Lecture Notes in Electrical Engineering 325

Jean-Louis Ferrier
Oleg Gusikhin
Kurosh Madani
Jurek Sasiadek
Editors

Informatics in Control, Automation and Robotics

10th International Conference, ICINCO 2013 Reykjavík, Iceland, July 29–31, 2013 Revised Selected Papers

Springer

Lecture Notes in Electrical Engineering

Volume 325

About this Series

"Lecture Notes in Electrical Engineering (LNEE)" is a book series which reports the latest research and developments in Electrical Engineering, namely:

- Communication, Networks, and Information Theory
- Computer Engineering
- Signal, Image, Speech and Information Processing
- Circuits and Systems
- Bioengineering

LNEE publishes authored monographs and contributed volumes which present cutting edge research information as well as new perspectives on classical fields, while maintaining Springer's high standards of academic excellence. Also considered for publication are lecture materials, proceedings, and other related materials of exceptionally high quality and interest. The subject matter should be original and timely, reporting the latest research and developments in all areas of electrical engineering.

The audience for the books in LNEE consists of advanced level students, researchers, and industry professionals working at the forefront of their fields. Much like Springer's other Lecture Notes series, LNEE will be distributed through Springer's print and electronic publishing channels.

More information about this series at http://www.springer.com/series/7818

Jean-Louis Ferrier · Oleg Gusikhin
Kurosh Madani · Jurek Sasiadek
Editors

Informatics in Control, Automation and Robotics

10th International Conference, ICINCO 2013 Reykjavík, Iceland, July 29–31, 2013 Revised Selected Papers

Springer

Editors
Jean-Louis Ferrier
Institute of Science and Technology
University of Angers
Angers
France

Oleg Gusikhin
Ford Research and Advanced Engineering
Dearborn, MI
USA

Kurosh Madani
University Paris-Est Créteil (UPEC)
Créteil
France

Jurek Sasiadek
Mechanical and Aerospace Engineering
Carleton University
Ottawa, ON
Canada

ISSN 1876-1100 ISSN 1876-1119 (electronic)
ISBN 978-3-319-10890-2 ISBN 978-3-319-10891-9 (eBook)
DOI 10.1007/978-3-319-10891-9

Library of Congress Control Number: 2013956806

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Organization

Conference Chair

Jean-Louis Ferrier, University of Angers, France

Program Co-Chairs

Oleg Gusikhin, Ford Research and Advanced Engineering, USA
Kurosh Madani, University of Paris-EST Créteil (UPEC), France
Jurek Sasiadek, Carleton University, Canada

Organizing Committee

Marina Carvalho, INSTICC, Portugal
Helder Coelhas, INSTICC, Portugal
Bruno Encarnação, INSTICC, Portugal
Ana Guerreiro, INSTICC, Portugal
André Lista, INSTICC, Portugal
Filipe Mariano, INSTICC, Portugal
Andreia Moita, INSTICC, Portugal
Raquel Pedrosa, INSTICC, Portugal
Vitor Pedrosa, INSTICC, Portugal
Cláudia Pinto, INSTICC, Portugal
Cátia Pires, INSTICC, Portugal
Ana Ramalho, INSTICC, Portugal
Susana Ribeiro, INSTICC, Portugal
Rui Rodrigues, INSTICC, Portugal

Sara Santiago, INSTICC, Portugal
André Santos, INSTICC, Portugal
Fábio Santos, INSTICC, Portugal
Mara Silva, INSTICC, Portugal
José Varela, INSTICC, Portugal
Pedro Varela, INSTICC, Portugal

Program Committee

El-Houssaine Aghezzaf, Belgium
Eugenio Aguirre, Spain
Hyo-Sung Ahn, Republic of Korea
Adel Al-Jumaily, Australia
Fouad Al-sunni, Saudi Arabia
Job Van Amerongen, The Netherlands
Stefan Andrei, USA
Peter Arato, Hungary
Rui Araujo, Portugal
Helder Araújo, Portugal
Alejandro Hernandez Arieta, Switzerland
Tomas Arredondo, Chile
Vijanth Sagayan Asirvadam, Malaysia
T. Asokan, India
Ali Bab-Hadiashar, Australia
Jacky Baltes, Canada
Ruth Bars, Hungary
Victor Becerra, UK
Laxmidhar Behera, India
Karsten Berns, Germany
Arijit Bhattacharya, Ireland
Robert Bicker, UK
Mauro Birattari, Belgium
Jean-louis Boimond, France
Magnus Boman, Sweden
Thomas Braunl, Australia
Mietek Brdys, UK
Glen Bright, South Africa
Kevin Burn, UK
Amaury Caballero, USA
Javier Fernandez de Canete, Spain
Giuseppe Carbone, Italy
Alessandro Casavola, Italy
Riccardo Cassinis, Italy

Yangquan Chen, USA
Albert Cheng, USA
Tsung-Che Chiang, Taiwan
Sung-Bae Cho, Republic of Korea
Ryszard S. Choras, Poland
Carlos Coello Coello, Mexico
James M. Conrad, USA
Yechiel Crispin, USA
José Boaventura Cunha, Portugal
Fabrizio Dabbene, Italy
Prithviraj (Raj) Dasgupta, USA
Mingcong Deng, Japan
Guilherme DeSouza, USA
Rüdiger Dillmann, Germany
António Dourado, Portugal
Venky Dubey, UK
Ashish Dutta, India
Marc Ebner, Germany
Petr Ekel, Brazil
Mohammed El-Abd, Kuwait
Eniko Enikov, USA
Ali Eydgahi, USA
Simon G. Fabri, Malta
David Fernández-Llorca, Spain
Jean-Louis Ferrier, France
Paolo Fiorini, Italy
Juan J. Flores, Mexico
Mauro Franceschelli, Italy
Heinz Frank, Germany
Georg Frey, Germany
S.G. Ponnambalam, Malaysia
John Qiang Gan, UK
Nicholas Gans, USA
Maria I. Garcia-Planas, Spain
Ryszard Gessing, Poland
Lazea Gheorghe, Romania
Paulo Gil, Portugal
Luis Gomes, Portugal
Bhaskaran Gopalakrishnan, USA
Frans C.A. Groen, The Netherlands
Da-Wei Gu, UK
Jason Gu, Canada

Kevin Guelton, France
Oleg Gusikhin, USA
Thomas Gustafsson, Sweden
Quang Phuc Ha, Australia
Maki K. Habib, Egypt
Wolfgang Halang, Germany
Jennifer Harding, UK
Victor Hinostroza, Mexico
Kaoru Hirota, Japan
Wladyslaw Homenda, Poland
Hesuan Hu, China
Atsushi Imiya, Japan
Sarangapani Jagannathan, USA
Ray Jarvis, Australia
Thira Jearsiripongkul, Thailand
Myong K. Jeong, USA
Ivan Kalaykov, Sweden
Mansour Karkoub, Qatar
Dusko Katic, Serbia
Kazuo Kiguchi, Japan
DaeEun Kim, Republic of Korea
Jonghwa Kim, Germany
Ashok K. Kochhar, UK
Waree Kongprawechnon, Thailand
Israel Koren, USA
Krzysztof Kozlowski, Poland
Mianowski Krzysztof, Poland
Masao Kubo, Japan
Kin Keung Lai, Hong Kong
H.K. Lam, UK
Alexander Lanzon, UK
Kathryn J. De Laurentis, USA
Graham Leedham, Australia
Kauko Leiviskä, Finland
Kang Li, UK
Tsai-Yen Li, Taiwan
Yangmin Li, China
Youfu Li, Hong Kong
Gordon Lightbody, Ireland
Huei-Yung Lin, Taiwan
Józef Lisowski, Poland
Changchun Liu, USA

Guoping Liu, UK
Luís Seabra Lopes, Portugal
Gonzalo Lopez-Nicolas, Spain
Iuan-Yuan Lu, Taiwan
Edwin Lughofer, Austria
Martin Lukac, Japan
José Tenreiro Machado, Portugal
Anthony Maciejewski, USA
Kurosh Madani, France
Nitaigour Mahalik, USA
Frederic Maire, Australia
Om Malik, Canada
Fabio Marchese, Italy
Philippe Martinet, France
Rene V. Mayorga, Canada
Ross McAree, Australia
Seán McLoone, Ireland
Claudio Melchiorri, Italy
Konstantinos Michail, Cyprus
Patrick Millot, France
Sanya Mitaim, Thailand
Francesco Carlo Morabito, Italy
Vladimir Mostyn, Czech Republic
Zoltan Nagy, UK
Saeid Nahavandi, Australia
Andreas Nearchou, Greece
Sergiu Nedevschi, Romania
Robert W. Newcomb, USA
Klas Nilsson, Sweden
Juan A. Nolazco-Flores, Mexico
Andrzej Obuchowicz, Poland
José Valente de Oliveira, Portugal
H.R.B. Orlande, Brazil
Christos Panayiotou, Cyprus
Igor Paromtchik, France
Bozenna Pasik-Duncan, USA
Pierre Payeur, Canada
Marco Antonio Arteaga Perez, Mexico
Jeff Pieper, Canada
Selwyn Piramuthu, USA
Angel P. Del Pobil, Spain
Marie-Noëlle Pons, France

Libor Preucil, Czech Republic
José Ragot, France
Robert Richardson, UK
Mihailo Ristic, UK
Rodney Roberts, USA
Eduardo Rocon, Spain
Juha Röning, Finland
Mehdi Roopaei, USA
Agostinho Rosa, Portugal
João Mauricio Rosário, Brazil
Danilo De Rossi, Italy
António Ruano, Portugal
Carlos Sagüés, Spain
Mehmet Sahinkaya, UK
Antonio Sala, Spain
Abdel-Badeeh Mohamed Salem, Egypt
Marcello Sanguineti, Italy
Elena De Santis, Italy
Nilanjan Sarkar, USA
Jurek Sasiadek, Canada
Hideyuki Sawada, Japan
Daniel Sbarbaro, Chile
Klaus Schilling, Germany
Carla Seatzu, Italy
Gerald Seet, Singapore
João Sequeira, Portugal
Michael Short, UK
Dan Simon, USA
Olivier Simonin, France
Adam Slowik, Poland
Andrzej Sluzek, UAE
Jae-Bok Song, Republic of Korea
Qing Song, Singapore
Safeeullah Soomro, Pakistan
Stefano Squartini, Italy
Burkhard Stadlmann, Austria
Sergiu Stan, Romania
Tarasiewicz Stanislaw, Canada
A. Frank Van Der Stappen, The Netherlands
Olaf Stursberg, Germany
Raúl Suárez, Spain
Kamesh Subbarao, USA

Katia Sycara, USA
Kazuya Takeda, Japan
Tianhao Tang, China
Daniel Thalmann, Singapore
Gui Yun Tian, UK
Sven Tomforde, Germany
N.G. Tsagarakis, Italy
Antonios Tsourdos, UK
Ali Emre Turgut, Belgium
Dariusz Ucinski, Poland
Angel Valera, Spain
Bram Vanderborght, Belgium
Ramiro Velazquez, Mexico
Damir Vrancic, Slovenia
Bernardo Wagner, Germany
James Whidborne, UK
Sangchul Won, Republic of Korea
Xun W. Xu, New Zealand
Bin Yao, USA
Marek Zaremba, Canada
Primo Zingaretti, Italy
Loredana Zollo, Italy
Argyrios Zolotas, UK

Auxiliary Reviewers

Fathi Abugchem, UK
José Baca, USA
Sotirios Diamantas, USA
Siavash Farzan, USA
Felix Felgner, Germany
Isma Hadji, USA
Akshay Jain, USA
Agnieszka Jastrzebska, Poland
Carl-Johan Jorgensen, France
Nicole Lazzeri, Italy
Mudassir Lone, UK
Federico Lorussi, Italy
Eduardo Montijano, Spain
Jesper Pedersen Notander, Sweden

Mobolaji Osinuga, UK
Rafael Pastor, Spain
Jinglin Shen, USA
Gaetano Valenza, Italy
Abolfazl Zaraki, Italy

Invited Speakers

Luis Paulo Reis, University of Minho, Portugal
Krzysztof Tchon, Wroclaw University of Technology, Poland
Klaus Schilling, University Würzburg, Germany
Libor Kral, Head of Unit, European Commission, DG CONNECT, Unit A2—Robotics, Luxembourg

Preface

This book includes extended and revised versions of a set of selected papers from the 10th International Conference on Informatics in Control Automation and Robotics (ICINCO 2013), held in Reykjavíc, Iceland, from 29 to 31 July 2013. The conference was organized into four simultaneous tracks such as Intelligent Control Systems and Optimization, Robotics and Automation, Systems Modelling, Signal Processing and Control and Industrial Engineering, and Production and Management.

ICINCO 2013 received 255 paper submissions, from 50 countries in all continents. From these, after a blind review process, 30 % were published and presented orally, from which 22 papers 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 as well as the logic equilibrium of the four tracks mentioned above. 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 2014, to be held in Vienna, Austria, and in future editions of this conference.

Furthermore, ICINCO 2013 included four plenary keynote lectures given by Luis Paulo Reis (University of Minho, Portugal), Krzysztof Tchon (Wroclaw University of Technology, Poland), Klaus Schilling (University Würzburg, Germany), and Libor Kral (Head of Unit, European Commission, DG CONNECT, Unit A2—Robotics, Luxembourg). We would like to express our appreciation to all of them and in particular to those who took the time to contribute to this book with a paper.

On behalf of the conference organizing committee, we would like to thank all participants. First all the authors, whose quality work is the essence of the conference and the members of the Program Committee, who helped us with their valuable expertise and diligence in reviewing the 255 received papers. As we all know, organizing a conference requires the effort of many individuals. We also wish to thank all the members of our organizing committee, whose work and commitment were invaluable. Aiming to provide the potential readers with an objective overview of the latest advances in the four major topics of the conference mentioned above, we hope that this book will be relevant for all researchers and practitioners whose work is related to using informatics in control, robotics, or automation.

December 2013

Jean-Louis Ferrier
Oleg Gusikhin
Kurosh Madani
Jurek Sasiadek

Contents

Part II Robotics and Automation

Part III Signal Processing, Sensors, Systems Modeling and Control

Part IV Industrial Engineering, Production and Management

Chapter 1
Invited Paper: Multimodal Interface for an Intelligent Wheelchair

Luís Paulo Reis, Brígida Mónica Faria, Sérgio Vasconcelos and Nuno Lau

Abstract Since the demographics of population, with respect to age, are continuously changing, politicians and scientists start to pay more attention to the needs of senior individuals. Additionally, the well-being and needs of disabled individuals are also becoming highly valued in the political and entrepreneurial society. Intelligent wheelchairs are adapted electric wheelchairs with environmental perception, semi-autonomous behaviour and flexible human-machine-interaction. This paper presents the specification and development of a user-friendly multimodal interface, as a component of the IntellWheels Platform project. The developed prototype combines several input modules, allowing the control of the wheelchair through flexible user defined input sequences of distinct types (speech, facial expressions, head

L.P. Reis (✉)
Departamento de Sistemas de Informação,
Escola de Engenharia da Universidade do Minho (DSI/EEUM),
Guimarães, Portugal
e-mail: lpreis@dsi.uminho.pt

L.P. Reis · B.M. Faria · S. Vasconcelos
Laboratório de Inteligência Artificial e Ciência de Computadores (LIACC),
Porto, Portugal

B.M. Faria
Escola Superior Tecnologia de Saúde do Porto,
Instituto Politécnico do Porto (ESTSP/IPP), Porto, Portugal
e-mail: btf@estsp.ipp.pt

S. Vasconcelos
Departamento de Engenharia Informática,
Faculdade de Engenharia da Universidade do Porto (DEI/FEUP),
Porto, Portugal
e-mail: ei05074@fe.up.pt

N. Lau
Departamento de Engenharia Eletrónica Telecomunicações e Informática,
Universidade de Aveiro (DETI/UA), Aveiro, Portugal
e-mail: nunolau@ua.pt

N. Lau
Instituto de Engenharia Eletrónica e Telemática de Aveiro (IEETA),
Aveiro, Portugal

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_1

movements and joystick). To validate the effectiveness of the prototype, two experiments were performed with a number of individuals who tested the system firstly by driving a simulated wheelchair in a virtual environment. The second experiment was performed using the real IntellWheels wheelchair prototype. The results achieved proved that the multimodal interface may be successfully used by people, due to the interaction flexibility it provides.

Keywords Multimodal interface · Intelligent robotics · Intelligent wheelchair · IntellWheels

1.1 Introduction

As the quality of life increases, also the average of population life tends to augment. According to data provided by the United Nations and the World Health Organization, a portion of 650 million people (10 % of world population) have incapacity, and 20 % of them have physical disabilities. Moreover, these numbers have been continuously increasing since the world population is growing and ageing. Other factors include environment degradation, sub nutrition and the appearance of chronic health issues. The main reasons for physical disability range from traffic accidents, war motives and landmines, to falls. Physical injuries could also be caused by medical conditions, like cerebral palsy, multiple sclerosis, respiratory and circulatory diseases, genetic diseases or chemical and drugs exposure [1].

The main objectives of this work are related to the study and development of an intelligent wheelchair (IW), based on the adaptation of a commercial electric wheelchair with flexible hardware. Sensorial and control system implementation, as well as navigational methods and multimodal interface design are the core research focus of this project. Another important objective is the minimization of the aesthetic characteristics of the wheelchairs, to assure the well-being of the patients while driving them. This paper is related to a multimodal interface (IMI) integrated in the intelligent wheelchairs of the IntellWheels project developed at LIACC (Artificial Intelligence and Computer Science Laboratory) [2] and with the collaboration of the INESC-TEC, IEETA, University of Porto, University of Aveiro, University of Minho, School of Allied Health Sciences of Porto and the Cerebral Palsy Association of Porto.

This paper starts with a brief overview of some human-machine-interaction recognition methods liable to be used in a project of this nature. Secondly, a short description of multimodal interfaces and related concepts is given. The fourth section of this paper is related with the IntellWheels project description and the multimodal interface created and implemented for this project. Following, a description of the performed experiments and achieved results is made. The last part of this paper presents the paper main conclusion, and some pointers for future work.

1.2 Human Machine Interaction

This section contains an overview of some low cost input devices that can be used in human computer interaction, as well as some recognition methods used to extract user intentions. Besides keyboard and mouse [3] there are other input devices that can be used in human machine interaction, such as microphone, video camera, joystick, tongue mouse, accelerometer, gyroscope, data gloves or even brain headsets.

1.2.1 Video based Systems

Video based systems can be used to recognize different types of human body motion. There are many publications [4] related to video based recognizers, ranging from head movements, facial expressions, mouth recognition, and other. Some intelligent wheelchair projects use video based systems to capture the facial expressions, gaze and position and orientation of the patients' head to control the wheelchair.

Facial expression analysis is concerned with the development of computer systems that can automatically recognize facial motions and feature changes from visual information [5]. The facial action coding system (FACS) is the most widely used method to measure facial movement. FACS defines 32 action units, each one representing a specific contraction or relaxation of one or more muscles of the face [6]. A number of intelligent wheelchair projects presented human-machine interfaces especially designed for quadriplegic individuals, by using facial expressions recognition as the main input to drive the wheelchair [7]. A survey on recent techniques for detecting facial expressions using Artificial Neural Networks, Hidden Markov Models and Support Vector Machines can be consulted in [8].

