}_k), \sigma^w) ,$$

which is a Gaussian Density with a standard deviation  $\sigma^w$ .

For a given Dirac mixture prior  $f^e(x_k)$  according to (1) given by

$$f^e(x_k) = \sum_{i=1}^L w_k^{(i)} \delta(x_k - x_k^{(i)}) , \tag{7}$$

the posterior according to (6) is a Gaussian mixture given by

$$f^p(x_{k+1}) = \sum_{i=1}^L w_k^{(i)} \mathcal{N}(x_{k+1}, g(x_k^{(i)}), \sigma^w) ,$$

which is a closed-form solution for the predicted state density.

Please note, that similar results can be derived for non-additive and non-Gaussian noise.

### 4.3 Filter Step

The filter step consists of fusing the predicted density  $f^P(x_k)$  and the likelihood function  $f^L(x_k, \hat{y}_k)$  governed by the measurement  $\hat{y}_k$  according to

$$f^e(x_k) = c \cdot f^P(x_k) \cdot f^L(x_k, \hat{y}_k) \quad , \quad (8)$$

where  $c$  is a normalizing constant. The likelihood function is given by

$$f^L(x_k, \hat{y}_k) = f(\hat{y}_k | x_k) \quad .$$

For a nonlinear system with additive noise, the conditional density for the measurement  $f(y_k | x_k)$  is given by

$$f(y_k | x_k) = f^v(y_k - h(x_k)) \quad ,$$

where  $f^v(\cdot)$  is the density of the measurement noise and  $h(x_k)$  is the nonlinear measurement function. In the case of zero-mean Gaussian measurement noise the likelihood function can be written as

$$f^L(x_k, \hat{y}_k) = \mathcal{N}(\hat{y}_k, h(x_k), \sigma^v) \quad .$$

We would like to emphasize, that in the nonlinear case this likelihood function is no proper density function and the update equation (8) cannot be solved analytically in general. Hence, a parametric representation of the posterior density can usually not be given. This prohibits the application of a recursive estimator scheme, since the derived prediction step depends on a parametric prior density.

Our solution to this problem is driven by the same observation made for solving the prediction step in Sec. 4.2. The likelihood can be evaluated at certain points  $\bar{x}_k$ , which yields constant values.

In order to calculate the product of a likelihood and a prediction, where the latter is already given as a Dirac mixture, it comes quite naturally to use the  $x_k^{(i)}$  points of the Diracs to evaluate the likelihood. The obtained values of  $f^L(\cdot)$  can then be used to reweight the predicted density according to

$$f^e(x_k) = \sum_{i=1}^L \bar{w}_k^{(i)} \delta(x_k - x_k^{(i)})$$

with

$$\bar{w}_k^{(i)} = c \cdot w_k^{(i)} \cdot f^v(\hat{y}_k - h(x_k^{(i)})) \quad ,$$

where  $w_k^{(i)}$  is the  $i$ 'th weight and  $x_k^{(i)}$  is the  $i$ 'th position of the approximated prediction  $f^P(x_k)$ . The normalization constant can be calculated as

$$c = \left( \sum_{i=1}^L w_k^{(i)} \cdot f^v(\hat{y}_k - h(x_k^{(i)})) \right)^{-1} \quad .$$

Naive approximation of the predicted density in a fixed interval may lead to many small weights, since not all regions of the state space supported by the prediction are as well supported by the likelihood. This phenomenon can be described as parameter degradation. To circumvent this problem, we make use of the weighting function  $r(x)$  in (3). Details on this approach are presented in the next section.

## 5 Optimal Number of Parameters

In this section, we describe how to tackle the problem of parameter degradation that is inherent to all filter approaches considering only discrete points of the density. We further describe a method for finding an optimal number of components for the approximation taking into account the prediction and filter steps as well.

To fight the problem of parameter degradation described in the previous section we make use of the fact, that although the likelihood function is not a proper density it decreases to zero for  $x \rightarrow \pm\infty$  in many cases. Therefore, we can define an area of support in the state space where the likelihood is larger than a certain value. This area of support is an interval and can be represented by the weighting function  $r(x)$  in (3). It guarantees, that all components of the approximation are located in this interval and are therefor not reweighed to zero in the filter step. In other words, the complete mass of the approximation function accounts for the main area of interest.

In [9], we introduced an algorithm for finding the optimal number of components required for the approximation with respect to the subsequent prediction step. We will now extend this algorithm in order to account for the preceding filter step as well.

---

**Algorithm 1 :** Optimal # of components w.r.t. filter step and posterior density.

---

```

1: Select max. Error Threshold  $G_{\max}$ 
2: Select initial number of Components  $L = L_0$ 
3: Select search step  $\Delta L$ 
4:  $f^L(x) = \text{likelihood}(\hat{y})$ 
5:  $r(x) = \text{support}(f^L(x))$ 
6:  $\kappa_t = \text{predict}(\text{filter}(\text{approx}(L_{\text{large}}, r(x)), \hat{y}))$ 
7: while  $G > G_{\max}$  do
8:    $\kappa = \text{predict}(\text{filter}(\text{approx}(L, r(x)), \hat{y}))$ 
9:    $G = G(\kappa_t, \kappa)$ 
10:   $L = L + \Delta L$ 
11: end while
12:  $L_l = L - 2\Delta L$ 
13:  $L_u = L - \Delta L$ 
14: while  $L_u - L_l > 1$  do
15:    $L_t = L_l + \lfloor \frac{L_u - L_l}{2} \rfloor$ 
16:    $\kappa = \text{predict}(\text{filter}(\text{approx}(L_t, r(x)), \hat{y}))$ 
17:    $G = G(\kappa_t, \kappa)$ 
18:   if  $G > G_{\max}$  then
19:      $L_l = L_t$ 
20:   else
21:      $L_u = L_t$ 
22:   end if
23: end while

```

---

At the beginning of Algorithm 1 in line 6, an initial approximation with a very large number of components is generated and passed through the prediction step, resulting in a continuous density representation with parameter vector  $\kappa_t$ . Due to the high number of components we can assume this density to be very close to the true density. An

efficient procedure for approximating arbitrary mixture densities with Dirac mixtures comprising a large number of components is given in [9].

In each search step of the algorithm, the distance measure of the approximated density at hand to the density defined by  $\underline{\kappa}_t$  is calculated. In this way the smallest number of components for a prespecified error can be found.

## 6 Experimental Results

In order to compare the performance of the proposed filter to state-of-the-art filters, we have simulated a nonlinear dynamic system according to the left part of Figure 1. We apply the filter to a cubic system and measurement function motivated by the cubic sensor problem introduced in [13].

The simulated system function is given by

$$g(x_k) = 2x_k - 0.5x_k^3 + w$$

and the additive noise is Gaussian with  $\sigma^w = 0.2$  standard deviation. The measurement function is

$$h(x_k) = x_k - 0.5x_k^3 + v$$

with additive Gaussian noise and  $\sigma^v = 0.5$ .

The generated measurements are used as input to our filter as well as to an unscented Kalman filter and a particle filter. The particle filter is applied in a variant with 20 particles and a variant with 1000 particles in order to compare the performance.