The first methods used for eye tracking took advantage of Electro-oculography (EOG), which is a biological measurement technique used to determine the resting potential of the retina. An example of a system using this technique is Eagle Eyes [9] (Gips et al., 1996), used in the Wheelesley wheelchair project. However, improvements in video recording and computer vision technology allowed the development of less invasive systems to track the user's eye motions [10]. Tall and Alapetite [11] presented a study about gaze-controlled driving using video-based systems to track the user's eyes gaze orientation.

Some video based systems capture the orientation and position of the user's head [12], which can be used as an input to represent specific desired outputs. This technique has been used in some wheelchair projects, namely at Osaka University [13]. In the WATSON wheelchair [14], a video system is used to detect the user's gaze and face orientation.

Also mouth recognition is used to associate the users' intention to obtain desired outputs. This method uses pattern recognition techniques to detect certain shapes of the mouth which act as input tokens. Shin and Kim [15] proposed a novel IW interface using face and mouth recognition for severely disabled individuals. The direction of the wheelchair's movement changes according to the user's face inclination. By changing the shape of the mouth, the user can make the wheelchair move forward or stop.

1.2.2 Speech Recognition

Speech recognition is the process of a computer to decode human spoken words to binary code comprehensible by the computer. The goal of speech recognition was to promote independence since it is used to convert human speech signals into effective actions [16]. In many cases, speech is the only way individuals with disabilities can communicate. Youdin [17] presented the first wheelchair system activated by voice. Equipped with an environmental control unit (ECU), users could control a number of devices, namely telephone, radio, fans and page-turner, using voice commands. This approach received positive feedback from a group of individuals with cerebral palsy, who preferred the use of voice commands instead of breath control systems [16]. As the robustness of the available speech recognition systems was improved during the last years, the widespread availability of this low-cost technology was used in many other intelligent wheelchair projects such as NavChair, RobChair, Senario, Tetra-Nauta, MIT Wheelchair.

Although today speech recognition can be used with satisfactory effectiveness in many situations, it still shows to be flawed when the surrounding environment is noisy. Other situations include cases where the user's voice does not match the training data, or when the user cannot achieve proper speaker adaptation. Sasou and Kojima [18] proposed a noise robust speech recognition applied to a voice-driven wheelchair. Their approach consists of using an array of microphones installed on the wheelchair, unlike the common method of using a singular head microphone placed close to the user's mouth. The achieved accuracy is very similar to the headset microphone method, and has the advantage of avoiding situations where the headset microphone position must be adjusted by the user. For hand disabled individuals, who are one of the major users of this wheelchair, this represents an interesting approach.

1.2.3 Gesture Recognition

Gesture recognition is the interpretation of a human gesture by a computing device. Gesture recognition composes an alternative for disabled individuals to interact with computing devices. Hand and finger gestures can be recognized using sensors such as accelerometers and gyroscopes CyberGlove [19, 20]. Alternatively the computing device can be equipped with a camera so that specific software can recognize and interpret the gestures [21].

1.2.4 Thought Recognition

A brain-computer interface is a type of device which allows interaction between users and computer systems, through the recognition of brainwave activity. Normally, brain-computer interfaces are used in medical contexts, with the objectives of augmenting cognitive and sensory-motor functions. There are two types of brain-computer interfaces. Invasive and partially-invasive BCIs require medical and surgical intervention, since they are implanted in the user's brain. On the other hand, non-invasive BCIs do not require brain implants (Electroencephalography). However, non-invasive BCIs are less effective when compared to invasive BCIs, since the obtainable signal of brainwave activity is weaker. The idea of integrating brain computer interfaces in intelligent wheelchairs is already present in the literature LURCH project [22]. It uses a non-invasive BCI that allows the user to drive the wheelchair.

The Maia project is a European project aiming at the development of an electroencephalography-based brain-computer interface for controlling an autonomous wheelchair [23]. The wheelchair control has automatic obstacle avoidance and is also capable of following walls. The user can control the wheelchair movement giving commands such as "go back" or "go right". The authors of [24] propose a slightly different approach for a user to drive an intelligent wheelchair, using a BCI. Instead of performing high-level commands, the user must continuously drive the wheelchair. Another project under development at the National University of Singapore consists of an autonomous wheelchair controlled through a P300-based BCI [25]. The wheelchair movements are limited to predefined paths. The user selects a destination, and the wheelchair automatically calculates the trajectory to the desired place. If an unexpected situation occurs, the wheelchair stops and waits for further commands.

Unfortunately, some problems may arise while trying to use a BCI. Brain activity varies greatly from individual to individual, and a person's brain activity also changes substantially over time [26]. These obstacles make it difficult to develop systems that can easily understand the user intentions, especially for long periods of time. Also, long periods of training are necessary before a user can correctly use a BCI to control a specific device [26].

1.2.5 Sip and Puff

Sip-and-puff is another method used by hand disabled individuals to control devices such as wheelchairs, by sipping and puffing on a straw. It is mainly used by people who do not have the use of their hands. These mechanisms are often used by individuals with severe disabilities, such as quadriplegia. Sip-and-puff technology can also be used to control mouse movement in a simple and efficient fashion. Additionally, using sip-and-puff together with scanning software allows disabled users to use programs accessible by keyboard. Some wheelchairs use sip and puff technology to aid in the navigation [27].

1.3 Multimodal Interfaces

“Multimodal interfaces process two or more combined user input modes, such as speech, pen, touch, manual gestures and gaze, in a coordinated fashion with multimedia system output” [28]. The interaction style provided by multimodal interfaces permits users to have a higher flexibility to combine modalities of inputs, or to switch from one input to another that may be better suited for a particular task or setting. For disabled individuals this interface paradigm can be a potential benefit.

The applications of multimodal systems is wide, ranging from virtual reality systems used for simulation and training, to biometric identification systems, and also medical and educational purposes [28]. In more recent years, recreational technology such as video game systems and cell phones has also been adopting the multimodal paradigm. Also, multimodal systems depend on synchronized parallel processing, since multiple recognizers are needed to interpret multimodal input. Additionally, the time sensitivity of these types of systems is crucial to distinguish between parallel or in sequence multimodal commands [29].

Several projects are involved in presenting multimodal interfaces. A classic example of a multimodal interface is the Media Room demonstration system. This system, presented by [30], composed one of the first attempts to combine speech and gesture recognition. Users could create objects on a map, and define their attributes, such as colour and size. Additionally, by saying “Put that there”, users could move objects positions on the map, firstly by pointing to the specific object, followed by pointing to the desired destination.

MATCHKiosk is a multimodal interactive city guide allowing users to access information of New York City subways and restaurants. Using pen input, users can interact with the interface by drawing on the display. It also allows synchronous combination of inputs, using both speech and pen input to ask detailed information to the system. The information is presented graphically, synchronized with a text-to-speech synthesizer [31].

Another widely known map-based application, QuickSet [32], also makes use of speech and gesture input. This military-training system allows users to draw out with a pen at a given position on the map, using predefined symbols to create new platoons. As an alternative, users can use voice commands to create such platoons, being also able to specify the new position for the platoon vocally. Additionally, users can also express the intent of creating a new platoon using voice recognition, while pointing with the pen the desired location on the map [32].

An example of an assistive multimodal interface intended for persons with hands and arm disabilities is Intellectual Computer AssistaNt for Disabled Operators (ICANDO). This multimodal interface has a speech recognizer for English, Russian and French. In combination with a head tracking module, the system enables hands-free interaction with a graphical user interface in a number of tasks, namely manipulation of graphical and text documents [33].

OpenInterface is an interesting project which goal is the development of an open source platform to enable rapid prototyping development of multimodal interactive systems, based on the reutilization of components developed by several research labs [34].

1.4 IntellWheels Project

The IntellWheels project main objective is the creation of a development platform for intelligent wheelchairs [35], entitled IntellWheels Platform (IWP). The project main focus is the research and design of a multi-agent platform, enabling easy integration of different sensors, actuators, devices for extended interaction with the user [36], navigation methods and planning techniques and methodologies for intelligent cooperation to solve problems associated with intelligent wheelchairs [37].

1.4.1 IntellWheels Platform

It is expected that this platform may facilitate the development and testing of new methodologies and techniques relating to intelligent wheelchairs. Within this concept, an attempt has been made at creating a set of tools (software and hardware) that can be easily integrated into any powered wheelchair, commercially available, with minor modifications. The project also takes into consideration issues such as low cost maintenance, and the preservation of the patient comfort, as well as the original wheelchair ergonomics.

Some relevant capabilities of an intelligent wheelchair can be enumerated: intelligent planning, autonomous navigation and semi-autonomous navigation using

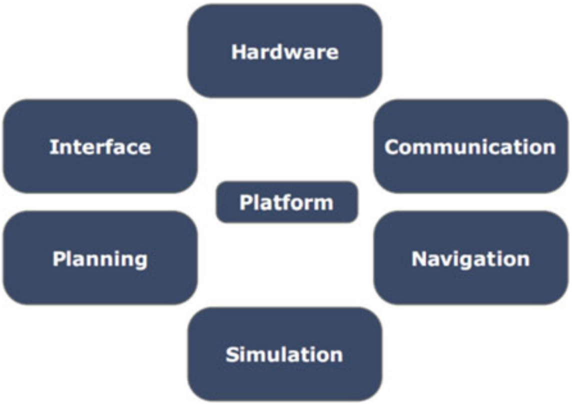


Fig. 1.1 IntellWheels Basic modules

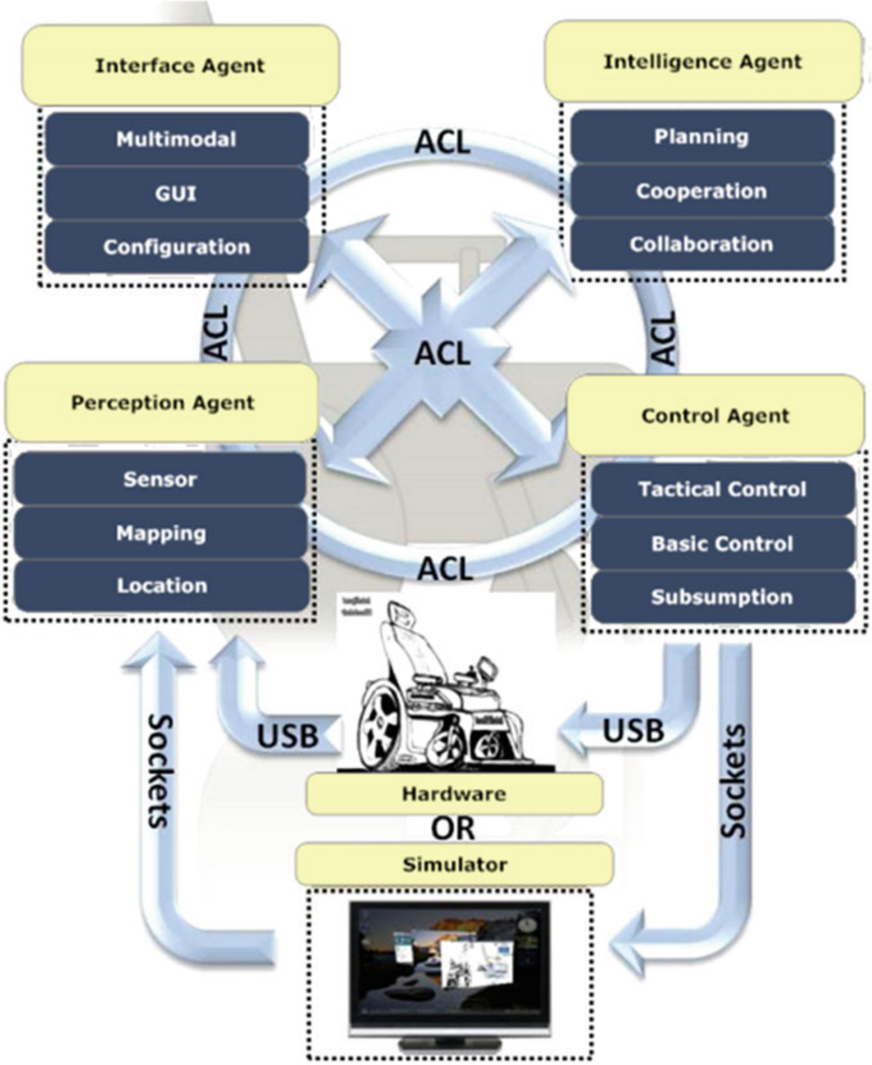


Fig. 1.2 Intellwheels software architecture

commands given by the user (patient) with a high level language. These capabilities can be achieved through an advanced control system, ranging from a simple shared control (in which the system can prevent crashes when the user is manually controlling the wheelchair) to the control of more complex tasks using high level commands (in this case, commands interpreted by a multimodal interface, automatic planning, autonomous driving and environment mapping). The system was designed with six basic modules, illustrated in Fig. 1.1, namely: planning, interface, simulation, communication, navigation and hardware.

1.4.1.1 Software Architecture

Figure 1.2 represents the software architecture proposed in this project. This architecture uses the paradigm of Multi-Agent Systems (MAS). The decision of using the MAS paradigm was due to its flexibility, in order to subserve the addition of new modules and the interaction between the wheelchairs, and other intelligent devices. In this architecture one can observe four basic agents, which compose a single wheelchair:

- Interface Agent: responsible for the interaction between the patient and the wheelchair;
- Intelligence Agent: responsible for the planning actions of the wheelchair;
- Perception Agent: responsible for reading the adequate sensor values for each context, location and environment mapping;
- Control Agent: responsible for the control activities of basic actions, wheel control, and obstacle avoidance.

These agents are heterogeneous and can collaborate with other agents of another wheelchair. It can be observed in the architecture, that the previously described agents can share the control of both a real wheelchair and a simulated wheelchair.

One of the most innovative features of the platform is the use of the mixed reality concept, which allows the interaction between a real intelligent wheelchair with virtual objects and virtual wheelchairs. This possibility of interaction makes it possible for high complexity testing using large sets of devices and wheelchairs, decreasing the costs associated with the development of new technologies, since there is no need to build a large set of real intelligent wheelchairs with an expensive and complex infrastructure. With this feature, it is possible to analyze and evaluate the interaction between a large number of IW, by setting an environment composed by a real wheelchair and a number of virtual wheelchairs.

The interaction between the agents in this scenario would result in the same interaction of a scenario composed by real wheelchairs only, once the involved agents are the same. The only basic difference would be the presence of the robot's body. Figure 1.3 illustrates the combination of the possible different operation modes provided by the platform.

The Intellwheels platform allows the system to work in real mode (the IW has a real body), simulated (the body of the wheelchair is virtual) or mixed reality (real IW with perception of real and virtual objects). In real mode it is necessary to connect the system (software) to the IW hardware. In the simulated mode, the software is connected to the IWP simulator. In the mixed reality mode, the system is connected to both (hardware and simulator).

1.4.1.2 Hardware Platform

For a wheelchair to be considered intelligent, it should be able to understand the surrounding environment, plan their actions, react to environmental changes and provide an expanded interface for optimized user interaction. The interface should provide means of recognizing the users' intentions in a flexible and configurable fashion, and turn them into navigation commands that are subsequently processed by the control. To meet these characteristics, a hardware kit has been developed, composed by low cost devices that can be easily adapted to commercial wheelchairs, and at the same time, fulfill the desired system requirements.

Several types of input devices were used in this project to allow people with different disabilities to be able to drive the IW. The intention is to offer the patient the freedom to choose the device they find most comfortable and safe to drive the wheelchair. These devices range from traditional joysticks, accelerometers, to commands expressed by speech or facial expressions. Moreover, these multiple inputs for interaction with the IW can be integrated with a control system responsible for the decision of enabling or disabling any kind of input, in case of any observed conflict or dangerous situation. For example, in a very noisy environment, where the speech recognizer may not have an accurate behaviour, this type of input can be temporarily disabled. To compose the necessary set of hardware to provide the wheelchair's ability to avoid obstacles, follow walls, map the environment and see the holes and unevenness in the ground, a bar was designed, containing a set of eight ultrasonic sensors and twelve infra-red sensors. To refine the odometry, two encoders were also included, and coupled to the wheels (for distance, velocity and position measurements). It is important to remember that the low cost and maintenance of ergonomics and comfort of the original commercial wheelchair were regarded as some of the project requirements. Figure 1.4 shows the real prototype of the intelligent wheelchair.

Thus, the use of high-cost sensors such as laser range-finders or 3D video cameras was ruled out. To complement the hardware platform, some other devices were used, namely:

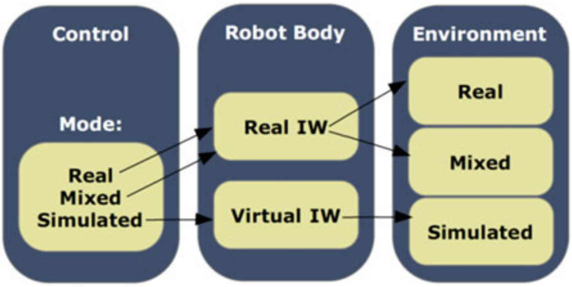


Fig. 1.3 Operation modes of IntellWheels Platform

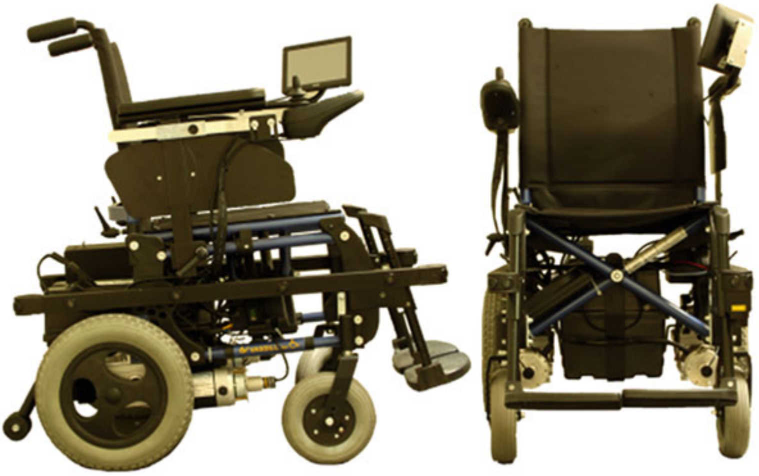


Fig. 1.4 Real prototype of the intelligent wheelchair

- Control and data acquisition board: This board is used to collect data from the sensors and send the motors the reference control for the power module. The board (ArduinoMega12 [38]) is connected to the computer platform via a USB connection.
- Power module: It converts the control signal in power for the motors and offers overcurrent protection (business module produced by PG Drives Technology, model VR2).
- Notebook: To run the platform software used, a laptop (HP Pavilion tx1270EP, AMD Turion 64 X2 TI60) is used. However, other computers with equivalent or superior performance might be used.

The system module, named IntellWheels Simulator [39] or more recent IntellSim, allows the creation of a virtual world where one can simulate the environment of an enclosure (e.g. a floor of a hospital), as well as wheelchairs and generic objects (tables, doors and other objects). The purpose of this simulator is essentially to support the testing of algorithms, analyse and test the modules of the platform and safely train

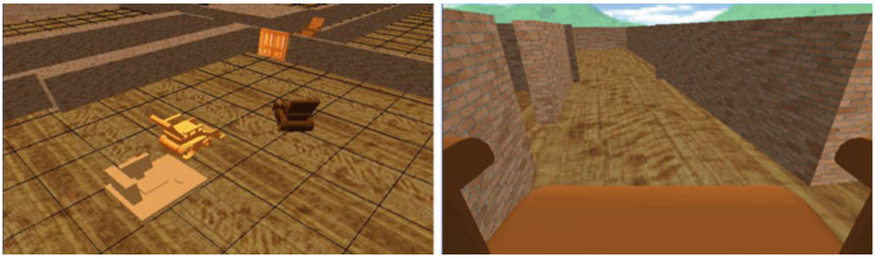


Fig. 1.5 Visualization modes of IntellWheels platform in the simulator

Table 1.1 High-level output actions implemented by the IWP control module

Action Name	Description
Go to X,Y	Go to a specific position of a known environment
Spin theta	Spin an user defined angle
Follow line	Move along the nearest landmarks
Follow right wall	Continuously follow the nearest obstacle on the right side
Follow left wall	Continuously follow the nearest obstacle on the left side
Go forward	Move forward at constant pre-defined speed
Go back	Backtrack at constant pre-defined speed
Right spin	Continuously spin to the right side, at constant pre-defined speed
Left spin	Continuously spin to the left side, at constant pre-defined speed
Right turn	Turn right at pre-defined constant linear/angular speeds
Left turn	Turn left at pre-defined constant linear/angular speeds

users of the IW in a simulated environment. Another purpose of the simulator is related to the testing scenarios involving a large number of intelligent wheelchairs, which would be impossible with real IW due to its cost and complexity. Since every small modification in the algorithms or hardware would imply, in real environments, a high increase of time and monetary costs, having a simulator is desirable and of great importance. Additionally, its use preserves the safety of the user, avoiding potentially dangerous situations during the test phases. The IntellWheels Simulator is an adaptation of the open-source simulator named Ciber-Rato [40]. Figure 1.5 shows examples of the IntellWheels simulator viewer, which provides different visualization modes.

The navigation module of the IWP includes a set of algorithms responsible for performing the treatment of the values of sensors on the wheelchair, in order to trace its location and map the environment [2]. The set of functions related to this module is implemented in a distributed manner among the agents Perception, Control and Intelligence. The IWP presents a multi-level control architecture, subdivided into three layers: strategic layer, tactical layer and basic control layer [37]. Some simple algorithms were implemented for the control of basic actions, such as: follow line, go to point (x, y) and turn to angle θ. These algorithms were based on the Cartesian position of the wheelchair. Other algorithms were based on the information of distance sensors (ultrasonic and infra-red sensors) were also implemented, such as following walls and avoiding obstacles. The implemented actions are depicted in Table 1.1.

1.4.2 IntellWheels Multimodal Interface

The problem of designing specific interfaces for individuals with physical disabilities presents many challenges to which many authors have tried to answer over time.

There are several publications in the literature of projects related to this issue, which tried different approaches to implement viable solutions to address this problem [14, 15, 23]. However, the vast majority of these projects present limited solutions concerning the accessibility and recognition methods offered to the user to drive a particular wheelchair. It is common to find solutions providing voice recognition only, while others focus merely on facial expressions recognition. Since the physical disability is very wide and specific to each individual, it becomes important to provide the greatest possible number of recognition methods to try to cover the largest possible number of individuals with different characteristics.

The proposed multimodal interface offers three basic methods of recognition: speech recognition, recognition of head movements and gestures, and the use of a generic gamepad/joystick. In addition, we propose an architecture that makes the interface extensible at this level, enabling the addition of new devices and recognition methods in a transparent way. It also presents a flexible paradigm that allows the user to define the sequences of inputs to assign to each action, allowing for an easy and optimized configuration for each user.

Following, the proposed IMI features and global architecture are presented and explained.

1.4.2.1 Input Sequence

The concept behind an input sequence is very important to understand the adopted interaction style. Some multimodal systems combine different input modalities in order to issue multimodal commands. Although the merging of different inputs can provide a very attractive way of human machine interaction, it is not ideal for people with physical disabilities.

The objective of the IntellWheels multimodal interface is to provide the user a large variety of different possible inputs that can be expressed by individuals with different physical capabilities. It is proposed a simple mechanism to allow the user to interact with the multimodal interface. It consists of intercepting the user inputs in a sequential fashion, without fusion. Since this is a system that aims to be used by disabled people, avoiding errors is of extreme importance. Consequently, having singular input tokens associated to the request of an output action is not advisable. For example, if a user associates a voice input which consists of pronouncing the word “Go” to the output action that makes the wheelchair start moving forward, this may origin mishaps in the way that a false recognition by the speech recognition module would originate a not desired event. Instead, it is proposed the possibility of creating input sequences, composed by simple input tokens that may be easily performed by the user.

Figure 1.6 contains a graphical illustration of an input sequence. It would consist of pronouncing the expression “This is an example”, followed by pressing the button 5 of the gamepad, and finally blinking the left eye, in order for the IW turning right.

1.4.2.2 User and Multimodal Interactions

This section presents the main interactions between the user and the IntellWheels Multimodal Interface. They were grouped in four different main modules (Figs. 1.7 and 1.8):

- Navigation Mode—This module uses previously saved associations and user inputs to derive the output actions during IW navigation. This represents the most important feature of the multimodal interface, which is to request the execution of a certain action, via input sequence, using a previously created association.
- Sequence Manager—This module is related to the management of the saved associations between input sequences and output actions. The user can create a new association by selecting the inputs and the desired sequence (which can be done by actually executing the input sequence) and map it to the desired IW action. It is also possible to consult associations created in the past, and to change or delete them.
- Device Manager—This module includes the interaction that allows the user to consult a small viewer which shows information regarding the state of basic input devices such as information about the wiimote motion sensors, battery level and active gamepad buttons. This viewer is intended to test the correct functioning of the input devices. The user can also adjust some minor configurations for these devices, to attain an optimized experience while using the multimodal interface. Configurations may consist of defining a minimum speech recognition trust-level and maximum speed for the gamepad joysticks.
- User Profiling—This module presents the interactions related to the creation or update of a user profile. During this process, the user can separately train the use of the basic input modalities offered by the IntellWheels multimodal interface.

One of the main goals of this system architecture proposal is to provide the user with full control of the interface.

1.4.2.3 Global Architecture

Considering all the proposed features, the overall architecture of the system is presented in Fig. 1.9. This subsection gives an overview of all the system components roles.

The IMI is composed by five main components:

- Generic Gamepad—Represents the thread responsible for implementing a driver to access a generic gamepad. It handles the device connection, and the state of each

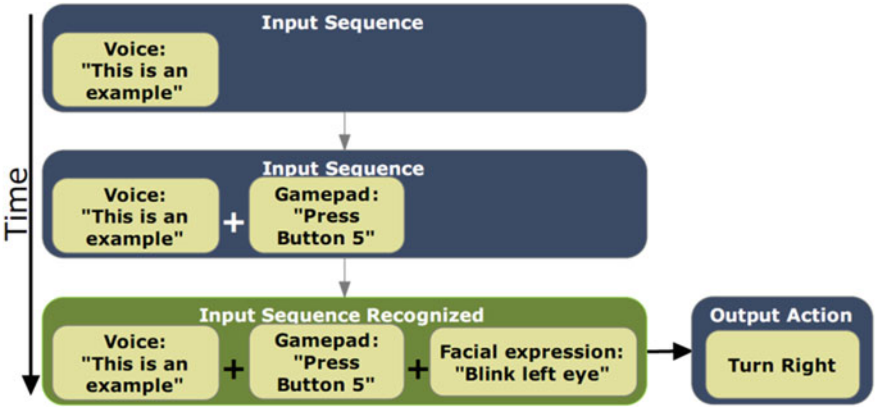
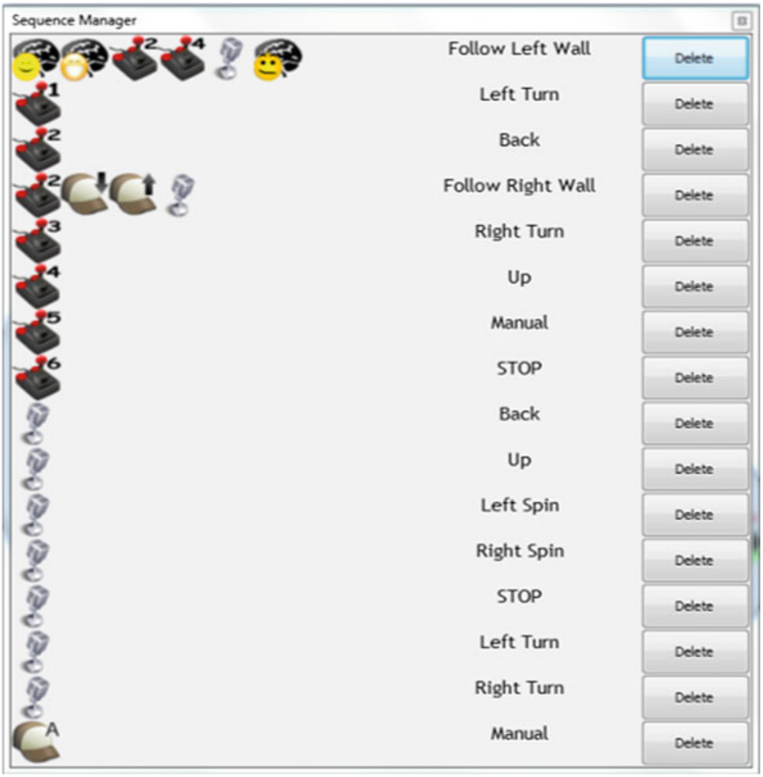

Time
Input Sequence
Voice: "This is an example"
Input Sequence
Voice: "This is an example"
+
Gamepad: "Press Button 5"
Input Sequence Recognized
Voice: "This is an example"
+
Gamepad: "Press Button 5"
+
Facial expression: "Blink left eye"
Output Action
Turn Right

Fig. 1.6 Concept of input sequence



2 4	Follow Left Wall	Delete
1	Left Turn	Delete
2	Back	Delete
2	Follow Right Wall	Delete
3	Right Turn	Delete
4	Up	Delete
5	Manual	Delete
6	STOP	Delete
	Back	Delete
	Up	Delete
	Left Spin	Delete
	Right Spin	Delete
	STOP	Delete
	Left Turn	Delete
	Right Turn	Delete
A	Manual	Delete

Fig. 1.7 Sequence manager with a command language

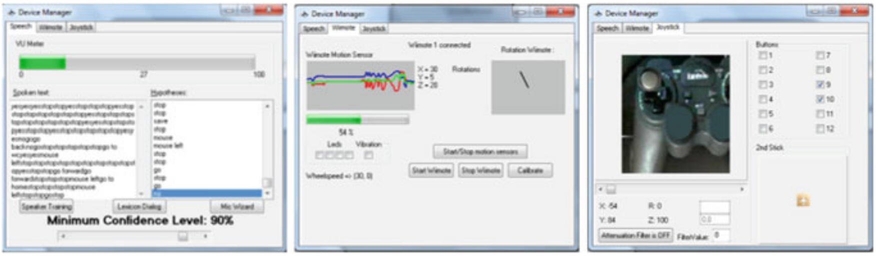


Fig. 1.8 Device manager with information about the basic input devices

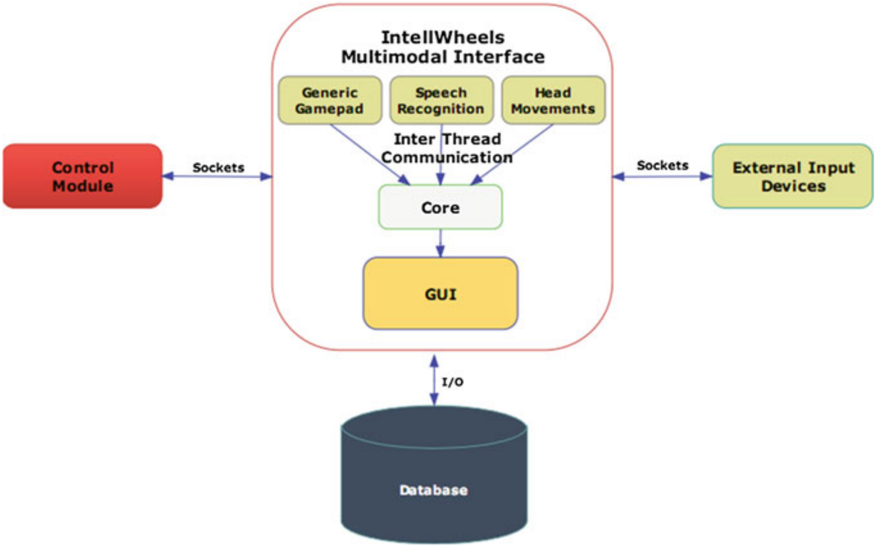


Fig. 1.9 IMI Database

button: pressed or released. It also reads the position of the two analog switches that compose a generic gamepad. Its task consists also of implementing a set of events used to notify the core component every time there is a change on the gamepad status, or a request by the component regarding analog positions is made.

- Speech Recognition—This component is responsible for accessing a microphone and handles the voice commands given by the user. It notifies the core component every time a known voice command is recognized.
- Head Movements—This is the component responsible for accessing the device that tracks the users' head movements. One of its functions is detecting specific head movements and notify the core component on every detection. Another function is to continuously send, upon request, the users' head relative position to the core component.

- Core—This is the main component of the IMI. One of its functions is to access the database to store and load all the necessary data. It is also responsible for receiving inputs of all the input devices available in the system (internal or external), and implement the algorithms responsible for matching input sequences with desired output actions. Furthermore, it manages the GUI state by sending it information regarding systems events. Finally, it implements the User Profile tracker.
- GUI—The GUI role is to make a graphical presentation of all the information that might be useful for the user.

The external components are:

- Control Module—This component interacts with the IMI by initially sending information concerning the actions it can perform. It is responsible for receiving IMI requests and generates the desired system output.
- External Input Devices—This component is composed by a variable number of input devices that might connect to the IMI to extend its multimodality. Each external input device should have a similar behaviour compared to the internal input devices. The only difference is that they are not embedded in the IMI.
- Database—The database task is to store all the information regarding available actions, existing associations and inputs. It also stores the user profiles and the description of the test sets used by the User Profile tracker.

1.4.3 System Implementation

A prototype of the IntellWheels Multimodal Interface was implemented [41]. This section presents the most relevant implementation details. It includes the description of the basic input modalities that were implemented, the data representation structures that were used, and the interaction between all the system's components. Furthermore, the used approach to analyse and validate input sequences is also documented. Finally, the methodologies used to track the user's profile are also fully explained. The development took advantage of Embarcadero Delphi [42]. It is an integrated software development environment that allows visual, event-oriented programming through Pascal programming language.

1.4.3.1 Basic Input Modalities

This section presents the three basic input modalities that were embedded to the IMI: Generic Gamepad, Speech Recognition and Head Movements (Fig. 1.10). Besides the use of a generic gamepad, a head set with an incorporated microphone was used for the speech recognition module.

To implement the component responsible for detecting the head movements, it was used a Nintendo Wii Remote. Since the main objective of this experiment was to try to prove the concept proposed, low cost devices that could be easily programmable

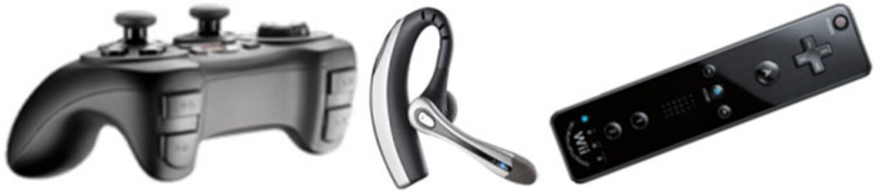


Fig. 1.10 Generic gamepad, headset with microphone and wiimote

were used. The following subsections present a more detailed description concerning the implementation of the IMI embedded input recognition modalities.

Generic Gamepad

USB Generic Gamepads are a bit more sophisticated compared to regular joysticks. This type of gamepad is widely used in video games once it offers a big number of buttons and analog axis that can be easily programmed to provide an attractive way of navigation. Figure 1.10 shows an example of a generic gamepad that can be used with the IWI. These gamepads provide ten programmable buttons and two analogical joysticks. Each analogical switch contains two axis.

Speech Recognition

This feature uses a headset microphone to capture the sound of the users' speech, which is then interpreted by a speech recognition engine. The speech recognition module used in the IMI takes advantage of the Microsoft Speech Recognition Engine. This engine provides an Application Programming Interface (API), named Speech Application Programming Interface [43]. SAPI allows the use of speech recognition and speech synthesis within Windows applications.

SAPI was chosen since it is a freely redistributable component which can be shipped with any Windows application that wishes to use speech technology. Although many versions of the available speech recognition and speech synthesis engines are also freely redistributable, SAPI offers a standard set of interfaces accessible from a variety of programming languages.

Therefore, it was possible to implement this feature without the need of installing and accessing external software. Besides, it proved to be a very efficient choice since the accuracy of recognition was extremely satisfactory, even without the user having to previously train the voice before the first use. One interesting feature of the Microsoft Windows SAPI is the possibility of designing grammar rules that can be used for specific contexts. This feature makes it possible to limit the set of recognition hypothesis to a custom set of voice commands, substantially increasing the recognition match accuracy. To implement the desired functionality for the IMI, a context-free grammar (CFG) was used to store the possible voice commands to be transformed in input tokens. This grammar follows the SAPI grammar format, and is stored inside a XML grammar file.

Head Movements

The Wii Remote, sometimes unofficially nicknamed "Wiimote" it is the primary controller for Nintendo's Wii console. A main feature of the Wii Remote is its motion sensing capability, which allows the user to interact with a computer via gesture recognition through the use of a three-axis accelerometer [44]. In our approach, besides using the Wii Remote accelerometer values to directly control the wheelchair's trajectory, we propose methods to detect four distinct head movements: leaning the head forward; the head back; the head right and the head left.

The approach consists of defining an amplitude value that triggers a small timer. The timer is started if the user leans the head an amplitude higher than the one defined. After the timer event is triggered, the head movement module checks if the head position (accelerometer value) is back to the original position. Both the amplitude and the interval between the triggered events are configurable and can be adjusted for each user. These head movements can be used to compose input sequences, having a similar utility of a voice command or a button on the gamepad.