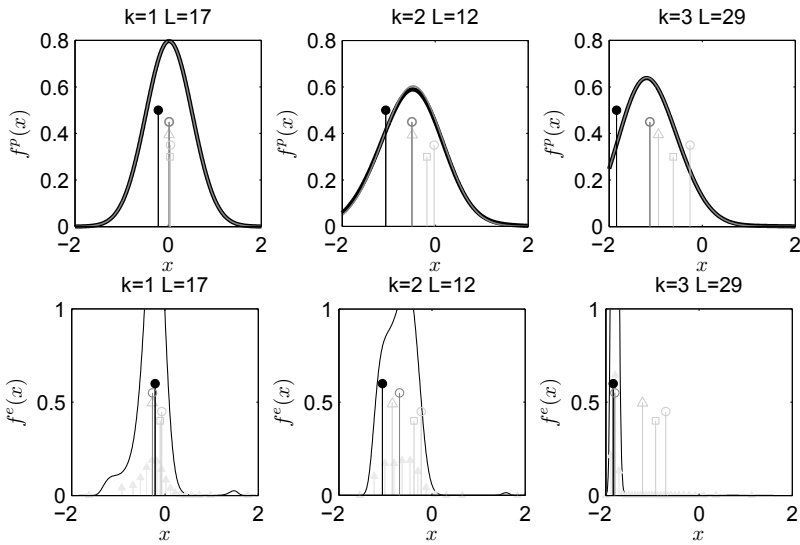
In a first run of the experiment we show  $T = 3$  steps in Figure 2. The upper row shows the prediction steps, the lower row shows the corresponding filter steps. The continuous prediction  $f^p(x_{k+1})$  of the Dirac mixture (DM) filter is depicted by the dark gray line. The black line underneath shows the true prediction computed numerically as a reference. The black plot in the lower line shows the likelihood function given by the current measurement and the filled light gray triangles depict the Dirac mixture components after the filter step.

Both rows also show the point estimates of the various applied filters in the current step. The filled black marker indicates the true (simulated) state, whereas dark gray stands for the Dirac mixture point estimate. Light gray with triangle is the UKF estimate and the other light gray markers are the particle filter estimates. The particle filter indicated by the circle uses 20 particles, the one indicated by the square uses 1000 particles.

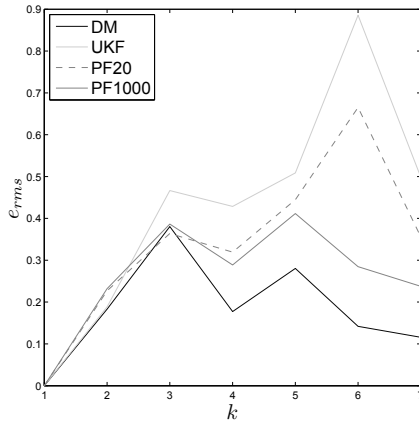
We simulated the system a further 10 times for  $T = 7$  steps in order to calculate the root means square error  $e_{\text{rms}}$  of the 4 filters. The results are shown in Figure 3. The plot shows that the point estimates of the Dirac mixture filter are much closer to the true state than the point estimates of the other filters.

## 7 Conclusions

In this paper, we presented a complete Dirac mixture filter that is based on the approximation of the posterior density. The filter heavily utilizes the properties of the Dirac



**Fig. 2.** The recursive filter for  $T = 3$  steps.  $k$  indicates the step number and  $L$  the number of components for the Dirac mixture. The upper row shows the prediction steps, the lower row shows the filter steps. **Upper row:** The dark grey curve is the continuous density predicted by the DM filter, the thick black line underneath is the true density. The filled black marker depicts the true (simulated) system state, the other markers depict the predicted point estimates of the following filters: dark gray=DM, light gray triangle=UKF, light gray circle=PF20, light gray square=PF1000. **Lower row:** The black line shows the likelihood. The encoding of the point estimates are similar to the upper line. Light gray filled triangles depict the Diracs.



**Fig. 3.** Root mean square error for 10 runs and  $T = 7$  steps.

mixture approximation for recursively calculating a closed form estimate. The key idea is that the approximation can be seen as an optimal representation of the true continuous density function. After each prediction step a full continuous density representation is used again in order to allow for an optimal reapproximation.



The new approach is natural, mathematically rigorous, and based on efficient algorithms [14, 8] for the optimal approximation of arbitrary densities by means of Dirac mixtures with respect to a given distance measure.

Compared to particle filters, the proposed method has several advantages. First, the Dirac components are systematically placed in order to minimize a given distance measure. The distance measure accounts for the actual measurement and guarantees that the prediction of the approximate densities is close to the true density of the next time step. As a result, very few components are sufficient for achieving an excellent estimation quality. Second, the optimization does not only include the parameters of the Dirac mixture approximation, i.e., weights and locations, but also the number of components. As a result, the number of components is automatically adjusted according to the complexity of the underlying true distribution and the support area of a given likelihood. Third, as the approximation is fully deterministic, it guarantees repeatable results.

Compared to the Unscented Kalman Filter, the Dirac mixture filter has the advantage, that it is not restricted to first and second order moments. Hence, multi-modal densities, which cannot be described sufficiently by using only the first two moments, can be treated very efficiently. Such densities occur quite often in strongly nonlinear systems. Furthermore, no assumptions on the joint distribution of state and measurement have to be made.

## References

1. Huber, M., Brunn, D., Hanebeck, U.D.: Closed-Form Prediction of Nonlinear Dynamic Systems by Means of Gaussian Mixture Approximation of the Transition Density. In: International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006), Heidelberg, Deutschland. (2006) 98–103
2. Doucet, A., Freitas, N.D., Gordon, N.: Sequential Monte Carlo Methods in Practice. Springer-Verlag, New York (2001)
3. Geweke, J.: Bayesian Inference in Econometric Models using Monte Carlo Integration. *Econometrica* **24** (1989) 1317–1399
4. Gordon, N.: Bayesian Methods for Tracking. PhD thesis, University of London (1993)
5. Julier, S., Uhlmann, J.: A New Extension of the Kalman Filter to Nonlinear Systems. In: Proceedings of SPIE AeroSense, 11th International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, Orlando, FL. (1997)
6. Alspach, D.L., Sorenson, H.W.: Nonlinear Bayesian Estimation Using Gaussian Sum Approximation. *IEEE Transactions on Automatic Control* **AC-17** (1972) 439–448
7. Hanebeck, U.D., Briechele, K., Rauh, A.: Progressive Bayes: A New Framework for Nonlinear State Estimation. In: Proceedings of SPIE. Volume 5099., Orlando, Florida (2003) 256–267 AeroSense Symposium.
8. Schrempf, O.C., Brunn, D., Hanebeck, U.D.: Dirac Mixture Density Approximation Based on Minimization of the Weighted Cramér–von Mises Distance. In: Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006), Heidelberg, Germany. (2006) 512–517
9. Schrempf, O.C., Hanebeck, U.D.: Recursive Prediction of Stochastic Nonlinear Systems Based on Dirac Mixture Approximations. In: Proceedings of the American Control Conference (ACC '07), New York City, USA. (2007)

10. Schrempp, O.C., Hanebeck, U.D.: A State Estimator for Nonlinear Stochastic Systems Based on Dirac Mixture Approximations. In: 4th Intl. Conference on Informatics in Control, Automation and Robotics (ICINCO 2007). Volume SPSMC., Angers, France (2007) 54–61
11. Kullback, S., Leibler, R.A.: On Information and Sufficiency. *Annals of Mathematical Statistics* **22** (1951) 79–86
12. Boos, D.D.: Minimum Distance Estimators for Location and Goodness of Fit. *Journal of the American Statistical association* **76** (1981) 663–670
13. Bucy, R.S.: Bayes Theorem and Digital Realizations for Non-Linear Filters. *Journal of Astronautical Sciences* **17** (1969) 80–94
14. Schrempp, O.C., Brunn, D., Hanebeck, U.D.: Density Approximation Based on Dirac Mixtures with Regard to Nonlinear Estimation and Filtering. In: Proceedings of the 45th IEEE Conference on Decision and Control (CDC'06), San Diego, California, USA. (2006)

# On the Geometry of Predictive Control with Nonlinear Constraints

Sorin Olaru<sup>1</sup>, Didier Dumur<sup>1</sup> and Simona Dobre<sup>2</sup>

<sup>1</sup> Automatic Control Department, SUPELEC, 3 rue Joliot Curie, Gif-sur-Yvette, France  
{sorin.olaru, didier.dumur}@supelec.fr

<sup>2</sup> CRAN, BP 239, 54506 Vandœuvre-lès-Nancy, France  
simona.dobre@cran.uhp-nancy.fr

**Abstract.** This paper proposes a geometrical analysis of the polyhedral feasible domains for the predictive control laws under constraints.