1.4.3.2 External Input Modalities

To accept inputs from external devices, a very simple protocol was implemented. The IMI acts as a server for receiving external connections. An interaction between an external device has three distinct phases. On the connection phase, the external application should identify itself by sending a command with its name. This information, via navigation assistant, is used to notify the user that a new input device is available. After this phase, the IMI becomes ready to receive inputs from the new input device. However, these inputs are only useful for defining input sequences and not to act as a parameter source, as the gamepad joysticks and wiimote and accelerometer. On disconnect, the external application should notify the IMI, sending a disconnect command.

1.4.3.3 Data Representation Structures

The structures used to represent the output actions, input sequences and associations are shown below.

Actions

The output actions provided by the IWP control module differ slightly in type and nature and can be divided into four types:

- High-level—the trajectory of the wheelchair's movement is calculated only by the control module of the platform (e.g., go forward, go back, turn right, follow right wall), using pre-defined values for the speed of the wheels.
- Medium-level—identical to the high level type, whereas the user is responsible for setting the desired speed/angle for the movement.

Table 1.2 Action type parameters

Action type	Set parameters	Parameter type
High-level	0	n/a
Mid-level	1	Linear speed or rotation angle
Manual-mode	2	Linear speed and angular speed
Stop	0	n/a

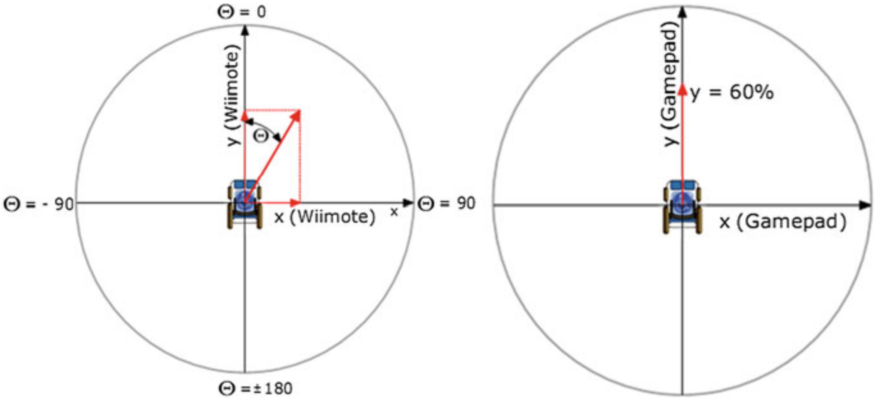


Fig. 1.11 Mid-level rotation angle and linear speed parameters

- Manual-mode—the user is responsible for controlling the wheelchair's trajectory. If the shared control option is activated, the control module of the platform may help only by detecting obstacles and avoiding collisions.
- Stop—the wheelchair is stopped and enters a stand-by mode until further request.

To design a generic structure to represent them on the multimodal interface side, the following descriptors were used: the name of the action (name); the action type (type); the kind of parameter to be passed (parameters) and the hint text for the TTS navigation assistant (hint).

Table 1.2 shows the relation between each action type and the number of parameters sent, which will generate a certain output.

The possible sources for extracting these parameters are the gamepad's analog switches and the wiimote's accelerometers, previously shown.

Depending on the parameter, one may need one or two axis to extract the desired output. For sending linear and angular speeds, the value of each axis is directly sent to the control module. A rotation angle is calculated based on the direction of the resulting vector. To represent the number of axis needed for each parameter type, the following descriptors were used: analog (1 axis) or vector (2 axis).

In the mid-level type, the parameter to be passed might be related to the wheelchair's linear speed (e.g. go forward at speed 30 %) or to the rotation angle (e.g. spin 90°), depending on the action nature. Since the linear speed requires only one axis, and the rotation angle requires two axis, the parameter descriptor for a mid-level may vary (analog or vector). Figure 1.11 shows an example of how a rotation angle is calculated according to the two axis that are used. In the example shown, the angle would be calculated using the Wiimote, so the user would lean the head to express the desired rotation angle. For a rotation angle, the {vector} descriptor is used.

To extract a linear speed, the analog descriptor is used, since it is only necessary to read the value of one axis. For example, to configure a parameter for an action named "Go Forward at Desired Speed", the user could chose the gamepad's left analog switch 'y' axis to express the desired speed. When requesting this action, the user should press the analog switch to a certain position, in order to set the desired speed. In the manual-mode type the vector descriptor is used, since two axis are needed to directly control the wheelchair. Therefore, the linear and angular speeds are sent, since the wheelchair used in this experimental validation automatically calculates the speed of both wheels according to these parameters [39]. The maximum value for both linear and angular speeds accepted by this wheelchair model is 100. To obtain a correct movement while controlling the wheelchair in manual mode, using the gamepad or the wiimote, it was necessary to parametrize the maximum values of each of both devices axis to 100. Additionally, after some initial experiments, we noticed a very abrupt variation on the wheelchair's movement every time we tried to perform curvilinear trajectories. After some tests, we discovered that in order to obtain a smooth variation of the wheelchair's direction while curving, it is necessary to truncate the linear and angular speeds to a limited space. The limited space is the one contained on the circumference of radius 100.

Input Sequences

An input sequence is composed by one more input tokens. An input token is formed by two parts: device descriptor and input descriptor, assuming the following format: device_id.input_id. This way, an input sequence must be formed by one or more input tokens. For example the input sequence <joystick.2 wiimote.right> means "Pressing the button 2 of the gamepad, followed by leaning the head to the right".

Associations

To represent an association between an action and an input sequence, it was used the structure: inputs—the sequence of input tokens that will trigger the action's request; action name—the output action to request and parameters—the source of the parameters.

1.4.4 Actions

Since the IMI offers a different set of features, it was important to think of a way to make them easily accessible by people with physical disabilities. Depending on the level of physical disability, controlling a mouse can represent an impossible task for

Table 1.3 List of IMI interface actions

Interface action	Description
Mouse left	Click left mouse button
Double mouse left	Double click left mouse button
Mouse right	Click right mouse button
View/hide output actions	Open/close the output actions viewer
View/hide associations	Open/close the list of saved associations
View/hide device manager	Open/close the input devices' tabs panel
New sequence	Start the sequence creator mode
Save sequence	Associate sequence to action and save
Calibrate wiimote	Reset the wiimote controller accelerometer values to 0
Enable/disable Speech	Turn the speech recognition module on or off
Enable/disable Assistant	Turn the TTS navigation assistant feature on or off
Mouse POV	Enable the control of the mouse using the gamepad's POV
Mouse wiimote	Enable the control of the mouse using head movements
Minimiza/maximize	Show or hide the applications
Profiler	Start the user profile tracker module

many individuals. For example, to consult the list of available output actions or saved associations, it is necessary to perform two clicks on the interface. Therefore, it was decided to adopt the same style of interaction, not only to request output actions of a certain control module, but also for controlling actions related to the IMI. Then it was possible to associate an input sequence to a different type of action: interface action. The set of implemented interface actions, embedded in the IMI, can be consulted in Table 1.3.

In spite of being a different action type, the structure used to save it is the one used for the output actions. Only the type field changes.

1.4.5 Multimodal Interaction Loop

1.4.5.1 Control Module Versus IMI

To allow the interaction between the IMI and the IWP control module, the latter had to be adapted in order to implement the action structure. After the connection process, the control module must send the list of the available actions. Additionally, it should also periodically send the availability of each action. This feature was implemented because in order to perform some actions, the control module may depend on external software or hardware, which might fail. Figure 1.12 shows the global flow for both the IMI and the control module.

1.4.5.2 Sequence Analysis

The input sequence analysis represents the most important feature of the IMI. Each time an input token is perceived by an input device recognizer, a notification is sent to the core component, which is responsible for checking if the current input sequence matches any of the existing associations. The sequence analyser implementation takes advantage of the method used to represent an input sequence. As it was explained before, an input token consists of the pair formed by the device descriptor and the input descriptor.

Our approach consists of keeping sorted all the input sequences associated to output actions. In this way, it is possible to apply a binary search every time a new input token is received. The Binary Search algorithm [45] has a fast performance and it is ideal to this situation, since an input sequence is sequentially growing. Therefore, it is necessary to constantly compare the current input sequence with fragments of the stored associations. If at any time, the user's input sequence does not match any fragment of the same size, the sequence is immediately discarded, and the user is notified. Inversely, if the current input sequence is a fragment of one or more associations, a timer is activated, in order to wait for further inputs that may transform the current input sequence into a valid match. When a match happens, it is necessary to verify if there is any other association composed by the current input sequence and one or more input tokens. In this situation, a different timer is activated. Meanwhile, the IMI waits for further inputs. If no input token has been given by the user when the timer event is triggered, the associated output action is requested to the control module. Otherwise, the timer is turned off and the process takes its normal flow.

Finally, if there is a perfect match which means that the current input sequence is mapped to an output action, and there is not any other input sequence partially equal, the associated output action is immediately requested.

1.4.5.3 Sequence Creator

In order to enable the creation of a new input sequence, and consequent association to an output action, the interface has switch to a different flow mode. When this process is started, the navigation mode is temporarily halted, by turning off the sequence analysis. Instead, the input sequence is sequentially updated and kept until the user decides to associate it with a desired output action, or, alternatively, cancel the creation of a new association. At any stage of this process, the user may exit the sequence manager option and return to the navigation mode. Figure 1.13 shows the IMI flow for the association of a new input sequence to a desired output action.

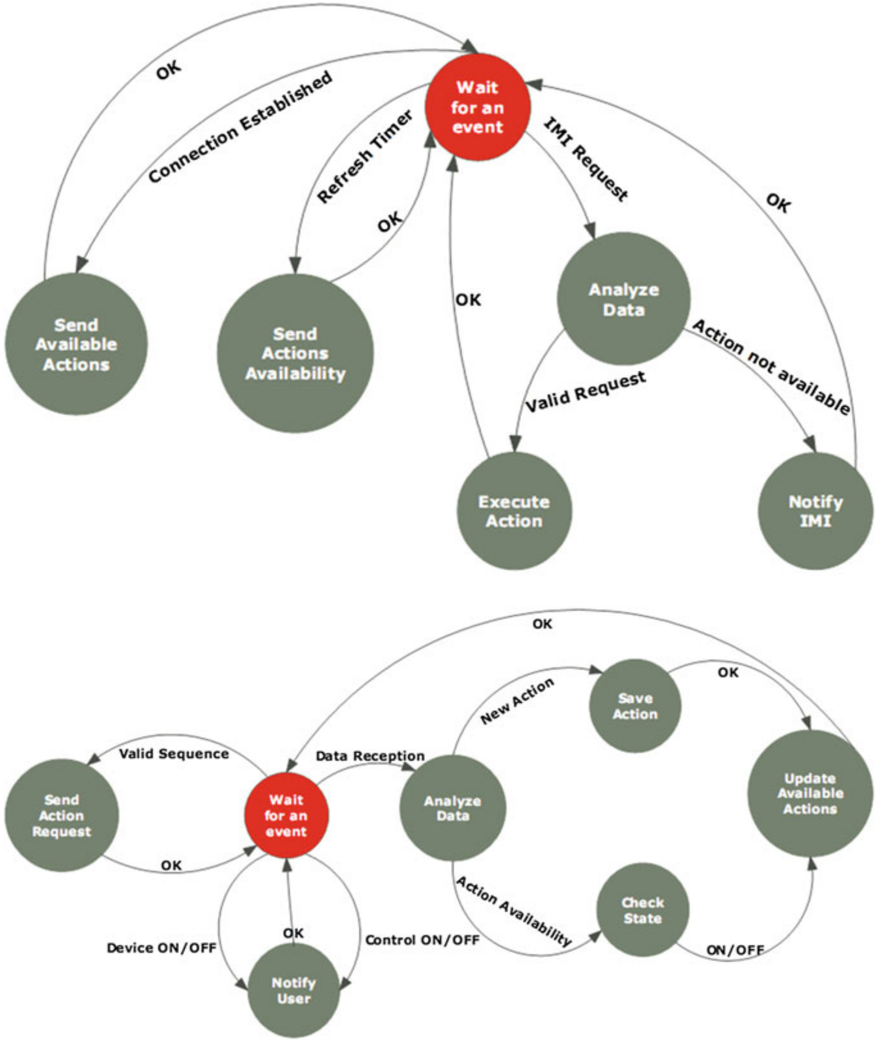


Fig. 1.12 Control module flow and IMI global flow

1.4.6 Graphical User Interface

A simple graphical user interface was created in order to present all the necessary information to the user in an attractively way. Figure 1.14 shows the developed graphical user interface for the IMI.

When the application is launched, the navigation assistant, presented at the top of the GUI, starts by greeting the user. To implement the TTS feature, we used Microsoft Anna Text to Speech Voice. The navigation assistant is responsible of interacting with

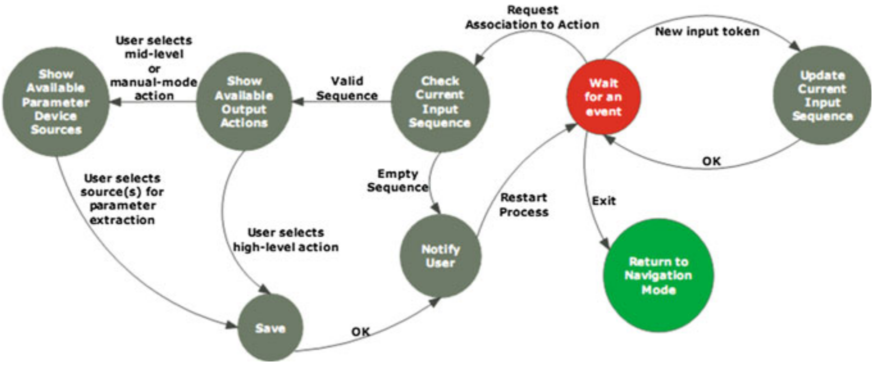

Show Available Parameter Device Sources
User selects mid-level or manual-mode action
Show Available Output Actions
Valid Sequence
Check Current Input Sequence
Request Association to Action
Wait for an event
New input token
Update Current Input Sequence
OK
User selects source(s) for parameter extraction
User selects high-level action
Empty Sequence
Restart Process
Exit
Notify User
Return to Navigation Mode
Save
OK

Fig. 1.13 Sequence creator flow

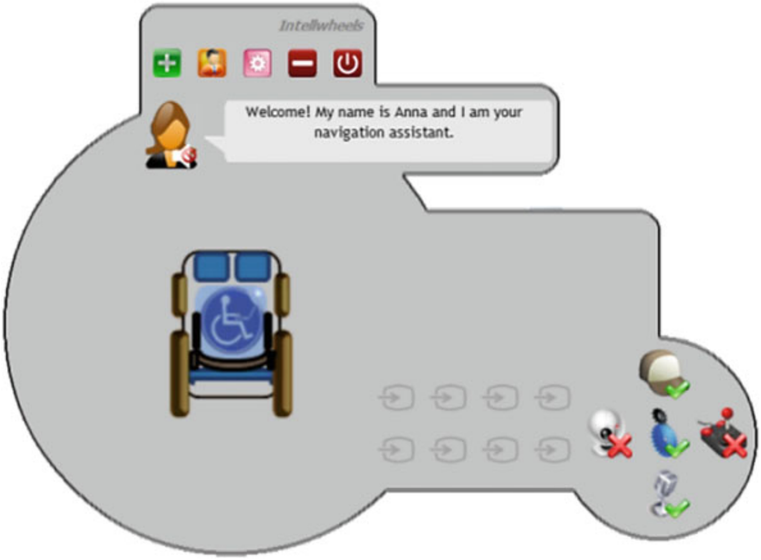

Intellwheels
Welcome! My name is Anna and I am your navigation assistant.

Fig. 1.14 IMI graphical user interface

the user by synthesizing the text presented in the talking balloon that stands on its right side. The text on the balloon changes every time a new event is triggered, in order to keep the user informed of the system's changes. The TTS feature can be switched on or off, either by clicking on the image chosen to represent the navigation assistant, or by associating an input sequence to the interface action responsible for the same setting.

On the top of the interface there is a set of five buttons, which implement the features related with the sequence manager, user profiler and device manager. To keep the user informed of the availability of each one of the input devices, a set of five icons was placed at the right side of the interface. Every time an input device connects or disconnects from the multimodal interface, its icon changes. Since part of the future work for the IMI is implementing a facial expression recognizer, an icon to represent a webcam was also prepared. Additionally, an icon to represent the control module is also present. In the example presented in Fig. 1.14, the wiimote controller, microphone and control module were connected to the multimodal interface. Inversely, the gamepad and the external input application concerning the facial expressions recognizer were not available. Other information provided by the IMI relates to the wheelchair's trajectory. This is done by presenting a set of directional arrows that surround the wheelchair image placed on the left centre of the interface. Finally, the small set of eight icons at the centre of the interface is reserved for the information regarding the input sequences. Every time the user gives a new input token, the respective image for that input token is shown.

1.5 Experiments and Results

In this section we present an evaluation of the results achieved by the system specification and by the implemented prototype. To test the integration of the multimodal interface with the previously developed IntellWheels Platform, two different test scenarios were prepared. The purpose of the first experiment was the evaluation of the multimodal interface performance using the IWP simulator. Another test scenario was prepared, this time using the IntellWheels wheelchair prototype on a real environment.

The first experiment involved 43 individuals. A simulated scenario of part of an institution was recreated, and a specific route was traced (using the IWP simulator). Figure 1.15 illustrates the 2D representation of the route.

The results evaluation was based on empirical research and a quasi-experimental design was followed. The main goal of this evaluation was to attain the performance and efficiency of the developed basic input modalities, and the overall behaviour of the IMI. The intention was to list the existence of possible bugs, faults and inconsistencies of the IMI, based on the individuals' feedback.

The methodology applied was the gathering of opinions using a questionnaire that incorporates the System Usability Scale (SUS) [46] which is a simple ten-item Likert scale giving a global view of individual assessments of usability [46]. The questions were organized into four main groups:

- User identification: Several questions about the gender, weight, height and experience using video games;
- Usability and Safety: Questions related with safety and control;
- Controls of the IW: Questions regarding the level of satisfaction with the different existing commands of the IW, such as joystick in manual and high level mode, voice commands and head movements, as well as the integration of all kind of modalities;

- Multimodal Interface: Questions related with the level of satisfaction about the information provided by the multimodal interface.

Several different ways of driving the IW to take the route were defined: using the gamepad joystick in manual mode; using the gamepad buttons in high-level mode; drive with head movements (Wii controller); drive the IW with voice commands; drive the IW having the freedom to choose any type of input (gamepad, voice, head movements). Since none of the users had any type of previous contact with the IntellWheels project, the first step was to provide an explanation of the main characteristics of the IMI, the type of output actions provided by the IWP control module, and the global goals of the experiment.

The second experiment included the participation of 12 individuals in a real institution. The goal was to evaluate the performance of the IMI using a real wheelchair.

Table 1.4 contains the defined input sequences, where GB means Gamepad Button, WB means Wiimote Button, GJ means Gamepad joystick and WM stands for Wiimote head movements.