The state vector is interpreted as a vector of parameters for the optimization problem to be solved at each sampling instant and its influence can be fully described by the use of parameterized polyhedra and their dual constraints/generators representation.

The construction of the associated explicit control laws at least for linear or quadratic cost functions can thus receive fully geometrical solutions. Convex nonlinear constraints can be approximated using a description based on the parameterized vertices. In the case of nonconvex regions the explicit solutions can be obtained using Voronoi partitions based on a collection of points distributed over the borders of the feasible domain.

## 1 Introduction

The philosophy behind Model-based Predictive Control (MPC) is to exploit in a "receding horizon" manner the simplicity of the open-loop optimal control. The control action  $u_t$  for a given state  $x_t$  is obtained from the control sequence  $\mathbf{k}_u^* = [u_t^T, \dots, u_{t+N-1}^T]^T$  as a result of the optimization problem:

$$\begin{aligned} \min_{\mathbf{k}_u} \quad & \varphi(x_{t+N}) + \sum_{k=0}^{N-1} l(x_{t+k}, u_{t+k}) \\ \text{subj. to : } & x_{t+1} = f(x_t) + g(x_t)u_t; \\ & h(x_t, \mathbf{k}_u) \leq 0 \end{aligned} \tag{1}$$

constructed for a finite prediction horizon  $N$ , cost per stage  $l(\cdot)$ , terminal weight  $\varphi(\cdot)$ , the system dynamics described by  $f(\cdot), g(\cdot)$  and the constraints written in a compact form using elementwise inequalities on functions linking the states and the control actions,  $h(\cdot)$ .

The control sequence  $\mathbf{k}_u^*$  is optimal for an initial condition -  $x_t$  and produces an open-loop trajectory which contrasts with the need for a feedback control law. This drawback is overcome by solving the local optimization (1) for every encountered (measured) state, thus indirectly producing a state feedback law.

For the optimization problem (1) within MPC, the current state serves as an initial condition and influences both the objective function and the topology of the feasible

domain. Globally, the system state can be interpreted as a vector of parameters, and the problems to be solved are part of the multiparametric optimization programming family. From the cost function point of view, the parametrization is somehow easier to deal with and eventually can be entirely translated toward the set of constraints to be satisfied (the MPC literature contains references to schemes based on suboptimality or even to algorithms restraining the demands to feasible solution of the receding horizon optimization [1]). Unfortunately, similar observation cannot be made about the feasible domain and its adjustment with respect to the parameters evolution. The optimal solution is thus often influenced by the constraints activation, the process being forced to operate at the designed constraints for best performance. The distortion of the feasible domain during the parameters evolution will consequently affect the structure of the optimal solution. Starting from this observation the present paper focuses on the analysis of the geometry of the domains described by the MPC constraints.

The structure of the feasible domain is depending on the model and the set of constraints taken into consideration in (1). If the model is linear, the linear constraints on inputs and states can be easily expressed by a system of linear inequalities. In the case of nonlinear systems, these properties are lost but there are several approaches to transform the dynamics to those of a linear system over the operating range as for example by piecewise linear approximation, feedback linearisation or the use of time-varying linear models.

In the present paper, the feasible domains will be analyzed with a focus on the parametrization mainly upon the concept of parameterized polyhedra [2], which appears in the MPC formulations like:

$$\begin{aligned} \min_{\mathbf{k}_u} \quad & F(x_t, \mathbf{k}_u) \\ \text{subj. to :} \quad & \begin{cases} A_{in}\mathbf{k}_u \leq b_{in} + B_{in}x_t \\ A_{eq}\mathbf{k}_u = b_{eq} + B_{eq}x_t \\ h(x_t, \mathbf{k}_u) \leq 0 \end{cases} \end{aligned} \quad (2)$$

where the objective function  $F(x_t, \mathbf{k}_u)$  is usually linear or quadratic. Secondly it will be shown that the optimization problem may take advantage during the real-time implementation from the construction of the explicit solution.

In the presence of nonlinearities  $h(x_t, \mathbf{k}_u) \leq 0$  two cases can be treated:

- feasible domain are convex - the approximation in terms of parameterized polyhedra leads to an approximate explicit solution using the same arguments as for the exact solutions;
- feasible domain is non-convex - an algorithmic construction of explicit control laws upon Voronoi partition of the parameters space will be used.

In the following, Section 2 introduces the basic concepts related to the parameterized polyhedra. Section 3 presents the use of the feasible domain analysis for the construction of the explicit solution for linear and quadratic objective functions. In Section 4 an extension to nonlinear type of constraints is addressed.

## 2 Parametrization of Polyhedral Domains

### 2.1 Double Representation

A mixed system of linear equalities and inequalities defines a polyhedron [3]. In the parameter free case, it is represented by the equivalent dual (Minkowski) formulation:

$$\begin{aligned} \mathcal{P} &= \{ \mathbf{k}_u \in \mathbb{R}^p \mid A_{eq} \mathbf{k}_u = b_{eq}; A_{in} \mathbf{k}_u \leq b_{in} \} \\ \iff \mathcal{P} &= \underbrace{\text{conv.hull} \mathbf{V} + \text{cone} \mathbf{R} + \text{lin.space} \mathbf{L}}_{\text{generators}} \end{aligned} \quad (3)$$

where  $\text{conv.hull} \mathbf{V}$  denotes the set of convex combinations of vertices  $\mathbf{V} = \{ \mathbf{v}_1, \dots, \mathbf{v}_\vartheta \}$ ,  $\text{cone} \mathbf{R}$  denotes nonnegative combinations of unidirectional rays in  $\mathbf{R} = \{ \mathbf{r}_1, \dots, \mathbf{r}_\rho \}$  and  $\text{lin.space} \mathbf{L} = \{ \mathbf{l}_1, \dots, \mathbf{l}_\lambda \}$  represents a linear combination of bidirectional rays (with  $\vartheta$ ,  $\rho$  and  $\lambda$  the cardinals of the related sets). This dual representation [4] in terms of generators can be rewritten as:

$$\begin{aligned} \mathcal{P} &= \left\{ \mathbf{k}_u \in \mathbb{R}^p \mid \mathbf{k}_u = \sum_{i=1}^{\vartheta} \alpha_i \mathbf{v}_i + \sum_{i=1}^{\rho} \beta_i \mathbf{r}_i + \sum_{i=1}^{\lambda} \gamma_i \mathbf{l}_i; \right. \\ &\quad \left. 0 \leq \alpha_i \leq 1, \sum_{i=1}^{\vartheta} \alpha_i = 1, \beta_i \geq 0, \forall \gamma_i \right\} \end{aligned} \quad (4)$$

with  $\alpha_i, \beta_i, \gamma_i$  the coefficients describing the convex, non-negative and linear combinations in (3). Numerical methods like the Chernikova algorithm [5] are implemented for constructing the double description, either starting from constraints (3) either from the generators (4) representation.