The results about the users' opinions concerning the two experiments were achieved through the analysis of the answers to the questionnaire. The performance achieved using the different modes for driving the IW were assessed by the time, number of collisions and total average deviation error from the given "ideal" trajectory relatively to the desired trajectory. The time for both experiments was measured by a chronometer. The number of collisions for the first experiment was collected through

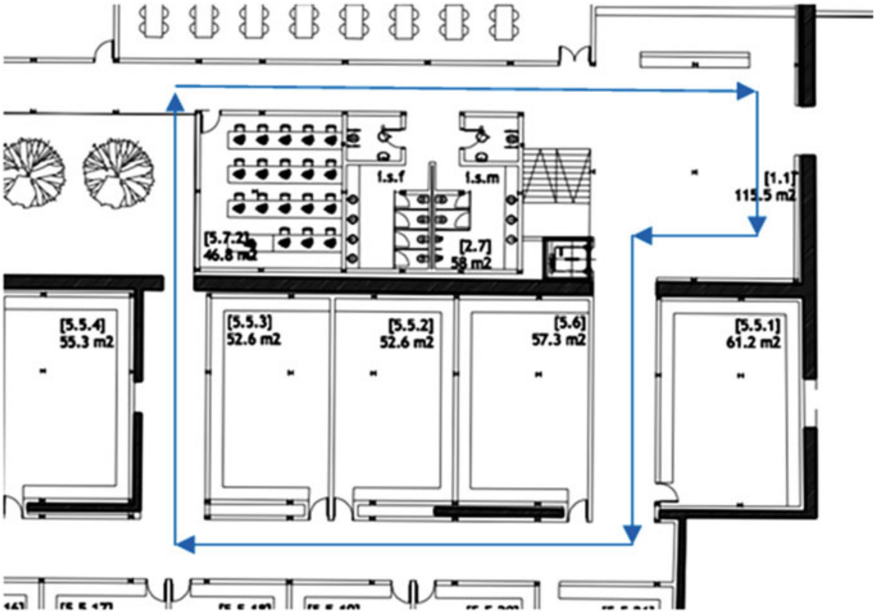


Fig. 1.15 Map of the route for the simulated experiment

the simulator logs, whereas for the second experiment this number was collected by an observer. In the first experiment, the measurement of the distance between the real and ideal trajectories was calculated applying the Euclidean distance of a point to a line segment. In the second experiment, the distance between the IW from the ideal trajectory was achieved using Ubisense [47] technology, which provides real-time location tracking. Table 1.5 shows the answers and the performance results obtained in the two experiments.

In terms of characterization of the independent samples, the participants have slightly different results in the age, height and weight. Most of the answers concerning previous experience with video games, in both groups, focus on Never or Rarely, showing the individuals' lack of experience with this kind of devices. The usability mean of the SUS score is high in the group of the real environment. This result is justified by the realism that may be lost in the simulated environment.

Another relevant aspect is related with the difficulty and attention needed to drive the IW in tight places. In both experiments, most of the participants affirmed attention is needed in order to drive the IW, mainly in tight places. However, participants also stated they felt they had good control of the IW both in the real and simulated environments. The level of satisfaction with the controls has a median of Satisfied (level 4 in a Likert scale of 5), except for the voice commands. In fact, since these kinds of commands had a latency period, the individuals needed more time to adapt to it.

The information provided by the multimodal interface was mostly classified has Indifferent. The performance of time, number of collisions and error from the "ideal" trajectory achieved using the several commands are worst in the case of voice commands by the same reason of the latency period previously referred. However it is necessary to show if there are statistical evidences to affirm these kinds of differences. For that, a significance level of 0.05 was applied to the statistical test. The tests used were independent samples t test and Mann-Whitney. In Table 1.6 there are the p values and the tests' power for a medium effect size of 0.5.

Table 1.4 List sequences used for the experiment using the simulated IW

Output action	Button input sequence	Voice commands
Go forward	GB1	"Go forward"
Go back	GB3	"Go back"
Right turn	GB2	"Turn right"
Left Turn	GB4	"Turn left"
Right spin	GB6	"Right spin"
Left spin	GB5	"Left spin"
Stop	GB7 or GB8	"Stop"
Manual mode using GJ	GB9	"Manual mode joystick"
Manual mode using WM	WB 'A'	"Manual mode wiimote"

GB—Gamepad button; WB—Wiimote button; GJ—Gamepad joystick; WM—Wiimote head movements

Table 1.5 Results of the experiments in simulated and real environments

	Simulated environment (n = 43)			Real environment (n = 12)		
	Mean	Median	Std.	Mean	Median	Std.
User identification						
Age	21.12	20	2.29	24.58	23.5	5.99
Weight (kg)	60.44	58	9.90	69.25	65.50	11.10
Height (cm)	164.84	165	6.283	173.75	172.5	11.07
Freq play per week		Rarely			Stimes	
Use of Wii controls		Rarely			Stimes	
Use joysticks		Never			Stimes	
Usability and safety						
Score SUS	62.79	65	15.65	76.04	78.75	12.31
Safety managing IW		Agree			Agree	
Control of the IW		Ind			Ind	
Easy to drive the IW in tight places		Dis			Dis	
The IW do not need to much attention		Dis			SDis	
Satisfaction level of controls of the IW						
Gamepad joystick manual mode		Satis			Satis	
Gamepad buttons high level mode		Satis			Satis	
Voice commands		Ind			Diss	
Head movements		Satis			Satis	
Using all commands		Satis				
Multimodal interface						
Information provide by Multimodal interface		Ind				
Performance Time (min)						
Gamepad Joystick manual mode	3.82	3.61	1.26	4.37	5.02	1.70
Gamepad buttons high level mode	3.85	3.25	1.82	5.88	4.23	3.96
Voice commands	6.75	6.32	2.13	6.32	6.38	1.80
Head movements	3.73	3.40	1.32	4.36	5.25	1.77
Using all commands	4.07	3.93	0.97			
N of collisions						
Gamepad Joystick manual mode	11.50	3.50	14.86	4.33	0	7.51
Gamepad buttons high level mode	6.07	3	8.03	2.33	1	3.22
Voice commands	29.43	28	20.98	21.67	19	20.13

(continued)

Table 1.5 (continued)

	Simulated environment (n = 43)			Real environment (n = 12)		
	Mean	Median	Std.	Mean	Median	Std.
Head movements	6.14	2.50	7.80	3.67	1	5.51
Using all commands	9.07	4	11.18			
Error of deviation from the trajectory asked						
Gamepad Joystick manual mode	0.26	0.26	0.10	0.20	0.23	0.05
Gamepad buttons high level mode	0.23	0.23	0.06	0.19	0.19	0.02
Voice commands	0.33	0.33	0.08	0.28	0.30	0.03
Head movements	0.30	0.28	0.11	0.27	0.27	0.07
Using all commands	0.32	0.26	0.20			

Legend: *Stimes* sometimes; *SDis* strongly disagree; *Dis* disagree; *Ind* indifferent; *SAgree* strongly agree; *VDiss* very dissatisfied; *Diss* dissatisfied; *Satis* satisfied; *VSatis* very satisfied

Table 1.6 Results of statistical tests in order to verify differences between the groups of simulated and real experiments

	T test (p value)	Mann-Whitney (p value)	Power
User identification			
Freq play per week		**0.014**	0.43
Use of Wii controls		0.076	0.43
Use joysticks		**0.002**	0.43
Usability and safety			
Score SUS*	**0.009**		0.32
Safety managing IW		0.420	0.43
Control of the IW		0.223	0.43
Easy to drive the IW in tight places		0.058	0.43
The IW do not need to much attention		0.359	0.43
Satisfaction level of Controls of the IW			
Gamepad joystick manual mode		0.066	0.43
Gamepad buttons high level mode		**0.000**	0.43
Voice commands		0.997	0.43
Head movements		0.170	0.43

Note * The *p* values were calculated with SPSS 18.0 and the Power achieved with G*Power 3.1.2. The Kolmogorov-Smirnov test was applied for the independent sample t test (p valuesimulated = 0.079 and p valuereal = 0.200) and the Levene test (p value = 0.009) in which it was assumed the equality of variances.

The results presented in Table 1.6 show statistical evidences to affirm there are differences between the simulated and real environments in terms of the SUS score. The experiment in using commands such joystick and level of satisfaction using gamepad buttons in high level also produce statistical evidences of being different. The usability result for the experiment using the real IW is higher since the perception of the environment while driving the IW is also higher when compared to the simulated environment.

1.6 Conclusions and Future Work

In this paper we presented the specification and development of a prototype of a multimodal interface, integrated in the IntellWheels platform. The final goal is to offer a multimodal style of driving the wheelchairs. To achieve the proposed goals, a study of a number of Intelligent Wheelchair projects was made to better understand the concepts behind this specific type of robot. Also, a revision of the existing input devices and recognition methods likely to be used by any kind of individual was also undertaken. The final step consisted of analyzing the concept of multimodal interface, its advantages, features and design principles that should be followed while implementing a system of this nature.

The next step was the specification of a new multimodal interface (IMI). The multimodal interface should offer three basic input modalities: generic gamepad, speech recognition and head movements and gestures (using a Nintendo Wii Remote). These modalities were chosen due to the very low cost of its implementation. Additionally, the proposed multimodal interface should act as a server for external input modalities, to make it possible to easily integrate and test recognition modules developed in the future. Other main feature consisted in offering a flexible configuration, by allowing user defined input sequences to be freely associated to available output actions.

The IWP modular architecture allowed the IMI to be designed without the need of embedding the algorithms responsible for directly controlling a wheelchair, since this part of the system is implemented on the control module side. Instead, generic structures for representing output actions were created, as well as a simple communication protocol. This approach enabled the interaction between the IMI and the IWP control module. This peculiarity of the system architecture allows the IMI to act as an input server for different control modules that can be applied to different contexts. A good example may consist of rapidly prototyping new video games without the need of addressing the implementation related with input devices and recognition methods. Other features consisted in designing a friendly user interface that could show useful information regarding the overall system state. Having the complete system specification, a prototype of the IntellWheels multimodal interface was developed. The only exception has to be with the development of the module responsible for tracking the user profile. Due to feature prioritization, the completeness of this module was left for further development.

Finally, the results achieved by the project were assessed. Two distinct experiments were performed to test the IMI integration with the IntellWheels Platform components. The individuals' feedback was important to detect bugs and inconsistencies, and to collect opinions regarding the system usability. Although the feedback of the individuals varied, the overall satisfaction of the attendants was very positive. In any case, one may say that in order to take full advantage of the developed multimodal interface a training session is advised. Moreover, some settings regarding each one of the input modalities should be configured taking into account the profile of each person.

Despite having a complete system specification and a functional prototype that implements most of the proposed features, further steps will be made to obtain a more mature system. An important feature under development is the user profile tracking module. The idea behind this module is to automatically adjust a set of interface settings by asking the user to perform series of configurable tests. These tests should extract information such as capacity of leaning the head, speaking ability, average time taken to perform different input sequences. Other feature will consist of developing a robust facial expression recognizer to be embedded to the IntellWheels multimodal interface. This feature will consolidate the range of available inputs that can be used by people with disabilities. Another aspect to improve is the viewer of the simulator in order to achieve higher realism in the experiments. Moreover, we intend to explore the recent developments in brain-computer interfaces. Although this type of modality is still under strong research, it is expected that one day it might break any physical disability barrier.

Acknowledgments The authors would like to acknowledge to FCT—Portuguese Science and Technology Foundation for the INTELLWHEELS project funding (RIPD/ADA/109636/2009), for the Ph.D. Scholarship FCT/SFRH/BD/44541/2008, LIACC—Laboratório de Inteligência Artificial e de Ciência de Computadores da Universidade do Porto, DETI/UA—Dep. Electrónica, Telecomunicações e Informática, IEETA—Instituto de Engenharia Electrónica e Telemática de Aveiro and ESTSP/IPP—Escola Superior de Tecnologia da Saúde Porto—IPP.

References

1. Dayal, H.: Management of rehabilitation personnel within the context of the national rehabilitation policy (2009)
2. Braga, R., Petry, M., Moreira, A. P., Reis, L. P.: Concept and design of the intellWheels platform for developing intelligent wheelchairs. Inf. Control Autom. Robot. 191–203 (2009)
3. Sharma, R., Pavlovic, V. I., Huang, T. S.: Toward multimodal human computer interface. Proc. IEEE **86**(5), 853–869 (1998)
4. Wang, H., Wang, Y., Cao, A.: Video-based face recognition: a survey. World Acad. Sci. Eng. Technol. **35**(4), 293–302 (2009)
5. Tian, Y.L., Kanade, T., Cohn, J.F.: Handbook of face recognition. Stan, L., Anil, J., (eds.) pp. 247–274 (2005)
6. Sayette, M.A., Cohn, J.F., Wertz, J.M., Perrott, M.A., Parrott, D.J.: A psychometric evaluation of the facial action coding system for assessing spontaneous expression. J. Nonverbal Behav. **25**, 167–185 (2001)
7. Silva. L.: Head gestures recognition. In: Proceedings of the International Conference on Image Processing, vol. 3, pp. 266–269 (2002)
8. Gavankar, C., Warnekar, C.: Automated system for interpreting non-verbal communication in video conferencing. Int. J. Comput. Sci. Eng. (IJCSE) **2**, 22–27 (2010)

9. Gips, J., Di Mattia, P., Curran, F.X., Olivieri, P.: Using eagle eyes—an electrodes based device for controlling the computer with your eyes to help people with special needs. In: Proceedings of the 5th International Conference on Computers Helping People with Special Needs. Part I, pp. 77–83. Munich, Germany (1996)
10. Ashwash, I., Hu, W., Marcotte, G.: Eye gestures recognition: a mechanism for hands-free computer control. Available at: http://www.cs.princeton.edu/courses/archive/fall08/cos436/FinalReports/Eye_Gesture_Recognition.pdf. Accessed 2011
11. Tall, M., Alapetite, A., Agustin, J.S., Skovsgaard, H.H.T., Hansen, J.P., Hansen, D.W.: Møllenbach, E.: Gaze-controlled driving. In: Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems. CHI'09, pp. 4387–4392. USA, ACM, New York (2009)
12. Jia, P., Hu, H.H., Lu, T., Yuan, K.: Head gesture recognition for hands-free control of an intelligent wheelchair. Ind. Rob. Int. J. **34**(1), 60–68 (2007)
13. Nakanishi, S., Kuno, Y., Shimada, N., Shirai, Y.: Robotic wheelchair based on observations of both user and environment. In: Proceedings of the International Conference on Intelligent Robots and Systems, vol. 2, pp. 912–917 (1999)
14. Matsumoto, Y., Ino, T., Ogasawara, T.: Development of intelligent wheelchair system with face and gaze based interface. In: Proceedings of the 10th IEEE International Workshop on Robot and Human Communication, pp. 262–267 (2001)
15. Ju, S., Shin, Y., Kim, Y.: Intelligent wheelchair (iw) interface using face and mouth recognition. In: Proceedings of the 13th International Conference on Intelligent User Interfaces, IUI'09, pp. 307–314. ACM (2009)
16. Manasse, P.: Speech recognition. Available at: http://aac.unl.edu/Speech_Recognition.html. Accessed on January 2011
17. Youdin, M., Sell, G., Reich, T., Clagnaz, M., Louie, H., Kolwicz, R.: A voice controlled powered wheelchair and environmental control system for the severely disabled. Med Prog Technol **7**, 139–143 (1980)
18. Sasou, A., Kojima, H.: Noise robust speech recognition applied to voice-driven wheelchair. EURASIP J. Adv. Sig. Proces. **41**, 1–41 (2009)
19. Cyber Glove Systems. Cyber glove ii. Available at: http://www.cyberglovesystems.com/products/cyberglove-ii/overview. Accessed on Nov 2011
20. AnthroTronix.: The AcceleGlove—capturing hand gesture in virtual reality. Available at: http://www.gizmag.com/go/2134/. Accessed on Jan 2011
21. Microsoft.: Kinect for xbox 360. Available at: http://www.xbox.com/pt-PT/kinect. Accessed on May 2011
22. LURCH Project.: Lurch—the autonomous wheelchair. Available at: http://airwiki.ws.dei.polimi.it/index.php/LURCH_The_autonomous_wheelchair. Accessed on May 2011
23. Philips, J., Millan, J., Vanacker, G., Lew, E., Galán, F., Ferrez, P., Van Brussel, H., Nuttin, M.: Adaptive shared control of a brain-actuated simulated wheelchair. In: Proceedings of the 10th IEEE International Conference on Rehabilitation Robotics, pp. 408–414. Noordwijk, The Netherlands, 6 (2007)
24. Blatt, R., Ceriani, S., Seno, B.D., Fontana, G., Matteucci, M., Migliore, D.: Brain control of a smart wheelchair. In 10th International Conference on Intelligent Autonomous Systems (2008)
25. Rebsamen, B., Burdet, E., Guan, C., Zhang, H., Teo, C. L., Zeng, Q., Laugier, C., Ang Jr., M. H.: Controlling a wheelchair indoors using thought. IEEE Intel. Sys. **22**, 18–24 (2007)
26. Mahl, C., Hayrettin, G., Danny, P., Marieke, E., Lasse, S., Matthieu, D., Alexandra, A.: Bacteriahunt: evaluating multi-paradigm BCI interaction. J. Multimodal User Interfaces **4**(1), 11–25 (2010)
27. Shepherd: How Shepherd Center works. Discovery & Fit Health. Available at: http://health.howstuffworks.com/medicine/healthcare-providers/shepherd-center5.htm. Accessed 2010
28. Oviatt, S.: A Handbook of Human-Computer Interaction. In: Jacko J., Sears, A. (eds.) New Jersey (2002)
29. Dumas, B., Lalanne, D., Oviatt, S.: Human Machine Interaction, vol. 5440 (Chap. Multimodal interfaces: a survey of principles, models and frameworks, pp. 3–26. Springer, Berlin (2009)

30. Bolt, R. A.: “Put-that-there”: voice and gesture at the graphics interface. In: Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques, pp. 262–270. New York, USA (1980)
31. Johnston, M., Bangalore, S.: Matchkiosk: a multimodal interactive city guide. In: Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions. Barcelona, Spain, Article No. 33 (2004)
32. Johnston, M., Cohen, Philip R., McGee, D., Oviatt, S. L., Pittman, J. A., Smith, I.: Unification-based multimodal integration. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, ACL-35, pp. 281–288. Morristown, NJ, U.S.A. (1997)
33. Karpov, A., Ronzhin, A.: ICANDO: low cost multimodal interface for hand disabled people. J. Multimodal User Interfaces **1**(2), 21–29 (2007)
34. Norman, D.: The Psychology of Everyday Things. Basic Books, New York. Open Interface. Open interface platform. Available at: http://www.openinterface.org/platform/. Accessed on May 2011
35. Braga, R., Petry, M., Moreira, A., Reis, L. P.: A development platform for intelligent wheelchairs for disabled people. In: 5th International Conference on Informatics in Control, Automation and Robotics **1**, 115–121 (2008)
36. Reis, L.P., Braga, R., Sousa, M., Moreira, A.: Intellwheels MMI: a flexible interface for an intelligent wheelchair. In: Baltes, J., Lagoudakis, M.G., Naruse, T., ShiryGhidary, S. (eds) RoboCup, vol. 5949. Lecture Notes in Computer Science, pp. 296–307. Springer, Berlin (2009)
37. Braga, R., Petry, M., Moreira, A., Reis, L. P.: Platform for intelligent wheelchairs using multi-level control and probabilistic motion model. In: 8th Portuguese Conference on Automatic Control, pp. 833–838 (2008)
38. Banzi, M., Cuartielles, D., Igoe, T., Martino, G., Mellis D.: Arduino. Available at: http://arduino.cc/. Accessed 2011
39. Braga, R., Malheiro, P., Reis, L.P.: Development of a realistic simulator for robotic intelligent wheelchairs in a hospital environment. RoboCup2009: Robot Soccer World Cup XIII. vol. 5949. Lecture Notes in Computer Science, pp. 23–34. Springer, Berlin (2010)
40. Lau, N., Pereira, A., Melo, A., Neves, A., Figueiredo, J.: Ciber-rato: Umambientedesimulaçãoderobotsmóveiseautónomos. RevistadoDETUA **3**(7), 647–650 (2002)
41. Vasconcelos, S.: Multimodal Interface for an intelligent wheelchair. University of Porto, Faculty of Engineering. Retrieved April 21, 2012, from http://hdl.handle.net/10216/62054
42. Embarcadero. Available at: http://www.embarcadero.com/products/delphi. Accessed on May 2011
43. SAPI: http://www.microsoft.com/en-us/tellme/. Accessed on Jan 2012
44. Nintendo: Wii controllers. Available at: http://nintendo.com/wii/console/controllers. Consulted on May 2011. Accessed on May 2011
45. Black, P.: Binary search algorithm. Available at: http://xw2k.nist.gov/dads/HTML/binarySearch.html. Accessed on Jan 2011
46. Brooke, J.: SUS: aquick and dirty usability scale. In: Jordan, P.W., Weerdmeester, B., Thomas, A., Mclelland, I.L. (eds.) Usability Evaluation in Industry. Taylor and Francis, London (1996)
47. Ubisense: Precise real-time location. Available at: http://www.ubisense.net/en/products/precise-real-time-location.html. Accessed on May 2011

Part I
Intelligent Control Systems and Optimization

Chapter 2
Cognitive Modeling for Automating Learning in Visually-Guided Manipulative Tasks

Hendry Ferreira Chame and Philippe Martinet

Abstract Robot manipulators, as general-purpose machines, can be used to perform various tasks. Though, adaptations to specific scenarios require of some technical efforts. In particular, the descriptions of the task result in a robot program which must be modified whenever changes are introduced. Another source of variations are undesired changes due to the entropic properties of systems; in effect, robots must be re-calibrated with certain frequency to produce the desired results. To ensure adaptability, cognitive robotists aim to design systems capable of learning and decision making. Moreover, control techniques such as visual-servoing allow robust control under inaccuracies in the estimates of the system's parameters. This paper reports the design of a platform called CRR, which combines the computational cognition paradigm for decision making and learning, with the visual-servoing control technique for the automation of manipulative tasks.

Keywords Cognitive robotics · Computational cognition · Artificial intelligence · Visual servoing.

2.1 Introduction

In the last decades, with the venue of fields of study such as cybernetics, artificial intelligence, neuroscience and psychology; remarkable progresses have been made in the understanding of what is required to create artificial life evolving in real-world environments [1]. Still, one of the remaining challenges is to create new cognitive models that would replicate high-level capabilities; such as, perception and information processing, reasoning, planning, learning, and adaptation to new situations.

H.F. Chame (✉) · P. Martinet
Robotics Team, Institut de Recherche en Communications et Cybernétique
de Nantes (IRCCyN), Nantes, France
e-mail: hendry.ferreira-chame@irccyn.ec-nantes.fr

P. Martinet
e-mail: philippe.martinet@irccyn.ec-nantes.fr

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_2

The study of knowledge representation and thinking has led to the proposal of the concept of Cognitive Architecture (CA). A CA can be conceived as a broadly-scoped, domain-generic computational cognitive model, which captures essential structures and processes of the mind, to be used for a broad, multiple-level, multiple-domain analysis of cognition and behavior [2]. For cognitive science (i.e., in relation to understanding the human mind) it provides a concrete mechaniscist framework for more detailed modeling of cognitive phenomena; through specifying essential structures, divisions of modules, relations between modules, and so on [3].

A robot that employs a CA to select its next action, is derived from integrated models of the cognition of humans or animals. Its control system is designed using the architecture and is structurally coupled to its underlying mechanisms [4]. However, there are challenges associated with using these architectures in real environments; notably, for performing efficient low-level processing [5]. It can be hard, thus, to generate meaningful and trustful symbols from potentially noisy sensor measurements, or to exert control over actuators using the representation of knowledge employed by the CA.

In practice, implementations of cognitive models usually require wide expertise in many other fields (i.e., probabilistic navigation, planning, speech recognition; among others). Moreover, cognitive models are derived from a large spectrum of computational paradigms that are not necessarily compatible when considering software architecture requirements. Scientists in cognition research, and actually higher-level robotic applications, develop their programs, models and experiments in a language grounded in an ontology based on general principles [6]. Hence, they expect reasonable and scalable performance for general domains and problem spaces.

On the side of cognitive roboticists, it would not be reasonable to replace already existing robust mechanisms ensuring sensory-motor control by less efficient ones. Such is the case of the servo-vision control technique (or visual servoing) which uses computer vision data to control the motion of the robot's effector [7]. This approach has the advantage of allowing the control of the robot from the error directly measured on the effector's interaction with the environment; making it robust to inaccuracies in estimates of the system parameters [8].

This research seeks to contribute to the debate standing from the point of view of cognitive roboticists. It can be conceived as an effort to assess to what extent it is feasible to build cognitive systems making use of the benefits of a psychologically-oriented CA; without leaving behind efficient control strategies such as visual servoing. The aim is to verify the potential benefits of creating an interactive platform under these technologies; and to analyze the resulting flexibility in automating manipulative tasks.

2.2 Cognitive Architectures

According to [9], two key design properties that underlie the development of any CA are memory and learning. Various types of memory serve as a repository for background knowledge about the world, the current episode, the activity, and oneself;

while learning is the main process that shapes this knowledge. Based on these two features, different approaches can be gathered in three groups: symbolic, non-symbolic, and hybrid models.

A symbolic CA has the ability to input, output, store and alter symbolic entities; executing appropriate actions in order to reach goals [2]. The majority of these architectures employ a centralized control over the information flow from sensory inputs, through memory; to motor outputs. This approach stresses the working memory executive functions, with an access to semantic memory; where knowledge generally has a graph-based representation. Rule-based representations of perceptions/actions in the procedural memory, embody the logical reasoning of human experts.

Inspired by connectionist ideas, a sub-symbolic CA is composed by a network of processing nodes [3]. These nodes interact with each other in specific ways changing the internal state of the system. As a result, interesting emergent properties are revealed. There are two complementary approaches to memory organization, globalist and localist. In these architectures, the generalization of learned responses to novel stimuli is usually good, but learning new items may lead to problematic interference with existent knowledge [10].

A hybrid CA combines the relative strengths of the first two paradigms [9]. In this sense, symbolic systems are good approaches to process and executing high-level cognitive tasks; such as, planning and deliberative reasoning, resembling human expertise. But they are not the best approach to represent low-level information. Sub-symbolic systems are better suited for capturing the context-specificity and handling low-level information and uncertainties. Yet, their main shortcoming are difficulties for representing and handling higher-order cognitive tasks.

2.3 Visual Servoing

The task in visual servoing (VS) is to use visual features, extracted from an image, to control the pose of the robot's end-effector in relation to a target. The camera may be carried by the end-effector (a configuration known by eye-in-hand) or fixed in (eye-to-hand) [7]. The aim of all vision-based control schemes is to minimize an error $e(t)$, which is typically defined by

$$e(t) = s(m(t), a) - s^*. \tag{2.1}$$

The vector $m(t)$ is a set of image measurements used to compute a vector of k visual features $s(m(t), a)$, based on a set of parameters a representing potential additional knowledge about the system (i.e., the camera intrinsic parameters, or a 3-D model of the target). The vector s^* contains the desired values of the features.

Depending on the characteristics of the task, a fixed goal can be considered where changes in s depend only on the camera's motion. A more general situation can also be modeled, where the target is moving and the resulting image depends both on the camera's and the target's motion. In any case, VS schemes mainly differ in the way

s is designed. For image-based visual servo control (IBVS), s consists of a set of features that are immediately available in the image data. For position-based visual servo control (PBVS), s consists of a set of 3D parameters, which must be estimated from image measurements. Once s is selected, a velocity controller relating its time variation to the camera velocity is given by

$$\dot{s} = L_{\mathrm{s}} V_{\mathrm{c}}. \tag{2.2}$$

The spatial velocity of the camera is denoted by $V_{\mathrm{c}} = (v_{\mathrm{c}}, \omega_{\mathrm{c}})$, with v_{c} the instantaneous linear velocity of the origin of the camera frame and ω_{c} the instantaneous angular velocity of the camera frame. $L_{\mathrm{s}} \in R^{6\times k}$ is named the interaction matrix related to s. Using (2.1) and (2.2), the relation between the camera velocity and the time variation of e can be defined by

$$\dot{e} = L_{\mathrm{e}} V_{\mathrm{c}}. \tag{2.3}$$

Considering V_{c} as the input to the controller, if an exponential decoupled decrease of e is desired, from (2.3) the velocity of the camera can be expressed by

$$V_{\mathrm{c}} = -\lambda {L_{\mathrm{e}}}^{+} e, \tag{2.4}$$

where $L^{+} \in R^{6\times k}$ is chosen as the Moore-Penrose pseudoinverse of L_{e}, that is ${L_{\mathrm{e}}}^{+} = ({L_{\mathrm{e}}}^{\mathrm{t}} L_{\mathrm{e}})^{-1} {L_{\mathrm{e}}}^{\mathrm{t}}$ when L_{e} is of full rank 6. In case $k = 6$ and $\det(L_{\mathrm{e}}) \neq 0$, it is possible to invert L_{e} such as $V_{\mathrm{c}} = -\lambda {L_{\mathrm{e}}}^{-1} e$.

Following (2.4), the six components of V_{c} are given as input to the controller. The control scheme may be expressed in the joint space by

$$\dot{q} = -\lambda({J_{\mathrm{e}}}^{+} e + P_{\mathrm{e}} e_{\mathrm{s}}) - {J_{\mathrm{e}}}^{+} \frac{\partial e}{\partial t}, \tag{2.5}$$

where J_{e} is the feature Jacobian matrix associated with the primary task e, $P_{\mathrm{e}} = (I_6 - \widehat{J_{\mathrm{e}}}^{+} \widehat{J_{\mathrm{e}}})$ is the gradient projection on the null space of the primary task to accomplish a secondary task e_{s}, and $\frac{\widehat{\partial e}}{\partial t}$ models the motion of the target. An example of VS is presented in Fig. 2.1.

2.4 The CRR Proposal

The Cognitive Reaching Robot (CRR) is a system designed to perform interactive manipulative tasks. When compared to non-cognitive approaches, CRR has the advantage of being adaptive to variations of the task; since the reinforcement learning (RL) mechanism reduces the need for explicitly reprogramming the behavior of the robot. Furthermore, CRR is robust to changes in the robotic system due to wear. It is

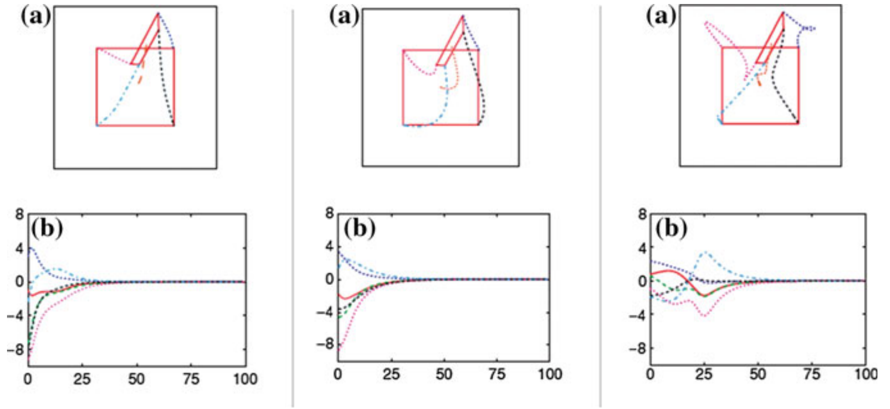


Fig. 2.1 Comparison between three IBVS (Image-base visual servoing) control schemes [8]. **a** Initial and final position of the target on the camera image, and the trajectory followed by each point and the *center* of the virtual polygon. **b** Evolution of V_c. From *left* to *right* the plots correspond to different calculations of the interaction matrix: ${L_e}^+$ (at each iteration), ${L_e}^+ = {L_{e^*}}^+$ (at equilibrium), and ${L_e}^+ = ({L_e}^+ + {L_{e^*}}^+)/2$.

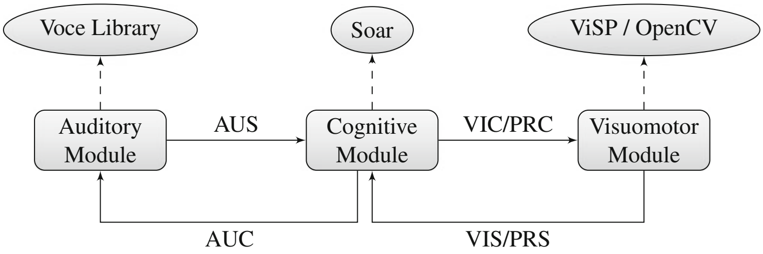


Fig. 2.2 The CRR architecture. The *boxes* represent modules and the *ovals* indicate the libraries wrapped inside the modules. The links between modules indicate topics. *AUS* Auditory sensory, *PRS* proprioceptive sensory, *VIS* visual sensory, *AUC* auditory command, *VIC* visual command, *PRC* proprioceptive command

tolerant to calibration errors by employing visual servoing; where modeling errors are compensated in the control loop (the camera directly measures the task errors).

The platform presents a modular organization (as shown in Fig. 2.2) and is composed by three modules. The cognitive module is responsible for symbolic decision making and learning. The auditory module processes speech recognition. The visuomotor module is in charge of applying the VS control. To enable inter-modular communication, six topics were defined. Topics are named buses over which modules exchange messages. According to the sensory modalities that compose CRR, auditory, proprioceptive and visual topics were defined. The aim of these topics is sending sensory information to the cognitive module. Similarly, the cognitive module sends commands to the auditory, visual and proprioceptive modules.

Hardware Components. The design of CRR aimed to praise the reusability of equipments, so its hardware components were chosen according to a criteria of accessibility in the robotic lab. The project considered a Stäubli TX-40 serial robot manipulator, an AVT MARLIN F-131C camera, and a DELL Vostro 1,500 laptop (Intel Core 2 Duo 1.8 GHz, 800 MHz front-side bus, 4.0 GB DDR2 667 MHz RAM memory, 256 MB NVIDIA GeForce 8,600 M GT graphic card).

Software Components. Three criteria grounded the choice for software technologies: source availability, efficiency and continuity of the development community. The sole exception was the use of SYMORO+ [11], a proprietary automatic symbolic modeling tool for robots. CRR was developed under Ubuntu Oneiric Ocelot and relied on Voce Library V0.9.1, ViSP V2.6.2, the symbolic CA Soar V9.3.2, and ROS Electric. Eclipse Juno V4.2 was used for cording and testing the algorithms.

2.5 Case Study

The experimental situation designed, consisted in a reaching, grasping, and releasing task, involving reinforcement learning. From the inputs received, and based on the rewards or punishments obtained, the robot must learn the optimal sequence policy $\pi : S \rightarrow A$ to execute the task, and thus, to maximize the reward obtained.

2.5.1 Task Definition

The experimenter is positioned in front of the robot for every trial and presents it an object accompanied by a verbal auditory cue (“wait” or “go”). The robot has to choose between sleeping or reaching the object. If the object is reached after a “wait” or the robot goes sleeping after a “go”, the experimenter sends an auditory verbal cue representing punishment (“stop”) and the trial ends. On the contrary, if the robot goes sleeping after getting a “wait” or follows the object after a “go”, it receives an auditory verbal cue representing reward (“great”). After being rewarded for following the object, the experiment enters the releasing phase. If the robot alternated the location for dropping the object it is rewarded, otherwise it is punished. Figure 2.3 presents the reinforcement algorithm.

The robot has two main goals in the experiment. It is required to learn when reaching or sleeping in the presence of the object; and if the object is grasped, to learn to drop it alternatively in one of two containers. Summarizing, the robot is required of perceptive abilities (recognizing the object and speech), visuomotor coordination, and decision making (while remembering events).

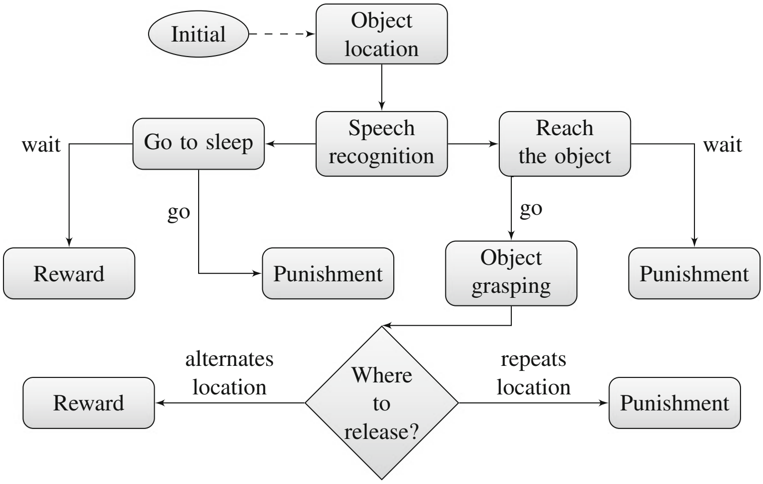


Fig. 2.3 Task reinforcement algorithm

2.5.2 Perception

Object Recognition. The recognition of the object was accomplished using the OpenCV library. The partition of the image into meaningful regions was achievement in two steps. The classification steps includes a decision process applied to each pixel assigning it to one of $C \in \{0 \ \dots \ C-1\}$ classes. For CRR a particular case using $C = 2$ known as *binarization* [12] was used. Formally, it is conceived as a monadic operation taking an image of size $I^{W \times H}$ as input, and producing an image $O^{W \times H}$ as output; such as

$$O[u, v] = f(I[u, v]), \ \forall (u, v) \in I. \tag{2.6}$$

The color image I is processed in HSV color space, and the f function used was

$$f(I[u, v]) = \begin{cases} 1 & \text{if } \epsilon_{\text{i}} < I[u, v] < \epsilon_{\text{f}} \\ 0 & \text{otherwise} \end{cases}. \tag{2.7}$$

The choice of f was based on simplicity and ease of implementation; however, it assumes constant illumination conditions throughout the experiment (which is the case since the environment is illuminated artificially). The thresholds ϵ were set to recognize red objects.