### 2.2 The Parametrization

A *parameterized polyhedron* [6] is defined in the implicit form by a finite number of inequalities and equalities with the note that the affine part depends linearly on a vector of parameters  $x \in \mathbb{R}^n$  for both equalities and inequalities:

$$\begin{aligned} \mathcal{P}(x) &= \{ \mathbf{k}_u(x) \in \mathbb{R}^p \mid A_{eq} \mathbf{k}_u = B_{eq} x + b_{eq}; A_{in} \mathbf{k}_u \leq B_{in} x + b_{in} \} \\ &= \left\{ \mathbf{k}_u(x) \mid \mathbf{k}_u(x) = \sum_{i=1}^{\vartheta} \alpha_i(x) \mathbf{v}_i(x) + \sum_{i=1}^{\rho} \beta_i \mathbf{r}_i + \sum_{i=1}^{\lambda} \gamma_i \mathbf{l}_i \right\} \\ &\quad 0 \leq \alpha_i(x) \leq 1, \sum_{i=1}^{\vartheta} \alpha_i(x) = 1, \beta_i \geq 0, \forall \gamma_i. \end{aligned} \quad (5)$$

This dual representation of the parameterized polyhedral domain reveals the fact that only the vertices are concerned by the parametrization (resulting the so-called *parameterized vertices* -  $\mathbf{v}_i(x)$ ), whereas the rays and the lines do not change with the parameters' variation. In order to effectively use the generators representation in (5), several aspects have to be clarified regarding the parametrization of the vertices (see for example [6]). The idea is to identify the parameterized polyhedron by a non-parameterized one in an augmented space:

$$\tilde{\mathcal{P}} = \left\{ \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \in \mathbb{R}^{p+n} \mid [A_{eq}] - B_{eq} \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} = b_{eq}; [A_{in}] - B_{in} \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \leq b_{in} \right\} \quad (6)$$

The original polyhedron in (5) can be found for any particular value of the parameters vector  $x$  through  $P(x) = \text{Proj}_{\mathbf{k}_u} \left( \tilde{P} \cap H(x) \right)$ , for any given hyperplane  $H(x_0) = \left\{ \begin{pmatrix} \mathbf{k}_u \\ x \end{pmatrix} \in \mathbb{R}^{p+n} \mid x = x_0 \right\}$  and using  $\text{Proj}_{\mathbf{k}_u}(\cdot)$  as the projection from  $\mathbb{R}^{p+n}$  to the first  $p$  coordinates  $\mathbb{R}^p$ .

Within the polyhedral domains  $\tilde{P}$ , the correspondent of the parameterized vertices in (5) can be found among the faces of dimension  $n$ . After enumerating these  $n$ -faces:  $\{F_1^n(\tilde{P}), \dots, F_j^n(\tilde{P}), \dots, F_\zeta^n(\tilde{P})\}$ , one can write:

$\forall i, \exists j \in \{1, \dots, \zeta\}$  s.t.  $[\mathbf{v}_i(x)^T \ x^T]^T \in F_j^n(\tilde{P})$  or equivalently:

$$\mathbf{v}_i(x) = \text{Proj}_{\mathbf{k}_u} \left( F_j^n(\tilde{P}) \cap H(x) \right) \quad (7)$$

From this relation it can be seen that not all the  $n$ -faces correspond to parameterized vertices. However it is still easy to identify those which can be ignored in the process of construction of parameterized vertices based on the relation  $\text{Proj}_x \left( F_j^n(\tilde{P}) \right) < n$  with  $\text{Proj}_x(\cdot)$  the projection from  $\mathbb{R}^{p+n}$  to the last  $n$  coordinates  $\mathbb{R}^n$  (corresponding to the parameters' space). Indeed the projections are to be computed for all the  $n$ -faces, those which are degenerated are to be discarded and all the others are stored as validity domains -  $D_{\mathbf{v}_i} \in \mathbb{R}^n$ , for the parameterized vertices that they are identifying:

$$D_{\mathbf{v}_i} = \text{Proj}_n \left( F_j^n(\tilde{P}) \right) \quad (8)$$

Once the parameterized vertices identified and their validity domain stored, the dependence on the parameters vector can be found using the supporting hyperplanes for each  $n$ -face:

$$\mathbf{v}_i(x) = \left[ \begin{array}{c} A_{eq} \\ \bar{A}_{in_j} \end{array} \right]^{-1} \left[ \begin{array}{c} B_{eq} \\ \bar{B}_{in_j} \end{array} \right] x + \left[ \begin{array}{c} b_{eq} \\ \bar{b}_{in_j} \end{array} \right] \quad (9)$$

where  $\bar{A}_{in_j}, \bar{B}_{in_j}, \bar{b}_{in_j}$  represent the subset of the inequalities, satisfied by saturation for  $F_j^n(\tilde{P})$ . The inversion is well defined as long as the faces with degenerate projections are discarded.

### 2.3 The Interpretation from the Predictive Control Point of View

The double representation of the parameterized polyhedra offers a complete description of the feasible domain for the MPC law as long as this is based on a multiparametric optimization with linear constraints. Using the generators representation one can compute the region of the parameters space where no parameterized vertex is defined:

$$\aleph = \mathbb{R}^n \setminus \{\cup D_{\mathbf{v}_i}; i = 1 \dots \vartheta\} \quad (10)$$

representing the set of infeasible states for which no control sequence can be designed due to the fact that the limitations are overly constraining. As a consequence the complete description of the infeasibility is obtained.

The vertices of the feasible domain cannot be expressed as convex combinations of other distinct points and, due to the fact that from the MPC point of view, they represent sequences of control actions, one can interpret them in terms of extremal performances of the controlled system (for example in the tracking applications the maximal/minimal admissible setpoint [7]).

### 3 Toward Explicit Solutions for Polyhedral Domains

In the case of sufficiently large memory resources, construction of the explicit solution for the multiparametric optimization problem (2) can be an interesting alternative to the iterative optimization routines. In this direction recent results were presented at least for the case of linear and quadratic cost functions (see [8],[9],[10],[11],[12]). In the following it will be shown that a geometrical approach based on the parameterized polyhedra can bring a useful insight as well.

#### 3.1 Linear Cost Function

The linear cost functions are extensively used in connection with model based predictive control and especially for robust case ([13], [14]). In a compact form, the multiparametric optimization problem is:

$$\begin{aligned} \mathbf{k}_u^*(x_t) &= \min_{\mathbf{k}_u} f^T \mathbf{k}_u \\ \text{subject to } A_{in} \mathbf{k}_u &\leq B_{in} x_t + b_{in} \end{aligned} \tag{11}$$

The problem deals with a polyhedral feasible domain which can be described as previously in a double representation. Further the explicit solution can be constructed based on the relation between the parameterized vertices and the linear cost function (as in [5]). The next result resumes this idea.

**Proposition.** The solution for a multiparametric linear problem is characterized as follows:

- a) For the subdomain  $\aleph \in \mathbb{R}^n$  where the associated parameterized polyhedron has no valid parameterized vertex the problem is infeasible;
- b) If there exists a bidirectional ray  $\mathbf{l}$  such that  $f^T \mathbf{l} \neq 0$  or a unidirectional ray  $\mathbf{r}$  such that  $f^T \mathbf{r} \leq 0$ , then the minimum is unbounded;
- c) If all bidirectional rays  $\mathbf{l}$  are such that  $f^T \mathbf{l} = 0$  and all unidirectional rays  $\mathbf{r}$  are such that  $f^T \mathbf{r} \geq 0$  then there exists a cutting of the parameters in zones where the parameterized polyhedron has a regular shape  $\bigcup_{j=1 \dots \rho} R_j = \mathbb{R}^n - \aleph$ . For each region  $R_j$  the minimum is computed with respect to the given linear cost function and for all the valid parameterized vertices:

$$\underline{m}(x) = \min \{ f^T \mathbf{v}_i(x) | \mathbf{v}_i(x) \text{ vertex of } \mathcal{P}(x) \} \tag{12}$$

The minimum  $\underline{m}(x)$  is attained by constant subsets of parameterized vertices of  $\mathcal{P}(x)$  over a finite number of polyhedral zones in the parameters space  $R_{ij} (\cup R_{ij} =$

$R_j$ ). The complete optimal solution of the multiparametric optimization is given for each  $R_{ij}$  by:

$$S_{R_{ij}}(x) = \text{conv.hull} \{ \mathbf{v}_1^*(x), \dots, \mathbf{v}_s^*(x) \} + \text{cone} \{ \mathbf{r}_1^*, \dots, \mathbf{r}_r^* \} + \text{lin.space} \mathcal{P} \quad (13)$$

where  $\mathbf{v}_i^*$  are the vertices corresponding to the minimum  $\underline{m}(x)$  over  $R_{ij}$  and  $\mathbf{r}_i^*$  are such that  $f^T \mathbf{r}_i^* = 0$  □

This result provides *the entire family of solutions* for the linear multiparametric optimization, even for the cases where this family is not finite (for example there are several vertices attaining the minimum). For the control point of view a continuous piecewise candidate is preferred, eventually by minimizing the number of partitions in the parameters space [15].