In the description phase the represented sets S are characterized in terms of scalar or vector-valued features such as size, location and shape. A particularly useful class of image features are moments [7], which are easy to compute and can be used to find the location of an object (centroid). For a binary image $B[x, y]$ the $(p+q)$th order moment is defined by

$$m_{pq} = \sum_{y=0}^{y_{\max}} \sum_{x=0}^{x_{\max}} x^p y^q B(x, y). \tag{2.8}$$

Moments can be given a physical interpretation by regarding the image function as a mass distribution. Thus m_{00} is the total mass of the region, and the centroid of the region is given by

$$x_c = \frac{m_{10}}{m_{00}}, y_c = \frac{m_{01}}{m_{00}}. \tag{2.9}$$

After the centroid is obtained, the last step consisted in proportionally defining two points beside it, forming an imaginary line of −45° slope. These two points are the output of the object recognition algorithm, later entered to ViSP to define 2D features and performing the VS control.

Speech Recognition. CRR used the Voce Library to process speech. It required no additional efforts than changing the grammar configuration file to include the vocabulary to be recognized.

2.5.3 Visuomotor Control

In order to perform visuomotor coordination to reach the object, an IBVS strategy was chosen given its robustness to modeling uncertainties [8]. The camera was located in the effector of the robot (eye-in-hand), thus the J_e component of (2.5) is defined by

$$J_e = L_e {}^c V_n {}^n J(q). \tag{2.10}$$

Two visuomotor subtasks were defined: reaching the object and avoiding joint limits.

Primary task. The subtask e consisted in positioning the end-effector in front of the object for grasping it. The final orientation of the effector was not important (assuming a spherical object), therefore, only 3 DOF were required to perform the task. Two 2D point features were used given its simplicity, each of them allowing to control 2 DOF. The resulting interaction matrix L_{e_i} was defined by

$$L_{e_i} = \begin{bmatrix} -1/Z_{e_i} & 0 & x_{e_i}/Z_{e_i} & x_{e_i} y_{e_i} & -(1 + x_{e_i}^2) & y_{e_i} \\ 0 & -1/Z_{e_i} & y_{e_i}/Z_{e_i} & (1 + y_{e_i}^2) & -x_{e_i} y_{e_i} & -x_{e_i} \end{bmatrix}. \tag{2.11}$$

The error vector for the primary task can be expressed by

$$e_i = \left[(x_{s_i} - x_{s_i*}) \; (y_{s_i} - y_{s_i*}) \right]^t. \tag{2.12}$$

Since two points are tracked, the resulting components dimension were $L_e^{4\times6}$ and $e^{4\times1}$.

Secondary Task. The remaining 3 DOF were used to perform the secondary task of avoiding joint limits. The strategy adopted was *activation thresholds* [13]. The secondary task is required only if one (or several) joint is in the vicinity of a joint limit. Thus, thresholds can be defined by

$$\widetilde{q}_{i_{\min}} = q_{i_{\min}} + \rho(q_{i_{\max}} - q_{i_{\min}}), \tag{2.13}$$

and

$$\widetilde{q}_{i_{\max}} = q_{i_{\max}} - \rho(q_{i_{\max}} - q_{i_{\min}}), \tag{2.14}$$

with $0 < \rho < 1/2$.

The vector e_{s} had 6 components, each defined by

$$e_{\text{s}_i} = \begin{cases} \frac{\beta(q_i - \widetilde{q}_{i_{\max}})}{q_{i_{\max}} - q_{i_{\min}}} & \text{if } q_i > \widetilde{q}_{i_{\max}} \\ \frac{\beta(q_i - \widetilde{q}_{i_{\min}})}{q_{i_{\max}} - q_{i_{\min}}} & \text{if } q_i < \widetilde{q}_{i_{\min}} \\ 0 & \text{otherwise} \end{cases}, \tag{2.15}$$

with the scalar constant β regulating the amplitude of the control law due to the secondary task.

2.5.4 Decision Making

Markov Decision Process (MDP) provided the mathematical framework for modeling decision making. The task space was represented by a set of $S = \{S_0, \ldots, S_{10}\}$ states, $A = \{a_0, \ldots, a_8\}$ actions and $P_a(s, s') = \{\alpha_0, \ldots, \alpha_{14}\}$ action-transition probabilities. The simplified MDP representation of the agent is given in Fig. 2.4.

Procedural Knowledge Modeling. Cognitive models in Soar 9.3.2 are stored in long-term production memory as productions. A production has a set of conditions and actions. If the conditions match the current state of working memory (WM), the production fires and the actions are performed. Some attributes of the state are defined by Soar (i.e., *io*, *input-link* and *name*) ensuring the operation of the architecture. The modeler has the choice to define custom attributes, which derives in a great control over the state.

The procedural knowledge implementation in Soar can be conceived as a mapping between an input to an output semantic structure. To develop the case study, it was necessary to define three types of productions: maintenance, MDP and RL rules. The first category includes rules that process inputs and outputs to maintain a consistent state in the WM; a typical task is clearing or putting data into the slots in order to access the modules functionalities. The second category includes rules related to the agent's task, such as, managing the MDP state transitions. The last group involves rules that guarantee the correct functioning of RL; it includes tasks like maintaining

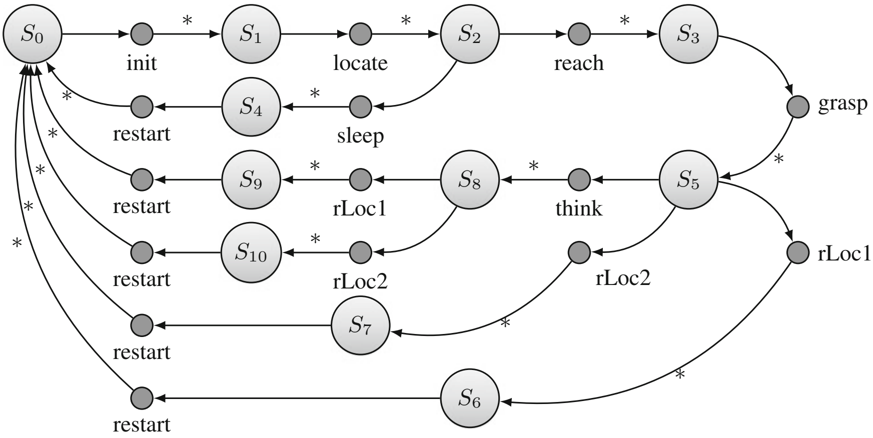


Fig. 2.4 The MDP task model. $* = (\alpha, \rho)$, where α is the transition probability from s to s' when taking the *action*, and ρ is the reward associated with the *state*. From all *actions* there is a link to S_0 (omitted for clarity) modeling errors on the process with probability $1 - \alpha$. The states are: S_0 Started, S_1 initialized, S_2 object located, S_3 object reached, S_4 sleeping, S_5 object grasped, S_6 object released in location *1*, S_7 object released in location *2*, S_8 thinking, S_9 object released in location *1* after thinking, S_{10} object released in location *2* after thinking. The action a_0 initializes the system, a_1 signals the localization of the object, a_2 signals the robot to reach the object, a_3 puts the robot in sleeping mode, a_4 signals the robot to close the gripper, a_5 explores past events, a_6 and a_7 signal the robot to release the object at location *1* or *2* respectively, and a_8 restarts the system. If a state receives a negative feedback from the user $\rho_i = -4$ (punishment). In case of positive feedback, $\rho_i = 2$ (reward)

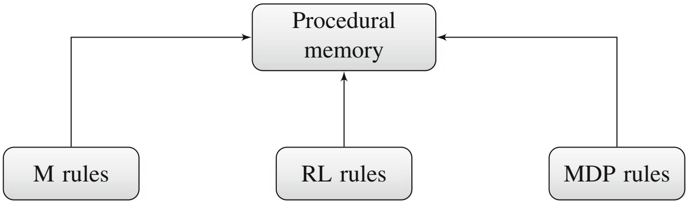


Fig. 2.5 Procedural memory. *M* maintenance, *RL* reinforcement learning, *MDP* mark of decision process

the operators' Q-values, or registering rewards and punishments. Figure 2.5 presents a qualitative view of the contents of the procedural memory. For modeling the case study, a total of 57 productions were defined.

Remembrance of Events. Functionalities in Soar are accessed through testing the current semantic structure of the WM. The same principle applies for querying data in the long term memory. In order to access the episodic or semantic memory, the programmer must define rules placing the query attributes and values on the attribute *epmem* (for episodic retrieval) or *smem* (for semantic retrieval). After each decision cycle, Soar checks the *epmem.command* node to match conditions for

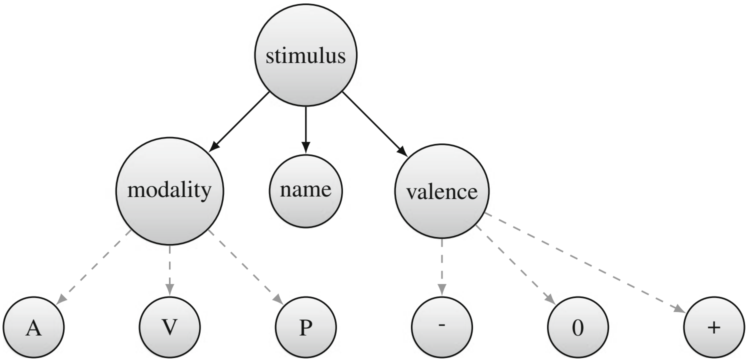


Fig. 2.6 Stimulus semantic knowledge. *A* auditory, *V* visual, *P* proprioceptive

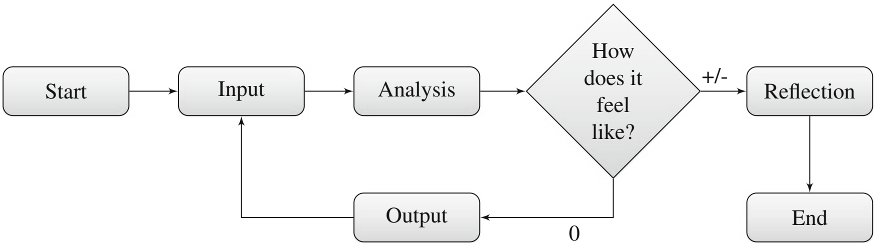


Fig. 2.7 Stimulus processing and reinforcement

episodic retrieval. A copy of the most recent match (if found) will be available on the *epmem.result* for the next decision cycle.

Remembrance of Facts. Facts about the world can be modeled through semantic structures. For the case study, the agent must know what are the stimuli received, or at least, how it feels like in relation to them. Thus, semantic information concerning stimuli was added to the system. The resulting graph was equivalent to a tree of height two (Fig. 2.6). A stimulus has a name, a sensory modality (visual, auditory or proprioceptive) and a valence (positive, negative or neutral).

Reinforcement Learning. The learning by reinforcement can be considered as equivalent to mapping situations to actions, so as to maximize a numerical reward signal [14]. The learner is not told which actions to take, but instead it must discover which actions yield the most reward by trying them. The RL module of Soar is based on the Q-learning algorithm [14]. In the case study a reward is applied whenever the state is not neutral. Figure 2.7 illustrates the processing of the stimuli. When an input arrives, procedural rules query the semantic memory to determine the valence associated with the stimulus. Following an analogy with respect to humans, the agent continues to work if it doesn't feel happy or sad about what it has done; if so, it stops to think about it.

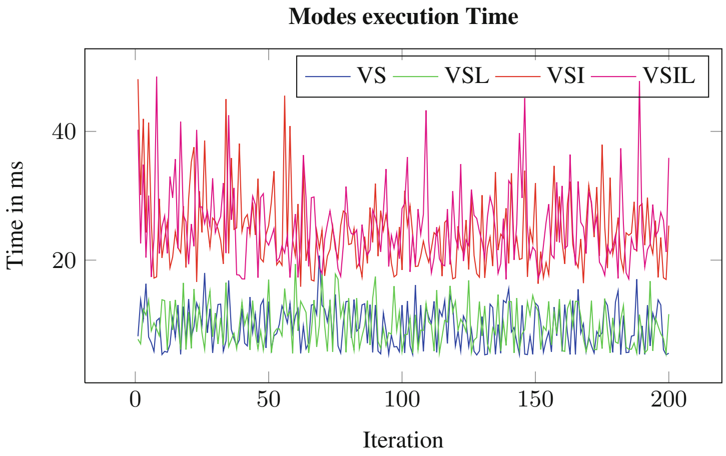


Fig. 2.8 Visuomotor module computing time

2.6 Results

The implementation of the functionalities of CRR took place incrementally. Given the independence between the different modules, each component could be developed and tested individually. The modules were connected to the platform through ROS Etectric; a comprehensive simulation was done, and the results obtained are presented below.

2.6.1 System Performance

The performance of the visuomotor module is quite acceptable for real-time control applications. The module was designed to operate in four different modalities. In the *VS* mode, only visual servoing is available. In the *VSI* mode, it is possible to have a real-time view of the camera. In the *VSL* mode, the system generates log files for joint positions and velocities, feature errors, and camera velocities. Finally, a combination of the last three is allowed in the *VSIL* mode. As it can be seen in Fig. 2.8, a Freq. near to 66 Hz (approx. 15 ms per iteration) can be reached. If the camera view is displayed (which can be useful for debugging but has no importance for execution) the Freq. drops to 20 Hz.

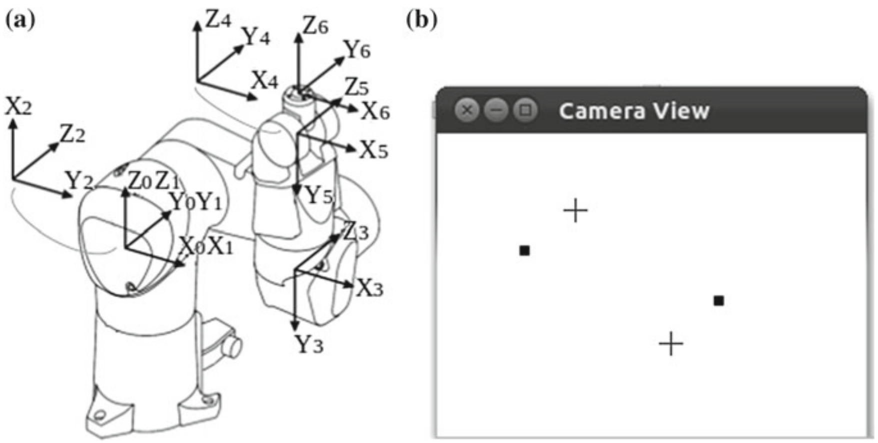


Fig. 2.9 Robot configuration for testing joint limits avoidance. **a** Joint positions in deg: $q_1 = 0$, $q_2 = 90$, $q_3 = -90$, $q_4 = 0$, $q_5 = 0$, $q_6 = 0$. **b** Simulated view, dots are the current feature locations and crosses are the desired locations

2.6.2 *Joint Limit Avoidance*

In order to test the joint limit avoidance property of the system, a simple simulation was designed. The robot was positioned in the configuration displayed in Fig. 2.9a. An object is assumed to be presented to the robot, rotated −10° around the z-axis of the camera frame. The simulated camera view is shown in Fig. 2.9b.

The primary task (moving the robot to the desired view of the features) can be solved in infinite ways given the current singularity between joint frames 4 and 6. For testing the limit avoidance control law, limits of $q_{6_{\min}} = -5°$ and $q_{6_{\max}} = 5°$ were set to joint 6. As it is shown in Fig. 2.10, if just the primary task is performed, the control law generated will mostly operate q_6 and the task will fall in local minima, since $q_{6_{\min}}$ will be reached. On the contrary, as shown in Fig. 2.11, setting a threshold $\rho = 0.5$ (which means it will be active when $q_6 < -2.5°$ or $q_6 > 2.5°$) solves the problem and the joint limit is avoided.

2.6.3 *Learning Task*

The task designed to run over CRR had two learning phases. In order to assess the correctness of the cognitive model and the learning algorithm; two experimental sets were defined. In the experimental set one (ES1), the objective was to teach the robot to identify when reaching the target. The ES1 evaluation consisted of five test cases varying the order of presentation of the clues “wait” and “go”. In all conditions the robot started without prior knowledge (the RL module was reset). The comparison

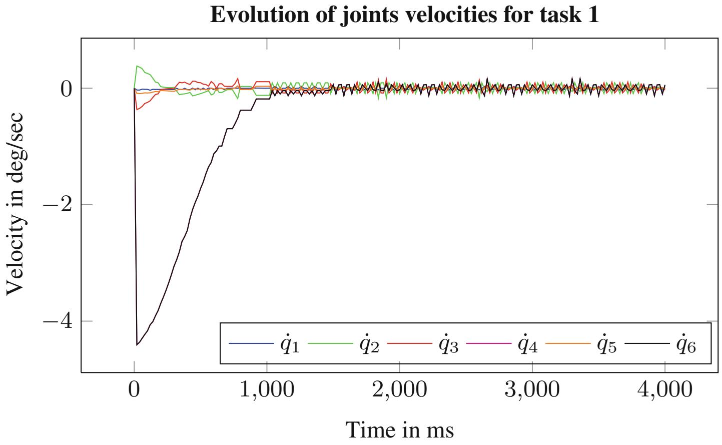


Fig. 2.10 Simulation of VS primary task

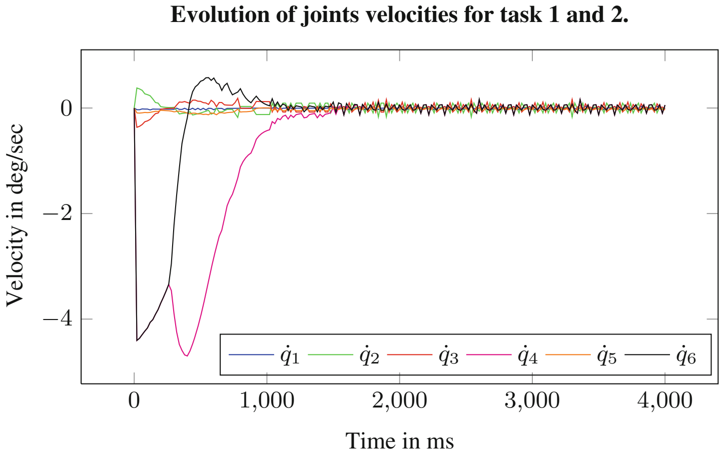


Fig. 2.11 Simulation of VS avoiding joint limits

between a RL and a random police is given in Table 2.1; as it can be seen, the robot was able to learn the task. The experimental set two (ES2) assumes ES1 was accomplished, so the agent properly grasped the object and must now learn where to drop it. The ES2 evaluation showed the agent was able to quickly learn the task using RL, and the resulting Q-values are presented in Table 2.2. For each test case of both ES1 and ES2, the first 20 responses of the robot were registered.