### 3.2 Quadratic Cost Function

The case of a quadratic cost function is one of the most popular for the linear nominal MPC. The explicit solution based on the exploration of the parameters space ([9], [11], [12]) is extensively studied lately. Alternative methods based on geometrical arguments or dynamic programming ([10], [8]) improved also the awareness of the explicit MPC formulations. The parameterized polyhedra can serve as a base in the construction of such explicit solution [2], for a quadratic multiparametric problem:

$$\begin{aligned} \mathbf{k}_u^*(x_t) &= \arg \min_{\mathbf{k}_u} \mathbf{k}_u^T H \mathbf{k}_u + 2 \mathbf{k}_u^T F x_t \\ \text{subject to } A_{in} \mathbf{k}_u &\leq B_{in} x_t + b_{in} \end{aligned} \quad (14)$$

In this case the main idea is to consider the unconstrained optimum:

$$\mathbf{k}_u^{sc}(x_t) = H^{-1} F x_t$$

and its position with respect to the feasible domain given by a parameterized polyhedron as in (5).

If a simple transformation is performed:

$$\tilde{\mathbf{k}}_u = H^{1/2} \mathbf{k}_u$$

then the isocost curves of the quadratic function are transformed from ellipsoid into circles centered in  $\tilde{\mathbf{k}}_u^{sc}(x_t) = H^{-1/2} F x_t$ . Further one can use the Euclidean projection in order to retrieve the multiparametric quadratic explicit solution.

Indeed if the unconstrained optimum  $\tilde{\mathbf{k}}_u^{sc}(x_t)$  is contained in the feasible domain  $\tilde{\mathcal{P}}(x_t)$  then it is also the solution of the constrained case, otherwise existence and uniqueness are assured as follows:

**Proposition.** For any exterior point  $\tilde{\mathbf{k}}_u(x_t) \notin \tilde{\mathcal{P}}(x_t)$ , there exists a unique point characterized by a minimal distance with respect to  $\tilde{\mathbf{k}}_u^{sc}(x_t)$ . This point satisfies:

$$(\tilde{\mathbf{k}}_u^{sc}(x_t) - \tilde{\mathbf{k}}_u^*(x_t))^T (\tilde{\mathbf{k}}_u - \tilde{\mathbf{k}}_u^*(x_t)) \leq 0, \forall \tilde{\mathbf{k}}_u \in \tilde{\mathcal{P}}(x_t) \quad \square$$

The construction mechanism uses the parameterized vertices in order to split the regions neighboring the feasible domain in zones characterized by the same type of projection.



## 4 Generalization to Nonlinear Constraints

If the feasible domain is described by a mixed linear/nonlinear set of constraints then the convexity properties are lost and a procedure for the construction of exact explicit solutions does not exist for the general case.

Consider now the case of mixed type of constraints (linear/nonlinear):

$$\mathbf{k}_u^* = \arg \min_{\mathbf{k}_u} 0.5\mathbf{k}_u^T H \mathbf{k}_u + \mathbf{k}_u^T F x \quad (15)$$

$$\begin{cases} h(x, \mathbf{k}_u) \leq 0 \\ A_{in} \mathbf{k}_u \leq b_{in} + B_{in} x \end{cases}$$

### 4.1 Optimality Conditions for Nonlinear Constraints

Let  $\bar{x}$  be a feasible parameter vector. The KKT optimality conditions can still be formulated as:

– Primal feasibility:

$$\begin{cases} h(\bar{x}, \mathbf{k}_u) \leq 0 \\ A_{in} \mathbf{k}_u \leq b_{in} + B_{in} \bar{x} \end{cases} \quad (16)$$

– Dual feasibility:

$$H \mathbf{k}_u + F^T \bar{x} + A_{in}^T \mu + \nabla_{\mathbf{k}_u} h(\bar{x}, \mathbf{k}_u)^T \nu = 0; \mu \geq 0, \nu \geq 0 \quad (17)$$

– Complementary slackness:

$$[\mu^T \nu^T] \begin{bmatrix} A_{in} \mathbf{k}_u - B_{in} \bar{x} - b_{in} \\ h(\bar{x}, \mathbf{k}_u) \end{bmatrix} = 0 \quad (18)$$

The difference resides in the fact that the KKT conditions are only necessary and not sufficient for optimality due to the presence of nonlinearity.

### 4.2 The Topology of the Feasible Domain

Indeed the sufficiency is lost due to the lack of constraint qualification (the Abadie constraint qualification holds automatically for the linear constraints but needs additional assumptions for the nonlinear case, see the next theorem).

**Theorem (KKT Sufficient Conditions) [10].** Let  $x = \bar{x}$  and the associated feasible domain  $\mathbf{U}(\bar{x})$  be a nonempty set in  $\mathbb{R}^{N^m}$  described by the constraints in (15), with  $h_i(\mathbf{k}_u) = h_i(\bar{x}, \mathbf{k}_u) : \mathbb{R}^{N^m} \rightarrow \mathbb{R}$ , the components of  $h(\mathbf{k}_u)$ . Let  $\mathbf{k}_u^* \in \mathbf{U}(\bar{x})$  and let  $\mathcal{I} = \{i : h_i(\bar{x}, \mathbf{k}_u^*) = 0\}$ ,  $\mathcal{J} = \{j : A_{in_j} \mathbf{k}_u^* - B_{in_j} \bar{x} - b_{in_j} = 0\}$ . Suppose the KKT conditions hold, such that:

$$H \mathbf{k}_u^* + F^T \bar{x} + \sum \mu_j A_{in_j}^T + \sum \gamma_i \nabla_{\mathbf{k}_u} h_i(\bar{x}, \mathbf{k}_u^*)^T = 0 \quad (19)$$

If  $h_i$  is quasiconvex at  $\mathbf{k}_u^* \forall i \in \mathcal{I}$ , then this represents a global solution to (15)  $\square$

Due to these problems, up to date, the explicit solutions for the general nonlinear multiparametric programming case were not tackled. Only for convex nonlinearities approximate explicit solutions were proposed [16].

In the following a solution based on linear approximation of feasible domains is proposed. This will answer the question regarding the optimality of a solution with piecewise linear structure.

### 4.3 Preliminaries for Linear Approximations of Mixed Linear/Nonlinear Feasible Domains

The idea is to exploit the existence of linear constraints in (15) and construct exact solutions as long as the unconstrained optimum can be projected on them. In a second stage if the unconstrained optimum is projected on the convex part of the nonlinear constraints, then an approximate solution is obtained by their linearization. Finally if the unconstrained optimum has to be projected on the nonconvex constraints then a Voronoi partition is used to construct the explicit solution. Before detailing the algorithms several useful tools are introduced.

**Gridding of the Parameter Space.** The parameters (state) space is sampled in order to obtain a representative grid  $\mathcal{G}$ . The way of distributing the points in the state space may follow a uniform distribution, logarithmic or tailored according to the a-priori knowledge of the nonlinearities.

For each point of the grid  $x \in \mathcal{G}$  a set of points on the frontier of the feasible domain  $D(x)$  can be obtained  $\mathcal{V}_x$  by the same kind of parceling. By collecting  $\mathcal{V}_x$  for all  $x \in \mathcal{G}$  a distribution of points  $\mathcal{V}_G$  in the extended arguments+parameters space is obtained.

**Convex Hulls.** A basic operation is the construction of the convex hull (or a adequate approximation) for the feasible domain in (15). Writing this parameterized feasible domain as:

$$D(x) = \{ \mathbf{k}_u \mid h(x, \mathbf{k}_u) \leq 0; A_{in} \mathbf{k}_u \leq b_{in} + B_{in} x \} \tag{20}$$

and using the distribution of points on the frontier  $\mathcal{V}_G$ , one can define in the extended (argument+parameters) space a convex hull  $\mathcal{C}_{\mathcal{V}_G}$ :

$$\begin{aligned} \mathcal{C}_{\mathcal{V}_G} = & \left\{ \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \in \mathbb{R}^{mN+n} \mid \exists \begin{bmatrix} \mathbf{k}_{u_i} \\ x_i \end{bmatrix}, i = 1..mN + n + 1, \mathbf{k}_{u_i} \in \mathcal{V}_G, \right. \\ & \left. s.t. \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} = \sum_{i=1}^{mN+n+1} \lambda_i \begin{bmatrix} \mathbf{k}_{u_i} \\ x_i \end{bmatrix}, \sum_{i=1}^{mN+n+1} \lambda_i = 1; \lambda_i \geq 0 \right\} \end{aligned} \tag{21}$$

**Voronoi Partition.** The Voronoi partition is the decomposition of a metric space  $\mathbb{R}^n$  in regions associated with a specified discrete set of points.

Let  $S = \{s_1, s_2, \dots, s_\nu\}$  be a collection of  $\nu$  points in  $\mathbb{R}^n$ . For each point  $s_i$  a set  $V_i$  is associated such that  $\bigcup_i V_i = \mathbb{R}^n$ . The definition of  $V_i$  will be:

$$V_i = \{x \in \mathbb{R}^n \mid \|x - v_i\|_2 \leq \|x - v_j\|_2, \forall j \neq i\} \tag{22}$$

It can be observed that each frontier of  $V_i$  is part of the bisection hyperplane between  $s_i$  and one of the neighbor points  $s_j$ . As a consequence of this fact, the regions  $V_i$  are polyhedrons. Globally, the Voronoi partition is a decomposition of space  $\mathbb{R}^n$  in  $\nu$  polyhedral regions.

### 4.4 Nonparameterized Case

In the following  $\mathfrak{F}(X)$  (and  $\mathfrak{Int}(X)$ ) represents the frontier (and the interior respectively) for a compact set  $X$ .

Consider the nonparameterized optimization problem:

$$\mathbf{k}_u^* = \arg \min_{\mathbf{k}_u} 0.5 \mathbf{k}_u^T \mathbf{k}_u + c^T \mathbf{k}_u \quad (23)$$

$$\begin{cases} h(\mathbf{k}_u) \leq 0 \\ A_{in} \mathbf{k}_u \leq b_{in} + B_{in} x \end{cases}$$

In relation with the feasible domain  $D$  of 23 we define:

$\mathfrak{R}_L(D)$  The set of linear constraints in the definition of  $D$

$\mathfrak{R}_{NL}(D)$  The set of nonlinear constraints in the definition of  $D$

$\mathfrak{S}(\mathfrak{R}_*, \mathbf{k}_u)$  The subset of constraints in  $\mathfrak{R}_*$  (either  $\mathfrak{R}_L$ , either  $\mathfrak{R}_{NL}$ ) saturated by the vector  $\mathbf{k}_u$

$\mathfrak{B}(\mathfrak{R}_*, \mathbf{k}_u)$  The subset of constraints in  $\mathfrak{R}_*$  violated by the vector  $\mathbf{k}_u$

Algorithm:

1. Obtain a set of points ( $\mathcal{V}$ ) on the frontier of the feasible domain  $D$
2. Construct the convex hull  $\mathcal{C}_{\mathcal{V}}$
3. Split the set  $\mathcal{V}$  as  $\tilde{\mathcal{V}} \cup \bar{\mathcal{V}}_L \cup \bar{\mathcal{V}}_{NL} \cup \hat{\mathcal{V}}$ 
  - $\tilde{\mathcal{V}} \in \mathfrak{F}(\mathcal{C}_{\mathcal{V}})$  and  $\mathcal{C}_{\mathcal{V}} = \mathcal{C}_{\mathcal{V} \setminus \tilde{\mathcal{V}}}$  (in words,  $\tilde{\mathcal{V}}$  contains those points in  $\mathcal{V}$  which lie on the frontier of  $\mathcal{C}_{\mathcal{V}}$  but are not vertices);
  - $\mathcal{V}_L \in \mathcal{V} \setminus \tilde{\mathcal{V}}$ ,  $\mathcal{V}_L \in \mathfrak{F}(\mathcal{C}_{\mathcal{V}})$  and  $\mathcal{V}_L$  saturates at least one linear constraint
  - $\mathcal{V}_{NL} \in \mathcal{V} \setminus \tilde{\mathcal{V}}$ ,  $\mathcal{V}_{NL} \in \mathfrak{F}(\mathcal{C}_{\mathcal{V}})$  and  $\mathcal{V}_{NL}$  saturates only nonlinear constraints
  - $\hat{\mathcal{V}} \in \mathfrak{Int}(\mathcal{C}_{\mathcal{V}})$
4. Construct the dual representation of  $\mathcal{C}_{\mathcal{V}}$ . This will be represented as an intersection of halfspaces  $\mathcal{H}$ .
5. Split  $\mathcal{H}$  in  $\bar{\mathcal{H}} \cup \hat{\mathcal{H}}$ 
  - $\hat{\mathcal{H}} \subset \mathcal{H}$  such that  $\exists x \in \mathcal{C}_{\mathcal{V}}$  with  $\mathfrak{S}(\hat{\mathcal{H}}, x) \neq \emptyset$  and  $\mathfrak{B}(\mathfrak{R}_{NL}, x) \neq \emptyset$
  - $\bar{\mathcal{H}} = \mathcal{H} \setminus \hat{\mathcal{H}}$
6. Project the unconstrained optimum  $\mathbf{k}_u = -c$  on  $\mathcal{C}_{\mathcal{V}}$ :

$$\mathbf{k}_u^* \leftarrow Proj_{\mathcal{C}_{\mathcal{V}}} \{-c\}$$

7. If  $\mathbf{k}_u^*$  saturates a subset of constraints  $\mathcal{K} \subset \hat{\mathcal{H}}$

(a) Retain the set of points:

$$S = \left\{ v \in \hat{\mathcal{V}} \mid \forall \mathbf{k}_u \in \mathcal{C}_{\mathcal{V}} \text{ s.t. } \mathfrak{S}(\hat{\mathcal{H}}, \mathbf{k}_u) = \mathcal{K}; \quad \mathfrak{B}(\mathfrak{R}_{NL}, \mathbf{k}_u) = \mathfrak{S}(\mathfrak{R}_{NL}, v) \right\}$$

(b) Construct the Voronoi partition for the collection of points in  $S$

(c) Position  $\mathbf{k}_u^*$  w.r.t. this partition and map the suboptimal solution  $\mathbf{k}_u^* \leftarrow v$  where  $v$  is the vertex corresponding to the active region

8. If the quality of the solution is not satisfactory, improve the distribution of the points  $\mathcal{V}$  by augmenting the resolution around  $\mathbf{k}_u^*$  and restart from (2).

#### 4.5 Explicit Solution - Taking into Account the Parametrization

Consider now the multiparametric optimization:

$$\mathbf{k}_u^* = \arg \min_{\mathbf{k}_u} 0.5\mathbf{k}_u^T H \mathbf{k}_u + \mathbf{k}_u^T F x \quad (24)$$

and the feasible combinations defined by the set:

$$D = \left\{ \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \in \mathbb{R}^{mN+n} \mid \begin{array}{l} h(x, \mathbf{k}_u) \leq 0 \\ A_{in} \mathbf{k}_u \leq b_{in} + B_{in} x \end{array} \right\}$$

Algorithm:

1. Grid the parameters space  $\mathbb{R}^n$  and retain the feasible nodes  $\mathcal{G}$
2. Obtain in the extended argument+parameters space a set of points ( $\mathcal{V}_G$ ) lying on the frontier of  $D$
3. Construct the convex hull  $\mathcal{C}_V$  for the points in  $\mathcal{V}_G$
4. Split the set  $\mathcal{V}_G$  as  $\tilde{\mathcal{V}} \cup \bar{\mathcal{V}}_L \cup \bar{\mathcal{V}}_{NL} \cup \hat{\mathcal{V}}$ 
  - $\tilde{\mathcal{V}} \in \mathfrak{F}(\mathcal{C}_V)$  and  $\mathcal{C}_V = \mathcal{C}_V \setminus \tilde{\mathcal{V}}$  (in words,  $\tilde{\mathcal{V}}$  contains those points in  $\mathcal{V}$  which lie on the frontier of  $\mathcal{C}_V$  but are not vertices);
  - $\bar{\mathcal{V}}_L \in \mathcal{V}_G \setminus \tilde{\mathcal{V}}$ ,  $\bar{\mathcal{V}}_L \in \mathfrak{F}(\mathcal{C}_V)$  and  $\bar{\mathcal{V}}_L$  saturates at least one linear constraint
  - $\bar{\mathcal{V}}_{NL} \in \mathcal{V}_G \setminus \tilde{\mathcal{V}}$ ,  $\bar{\mathcal{V}}_{NL} \in \mathfrak{F}(\mathcal{C}_V)$  and  $\bar{\mathcal{V}}_{NL}$  saturates only nonlinear constraints
  - $\hat{\mathcal{V}} \in \text{Int}(\mathcal{C}_V)$
5. Construct the dual representation of  $\mathcal{C}_V$ . This will be represented as a intersection of halfspaces  $\mathcal{H}$ .
6. Split  $\mathcal{H}$  in  $\bar{\mathcal{H}} \cup \hat{\mathcal{H}}$ 
  - $\hat{\mathcal{H}} \subset \mathcal{H}$  such that  $\exists x \in \mathcal{C}_V$  with  $\mathfrak{S}(\hat{\mathcal{H}}, x) \neq \emptyset$  and  $\mathfrak{B}(\mathfrak{A}_{NL}, x) \neq \emptyset$
  - $\bar{\mathcal{H}} = \mathcal{H} \setminus \hat{\mathcal{H}}$
7. Project the set

$$U = \left\{ \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \mid \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} = \begin{bmatrix} H^{-1}F \\ I \end{bmatrix} x, \forall x \in \mathbb{R}^n \right\}$$

on  $\mathcal{C}_V$ :

$$U^* \leftarrow \text{Proj}_{\mathcal{C}_V} U$$

8. **If**  $\exists x_0$  such that the point:

$$\begin{bmatrix} \mathbf{k}_u^* \\ x_0 \end{bmatrix} = U^* \cap \left\{ \begin{bmatrix} \mathbf{k}_u \\ x \end{bmatrix} \mid x = x_0 \right\}$$

saturates a subset of constraints

$$\mathcal{K}(x_0) = \mathfrak{S} \left( \mathcal{H}, \begin{bmatrix} \mathbf{k}_u \\ x_0 \end{bmatrix} \right) \subset \hat{\mathcal{H}}$$

**then:**

(a) Construct

$$U_{NL}(x_0) = \left\{ \left[ \begin{array}{c} \mathbf{k}_u \\ x \end{array} \right] \in U \mid \left[ \begin{array}{c} \mathbf{k}_u^* \\ x \end{array} \right] \in U^* \text{ s.t. } \mathfrak{S} \left( \mathcal{H}, \left[ \begin{array}{c} \mathbf{k}_u \\ x \end{array} \right] \right) = \mathcal{K}(x_0) \right\}$$

(b) Perform:

$$U^* = U^* \setminus \left\{ \left[ \begin{array}{c} \mathbf{k}_u \\ x \end{array} \right] \mid \mathfrak{S} \left( \mathcal{H}, \left[ \begin{array}{c} \mathbf{k}_u \\ x \end{array} \right] \right) = \mathcal{K}(x_0) \right\}$$

(c) Retain the set of points:

$$S = \left\{ v \in \widehat{\mathcal{V}} \mid \forall \left[ \begin{array}{c} \mathbf{k}_u \\ x \end{array} \right] \in \mathcal{C}_{\mathcal{V}} \quad \text{with } \mathfrak{S}(\widehat{\mathcal{H}}, \left[ \begin{array}{c} \mathbf{k}_u \\ x \end{array} \right]) = \mathcal{K}(x_0) \Rightarrow \mathfrak{B}(\mathfrak{R}_{NL}, x) = \mathfrak{S}(\mathfrak{R}_{NL}, v) \right\}$$

(d) Construct the Voronoi partition for the collection of points in  $S$

(e) Position  $U_{NL}(x_0)$  w.r.t. this partition and map the suboptimal solution

$U_{NL}^*(x_0) \leftarrow U_{NL}(x_0)$  by using the vertex  $v$  for each active region.

$$\left[ \begin{array}{c} \mathbf{k}_u^* \\ x \end{array} \right] = v \leftarrow \left[ \begin{array}{c} \mathbf{k}_u \\ x \end{array} \right]$$

*else*: jump to (10)

9. Return to point (8)

10. If the quality of the solution is not satisfactory, improve the distribution of the points  $\mathcal{V}_{\mathcal{G}}$  and restart from (2).

## 5 Numerical Example

Consider the MPC problem implemented using the first control action of the optimal sequence:

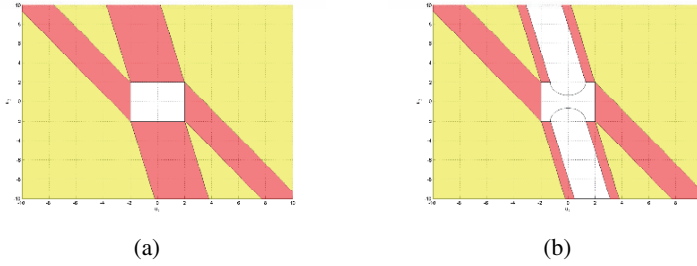
$$k_u^* = \arg \min_{k_u} \sum_{i=0}^{N-1} x_{t+k|t}^T Q x_{t+k|t} + u_{t+k|t}^T R u_{t+k|t} + x_{t+N|t}^T P x_{t+N|t} \quad (25)$$

$$\left\{ \begin{array}{l} x_{t+k+1|t} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_{t+k|t} + \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} u_{t+k|t} \quad k \geq 0 \\ \begin{bmatrix} -2 \\ -2 \end{bmatrix} \leq u_{t+k|t} \leq \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \quad \sqrt{(u_{t+k|t}^1)^2 + (u_{t+k|t}^2 \pm 2)^2} \geq \sqrt{3}; \quad 0 \leq k \leq N_u - 1 \\ u_{t+k|t} = \underbrace{\begin{bmatrix} 0.59 & 0.76 \\ -0.42 & -0.16 \end{bmatrix}}_{K_{LQR}} x_{t+k|t} \quad N_u \leq k \leq N_y - 1 \end{array} \right.$$

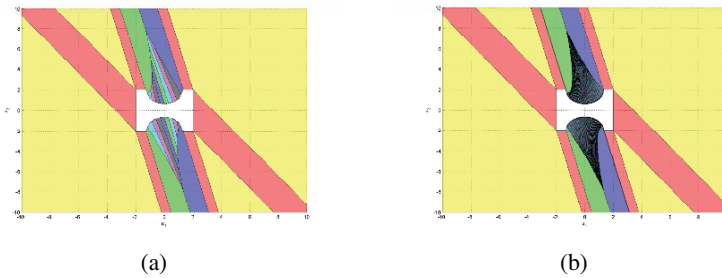
with

$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}; R = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}; P = \begin{bmatrix} 13.73 & 2.46 \\ 2.46 & 2.99 \end{bmatrix}; N_u = 1; N = 2$$

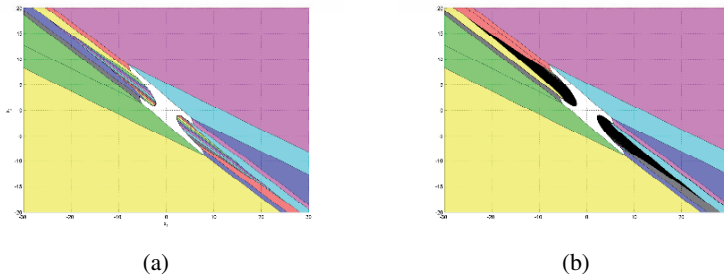
By following the previous algorithm, in the first stage, the partition of the state space is performed by considering only the linear constraints (figure 1(a)). Each such region



**Fig. 1.** a) Partition of the arguments space (linear constraints only). b) Retention of the regions with feasible linear projections.



**Fig. 2.** Partition of the arguments space (nonlinear case) - a) 10 points per active nonlinear constraint; b) 100 points per nonlinear constraint.

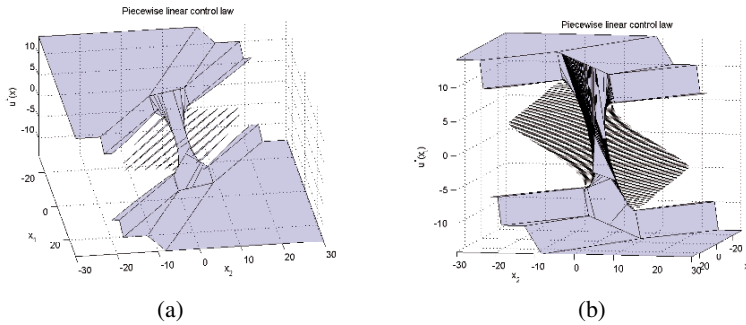


**Fig. 3.** Partition of the state space - a) 10 points per nonlinear constraint; b) 100 points per nonlinear constraint.

corresponds with a specific projection law. By simply verifying the regions where this projection law obeys the nonlinear constraints, the exact part of the explicit solution is obtained (figure 1(b)).

Further, a distribution of points on the nonlinear frontier of the feasible domain has to be obtained with the associated Voronoi partition. By superposing it to the regions non covered at the previous step a complete partition of the arguments space is realized. Figures 2(a)-2(a) depict such a partitions for 10 and 100 points for each nonlinear constraint.

By correspondence, the figures 3(a) and 3(b) describe the partition of the state space for the explicit solution. Finally the complete explicit solution for the two cases are



**Fig. 4.** Explicit control law - a) 10 points per nonlinear constraint; b) 100 points per nonlinear constraint.

described in figures 4(a) and 4(b). The discontinuities are observable as well as the increase in resolution over the nonlinearity with the augmentation of the number of points in the Voronoi partition. In order to give an image of the complexity it must be said that the explicit solutions have 31 and 211 regions respectively and the computational effort was less than  $2s$  in the first case and  $80s$  in the second case, mainly spent in the construction of supplementary regions in the Voronoi partition.

## 6 Conclusions

The parameterized polyhedra offer a transparent characterization of the MPC degrees of freedom. Once the complete description of the feasible domain as a parameterized polyhedron is obtained, explicit MPC laws can be constructed using the projection of the unconstrained optimum. The topology of the feasible domain can lead to explicit solution even if nonlinear constraints are taken into consideration. The price to be paid is found in the degree of suboptimality.

## References

1. Sokaert, P.O., Mayne, D.Q., Rawlings, J.B.: Suboptimal model predictive control (feasibility implies stability). In: IEEE Transactions on Automatic Control. Volume 44. (1999) 648–654
2. Olaru, S., Dumur, D.: A parameterized polyhedra approach for explicit constrained predictive control. (In: 43rd IEEE Conference on Decision and Control, 2004.) 1580–1585 Vol.2
3. Motzkin, T.S., R.H.T.G., R.M., T.: The Double Description Method, republished in *Theodore S. Motzkin: Selected Papers*, (1983). Birkhauser (1953)
4. Schrijver, A.: Theory of Linear and Integer Programming. John Wiley and Sons, NY (1986)
5. Leverage, H.: A note on chernikova's algorithm. In: Technical Report 635, IRISA, France (1994)
6. Loechner, V., Wilde, D.K.: Parameterized polyhedra and their vertices. International Journal of Parallel Programming **V25** (1997) 525–549
7. Olaru, S., Dumur, D.: Compact explicit mpc with guarantee of feasibility for tracking. In: 44th IEEE Conference on Decision and Control, and European Control Conference. (2005) 969–974

8. Seron, M., Goodwin, G., Dona, J.D.: Characterisation of receding horizon control for constrained linear systems. In: *Asian Journal of Control*. Volume 5. (2003) 271–286
9. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.: The Explicit Linear Quadratic Regulator for Constrained Systems. *Automatica* **38** (2002) 3–20
10. Goodwin, G., Seron, M., Dona, J.D.: *Constrained Control and Estimation*. Springer, Berlin (2004)
11. Borelli, F.: *Constrained Optimal Control of Linear and Hybrid Systems*. Springer-Verlag, Berlin (2003)
12. Tondel, P., Johansen, T., Bemporad, A.: Evaluation of piecewise affine control via binary search tree. *Automatica* **39** (2003) 945–950
13. Bemporad, A., Borrelli, F., Morari, M.: Robust Model Predictive Control: Piecewise Linear Explicit Solution. In: *European Control Conference*. (2001) 939–944
14. Kerrigan, E., Maciejowski, J.: Feedback min-max model predictive control using a single linear program: Robust stability and the explicit solution. *International Journal of Robust and Nonlinear Control* **14** (2004) 395–413
15. Olaru, S., Dumur, D.: On the continuity and complexity of control laws based on multiparametric linear programs. In: *45th IEEE Conference on Decision and Control*. (2006)
16. Grancharova, A., Tondel, P., Johansen, T.A.: *International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control*. (2005)



## Author Index

Aeyels, D. ....	195	Lechner, D. ....	125
Alici, G. ....	109	Lhommeau, M. ....	265
Auer, E. ....	139	Li, X. ....	181
Azorín, J. ....	169	Li, Z. ....	109
Baffet, G. ....	125	Luther, W. ....	139
Balmat, J.-F. ....	37	Mahieu, B. ....	23
Brown, C. ....	273	McLoone, S. ....	273
Charara, A. ....	125	Mikaelian, A. ....	51
Cook, C. ....	109	Millot, P. ....	3
Delanoue, N. ....	265	Montmain, J. ....	23
Dobre, S. ....	301	Morales, L. ....	253
Duin, S. ....	109	Nikolos, I. ....	153
Dumur, D. ....	301	Olaru, S. ....	301
García, N. ....	169	Olguín-Díaz, E. ....	207
Gelly, S. ....	95	Parra-Veja, V. ....	207
Gómez, A. ....	75	Pérez, C. ....	169
Gracia, L. ....	169	Pessel, N. ....	37
Hanebeck, U. ....	239, 287	Rogge, J. ....	195
Hardouin, L. ....	265	Röning, J. ....	87
Huber, M. ....	239	Sabater, J. ....	169
Hung, P. ....	273	Sanchez, C. ....	23
Irwin, G. ....	273	Schrempf, O. ....	287
Jaulin, L. ....	265	Simas, E. ....	75
Jouandeau, N. ....	63	Stocco, L. ....	225
Juutilainen, I. ....	87	Teytaud, O. ....	95
Kee, R. ....	273	Thomas, D. ....	125
Koskimäki, H. ....	87	Tsourveloudis, N. ....	153
Kryzhanovsky, B. ....	51	Vinches, M. ....	23
Kryzhanovsky, V. ....	51	Weissel, F. ....	239
Lafont, F. ....	37	Yedlin, M. ....	225
Laurinen, P. ....	87	Zell, A. ....	181