Table 2.1 ES1 evaluation results

Test	RL-S	RL-C	R-S	R-C
C1	17	0.85	8	0.40
C2	18	0.90	11	0.55
C3	17	0.85	12	0.60
C4	18	0.90	9	0.45
C5	18	0.90	10	0.50

RL-S Number of successes applying a RL policy, *RL-C* RL-S/attempts, *R-S* number of successes applying a random policy, *R-C* R-S/attempts

Table 2.2 ES2 evaluation results

Action	Frequency	Reward
think-Remember	15	4.9302
think-release-loc-2-A	1	−2.2800
think-release-loc-2-B	7	6.9741
think-release-loc-1-A	7	6.9741
think-release-loc-1-B	0	0.0000
release-loc-2	2	0.6840
release-loc-1	3	0.4332

The robot attempted to release the object without remembering 5 times (taking the *release-loc-1* and *release-loc-2* actions). However, it learned to maximize the reward by tacking the *think-Remember* action, which was selected 15 times. Finally, after recalling the last location, the agent learned to alternate between the *think-release-loc-2-B* and *think-release-loc-1-A* actions

2.7 Discussion

Starting from the definition of a platform for executing visually guided tasks, a case study based on reinforcement learning was designed and required of perceptive abilities (such as, recognizing the object and speech), visuomotor coordination, and decision making (while remembering events). Different sections of the paper were devoted to detail the design criteria and the development of these components in the CRR platform.

In the contemplated scenario, the recognition of stimuli was accomplished with relative ease. For the case of visual recognition, the OpenCV library proved to be a useful tool by offering a comprehensive set of procedures, thus facilitating the attainment of complex tasks with a reduced number of function calls. For speech recognition, no further effort was required than specifying the vocabulary to be recognized.

In order to ensure visuomotor coordination, the technique of IBVS was chosen with the configuration eye-in-hand to avoid occlusions in the scene. Three DOF of the robot where assigned to the tracking task, while the remaining were assigned to the secondary task of joint limits avoidance. It was observed that both tasks efficiently fulfilled their role in the system. The ViSP library showed to be a valuable tool for implementing real-time visual servoing control laws. The encapsulation of tracking algorithms abstracts the designer from the robust handling of image processing, which led to shorter development times.

The development of cognitive models in Soar presented a slow learning curve. However, the available documentation and resources included in the distribution (specially the Soar Debugger) are sufficient and allowed to identify the errors; and gradually, to understand the concepts behind the architecture.

The MDP framework showed to be a valuable tool for treating RL-based experiments. The integration of the MDP formalism to Soar was a relatively simple task to do, given that the architecture implements the Q-learning algorithm. This algorithm requires of the definition of rules that generate Q-values for each state-action pairs. Soar provides mechanisms for generating these rules, even for problems whose dimensions are not known ahead of time.

The Soar syntax to encode production rules is simple. However, the procedural memory contains more than translations from English of the productions relative to the task (also modeled using the MDP formalism). That is, the cognitive model requires of the procedural knowledge extracted through the methodology of knowledge engineering. But it also requires of rules whose purpose is to manage the WM contents, thus, ensuring coherence during the execution of the agent while accessing the architecture's functionalities (i.e., events and facts remembrance, or RL).

In favor of alleviating the implementation efforts for the MDP representation in the case of similar task spaces; the proposed approach could be extended with the benefits of an ontology-based methodology. Thus, the system could be enhanced with a new component in charge of translating (or mapping) the content represented by the ontology, to the set of production rules that will be executed on the CRR platform.

2.8 Conclusions

This work started from the interest in developing cognitive robotic systems for executing manipulative tasks. To this purpose, an approach emphasizing multidisciplinary theoretical and technical formulations was adopted. A methodological proposal for integrating a psychologically-oriented cognitive architecture to the visual servoing control technique has been presented; and resulted in the development of a modular system capable of auditory and visual perception, decision making, learning and visuomotor coordination. The evaluation of the case study, showed that CRR is a system whose operation is adequate for real-time interactive manipulative applications.

Acknowledgments This research was accomplished thanks to the founding of the National Agency of Research through the EQUIPEX ROBOTEX project (ANR-10-EQX-44), of the European Union through the FEDER ROBOTEX project 2011-2015, and of the Ecole Centrale of Nantes.

References

1. Arbib, M.A., Metta, G., van der Smagt, P.P.: Neurorobotics: from vision to action. In: Springer Handbook of Robotics, pp. 1453–1480 (2008)
2. Newell, A.: Unified Theories of Cognition. William James Lectures, Harvard University Press (1994)
3. Duch, W., Oentaryo, R.J., Pasquier, M.: Cognitive architectures: where do we go from here? In: Proceedings of the IROS workshop on current software frameworks in cognitive robotics integrating different computational paradigms, pp. 122–136, Nice, France 22 Sept 2008
4. Sun, R.: Multi-agent Systems for Society. Springer-Verlag, Berlin (2009)
5. Hanford, S., Long, L.: A cognitive robotic system based on the Soar cognitive architecture for mobile robot navigation, search, and mapping missions. In: PhD thesis, Aerospace Engineering, University Park, Pa, USA (2011)
6. Huelse, M., Hild, M.: A brief introduction to current software frameworks in cognitive robotics integrating different computational paradigms. In: Proceedings of the IROS workshop on current software frameworks in cognitive robotics integrating different computational paradigms. Nice, France, 22 Sept 2008
7. Corke, P.I.: Robotics, Vision & Control: Fundamental Algorithms in Matlab. Springer, Berlin (2011)
8. Chaumette, F., Hutchinson, S.: Visual servo control, part I: basic approaches. IEEE Robot. Autom. Mag. **13**, 82–90 (2006)
9. Kelley, T.D.: Symbolic and sub-symbolic representations in computational models of human cognition: what can be learned from biology? Theor. Psychol. **13**(6), 847–860 (2003)
10. O'Reilly, R., Munakata, Y.: Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain. Bradford Books, MIT Press, Cambridge (2000)
11. Khalil, W., Creusot, D.: Symoro+: a system for the symbolic modelling of robots. Robotica **15**(2), 153–161 (1997)
12. Pratt, W.: Digital Image Processing: PIKS Scientific Inside. Wiley-Interscience publication, Wiley (2007)
13. Marchand, E., Chaumette, F., Rizzo, A.: Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. In: IEEE/RSJ international conference on intelligent robots and systems, IROS'96. vol. 3, Osaka, Japan, pp. 1083–1090 (Nov 1996)
14. Kaelbling, L., Littman, M., Moore, A.: Reinforcement learning: a survey. J. Artif. Intell. Res. **4**, 237–285 (1996)

Chapter 3
Computational Experience with a Modified Newton Solver for Continuous-Time Algebraic Riccati Equations

Vasile Sima

Abstract Improved Newton solvers, with or without line search, for continuous-time algebraic Riccati equations are discussed. The basic theory and algorithms are briefly presented. Algorithmic details, the computational steps, and convergence tests are described. The main results of an extensive performance investigation of the Newton solvers are compared with those obtained using the widely-used MATLAB solver, `care`. Randomly generated systems with orders till 2,000, as well as the systems from the large COMPl$_e$ib collection of examples, are considered. Significantly improved accuracy, in terms of normalized and relative residuals, and often greater efficiency than for `care` have been obtained. The results strongly recommend the use of such algorithms, especially for improving the solutions computed by other solvers.

Keywords Algebraic Riccati equation · Numerical methods · Optimal control · Optimal estimation

3.1 Introduction

The numerical solution of algebraic Riccati equations (AREs) is an essential step in many computational methods for model reduction, filtering, and controller design for linear control systems. Let A, E, $Q \in \mathbb{R}^{n\times n}$, B, $L \in \mathbb{R}^{n\times m}$, $Q = Q^T$, and $R = R^T \in \mathbb{R}^{m\times m}$, with E and R assumed nonsingular. In a compact notation, the generalized continuous-time AREs (CAREs), with unknown $X = X^T \in \mathbb{R}^{n\times n}$, are defined by

$$0 = Q + \mathrm{op}(A)^{\,T} X\,\mathrm{op}(E) + \mathrm{op}(E)^{\,T} X\,\mathrm{op}(A) - \mathcal{L}(X) R^{-1} \mathcal{L}(X)^T =: \mathcal{R}(X), \quad (3.1)$$

V. Sima (✉)
National Institute for Research and Development in Informatics,
Bd. Mareşal Averescu 8–10, 011455 Bucharest, Romania
e-mail: vsima@ici.ro
http://www.ici.ro

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_3

where $\mathcal{L}(X) := L + \mathrm{op}(E)^T XB$. The operator $\mathrm{op}(M)$ represents either M or M^T. An optimal regulator problem involves the solution of an ARE with $\mathrm{op}(M) = M$; an optimal estimator problem involves the solution of an ARE with $\mathrm{op}(M) = M^T$, *input matrix* B replaced (by duality) by the transpose of the *output matrix* $C \in \mathbb{R}^{p \times n}$, and m replaced by p. (In practice, often Q and L are given as $C^T \bar{Q} C$ and $L = C^T \bar{L}$, respectively.) The solutions of an ARE are the matrices X for which $\mathcal{R}(X) = 0$. Usually, what is needed is a *stabilizing solution*, X_s, for which the matrix pair $(\mathrm{op}(A - BK(X_s)), \mathrm{op}(E))$ is stable, where $\mathrm{op}(K(X_s))$ is the gain matrix of the optimal regulator or estimator, and

$$K(X) := R^{-1} \mathcal{L}(X)^T \tag{3.2}$$

(with X replaced by X_s).

There is a vast literature concerning AREs and their use for solving optimal control and estimation problems; see, e.g., the monographs [1, 18, 23] for many theoretical results. The optimization criterion for linear control systems is a quadratic performance index in terms of the system state and control input. By minimizing this criterion, a solution to the optimal systems stabilization and control is obtained, expressed as a state-feedback control law. Briefly speaking, this control law achieves a trade-off between the regulation error and the control effort. The optimal estimation or filtering problem, for systems with Gaussian noise disturbances, can be solved as a dual of an optimal control problem, and its solution gives the minimum variance state estimate, based on the system output. The results of an optimal design are often better suited in practice than those found by other approaches. For instance, pole assignment may deliver too large gain matrices, producing high-magnitude inputs, which might not be acceptable. In both control and estimation problems, including those stated in the H_∞ theory (e.g., [13]), a major computational step is the solution of an ARE. Due to their importance, numerous numerical methods have been proposed for solving AREs; see, for instance, [23, 30]. There are also several software implementation, e.g., in MATLAB [22], or in the SLICOT Library [8–10, 38].

Newton's method for solving AREs has been considered by many authors, for instance, [4, 5, 17, 18, 23, 30]. Moreover, the matrix sign function method for AREs, e.g., [11, 14, 28, 35], uses a specialized Newton's method to compute the square root of the identity matrix of order $2n$. This paper merely reports on implementation details and numerical results. The new solvers have improved functionality, flexibility, reliability, and efficiency. The paper extends the results of [34] in some details, and by investigating the numerical behavior of the current Newton-based CARE solvers for high-order random systems, and for systems from the COMPl$_e$ib collection [21]. It is worth mentioning that Newton's method has been applied in [26] for solving special classes of large-order AREs, using low rank Cholesky factors of the solutions of the Lyapunov equations built during the iterative process [25]. Additional numerical results, for randomly generated systems with $n \le 600$, and comparison with MATLAB and SLICOT solvers are presented in [31]. However, the specialized solvers used (`lp_lrnm` and `lp_lrnm_i`) require the following main assumptions: (1) the matrix A is structured or sparse; (2) the solution X has a small rank in comparison with n. (These solvers use the possibly sparse structure of the matrix A and operations of the form Ab or $A^{-1}b$, where b is a vector.) The solvers discusssed in this paper are general, and can solve large dense problems.

The paper compares the performance of the Newton solvers with or without line search (briefly called *modified* and *standard* Newton solvers, respectively) with the performance of the state-of-the-art commercial solver `care` from MATLAB Control System Toolbox. The MATLAB solver uses a (generalized) eigenvalue approach, based on the results in, e.g., [3, 20, 36]. Relatively recent research, including both theoretical and numerical investigation, has been directed to exploit the Hamiltonian or symplectic structure of the eigenproblem associated to the ARE [6, 7, 27, 32, 33]. A recursive method for computing the positive definite stabilizing solution of an ARE with an indefinite quadratic term has been recently proposed in [19].

One drawback of the Newton's method is its dependence on an initialization, X_0. When searching for a stabilizing solution X_s, the initialization X_0 should also be stabilizing, i.e., $(\text{op}(A - BK(X_0)),\ \text{op}(E))$ should be stable. Finding a suitable initialization can be a difficult task. Stabilizing algorithms have been proposed, mainly for standard systems, e.g., in [15, 17, 29, 39]. However, often these algorithms produce a matrix X_0 and/or the following several matrices X_i, $i = 1, 2, \ldots$ (computed by the Newton's method), with very large norms, and the Newton solvers may encounter severe numerical difficulties. For this reason, Newton's method is best used for iterative improvement of a solution or as defect correction method [24], delivering the maximal possible accuracy when starting from a good approximate solution. Moreover, it is preferred in implementing certain fault-tolerant systems, which require controller updating, see, e.g., [12] and the references therein.

The organization of the paper is as follows. Section 3.2 starts by summarizing the basic theory and Newton's algorithms for CAREs. Algorithmic details, computation of the Newton direction, computation of the Newton step size, and convergence tests are discussed in separate subsections. Section 3.3 presents the main results of an extensive performance investigation of the solvers based on Newton's method, in comparison with the MATLAB solver `care`. Randomly generated systems with order till 1,000 (but also a system with order 2,000), as well as systems from the COMPl$_e$ib collection [21], are considered in the two subsections. Section 3.4 summarizes the conclusions.

3.2 Basic Theory and Newton's Algorithms

The following assumptions are made.
Assumptions A

- Matrix E is nonsingular.
- Matrix pair $(\,\text{op}(E)^{-1}\,\text{op}(A)\,,\ \text{op}(E)^{-1}B)$ is stabilizable.
- Matrix R is positive definite ($R > 0$).
- A stabilizing solution X_s exists and it is unique.

The algorithm considered in the sequel is an enhancement of Newton's method, employing a *line search* procedure to minimize the residual along the Newton direction.

The conceptual algorithm can be stated in the following form:

Algorithm N Newton's method with line search for CARE.
Input: The matrices E, A, B, Q, R, and L, and an initial matrix $X_0 = X_0^T$.
Output: The approximate solution X_k of CARE.
FOR $k = 0, 1, \ldots, k_{\max}$, DO

1. If convergence or non-convergence is detected, return X_k and/or a warning or error indicator value.
2. Compute $K_k := K(X_k)$ with (3.2) and $\text{op}(A_k)$, where $A_k = \text{op}(A) - BK_k$.
3. Solve in N_k the continuous-time generalized Lyapunov equation

$$\text{op}(A_k)^T N_k \,\text{op}(E) + \text{op}(E)^T N_k \,\text{op}(A_k) = -\mathcal{R}(X_k). \tag{3.3}$$

4. Find a step size t_k which minimizes, with respect to t, the squared Frobenius norm $\|\mathcal{R}(X_k + tN_k)\|_F^2$.
5. Update $X_{k+1} = X_k + t_k N_k$.

END

Standard Newton's algorithm is obtained by taking $t_k = 1$ at Step 4 at each iteration. When the initial matrix X_0 is far from a Riccati equation solution, the modified Newton's method, with line search, often outperforms the standard Newton's method.

Basic properties for the standard and modified Newton's algorithms for CAREs can be stated as follows [4]:

Theorem 1 (Convergence of Algorithm N, Standard Case) *If the Assumptions A hold, and X_0 is stabilizing, then the iterates of the Algorithm N with $t_k = 1$ satisfy*

(a) All matrices X_k are stabilizing.
(b) $X_s \le \cdots \le X_{k+1} \le X_k \le \cdots \le X_1$.
(c) $\lim_{k\to\infty} X_k = X_s$.
(d) Global quadratic convergence: There is a constant $\gamma > 0$ such that

$$\|X_{k+1} - X_s\| \le \gamma \|X_k - X_s\|^2, \quad k \ge 1.$$

Theorem 2 (Convergence of Algorithm N) *If the Assumptions A hold, X_0 is stabilizing, $(\,op(E)^{-1}\,op(A)\,,\; op(E)^{-1}B)$ is controllable and $t_k \ge t_L > 0$, for all $k \ge 0$, then the iterates of the Algorithm N satisfy*

(a) All iterates X_k are stabilizing.
(b) $\|\mathcal{R}(X_{k+1})\|_F \le \|\mathcal{R}(X_k)\|_F$ *and equality holds if and only if* $\mathcal{R}(X_k) = 0$.
(c) $\lim_{k\to\infty} \mathcal{R}(X_k) = 0$.
(d) $\lim_{k\to\infty} X_k = X_s$.
(e) In a neighbourhood of X_s, the convergence is quadratic.

(f) $\lim_{k\to\infty} t_k = 1$.

Theorem 2 does not ensure monotonic convergence of the iterates X_k in terms of definiteness, contrary to the standard case (Theorem 1, item (b)). On the other hand, under the specified conditions, Theorem 2 states the monotonic convergence of the residuals to 0, which is not true for the standard algorithms. It is conjectured that Theorem 2 also holds under the weaker assumption of stabilizability instead of controllability. This is supported by the numerical experiments.

3.2.1 Algorithmic Details

The essential steps of Algorithm N will be detailed below.

Equation (3.1) can be rewritten in a simpler form. Specifically, setting

$$\tilde{A} = A - BR^{-1}L^T, \quad \tilde{Q} = Q - LR^{-1}L^T, \quad G := BR^{-1}B^T, \tag{3.4}$$

after redefining A and Q as $\tilde{A}$ and $\tilde{Q}$, respectively, (3.1) reduces to

$$0 = \operatorname{op}(A)^T X \operatorname{op}(E) + \operatorname{op}(E)^T X \operatorname{op}(A) - \operatorname{op}(E)^T XGX \operatorname{op}(E) + Q =: \mathcal{R}(X), \tag{3.5}$$

or, in the standard case ($E = I_n$), to

$$0 = \operatorname{op}(A)^T X + X \operatorname{op}(A) - XGX + Q =: \mathcal{R}(X). \tag{3.6}$$

The tilde transformations in (3.4) eliminate the matrix L from the formulas to be used by Newton's algorithms. It is more economical to solve (3.5) or (3.6) than (3.1), since otherwise the calculations involving L must be performed at each iteration. In this case, the matrix K_k is no longer computed in Step 3.2, and $A_k = \operatorname{op}(A) - GX_k \operatorname{op}(E)$ (or $A_k = \operatorname{op}(A) - DD^T X_k \operatorname{op}(E)$, if G is factored, $G = DD^T$).

Algorithm N was implemented in a Fortran 77 subroutine, `SG02CD`, and few auxiliary routines, following the SLICOT Library [9, 37, 38] implementation and documentation standards.[1] The implementation deals with generalized AREs, possibly for the discrete-time case, without inverting the matrix E. This is very important for numerical reasons, especially when E is ill-conditioned. Standard AREs (including the case when E is specified as I_n, or even `[]` in MATLAB), are solved with the maximal possible efficiency. Moreover, both control and filter AREs can be solved by the same routine, using an option ("mode") parameter, which specifies the op operator. The matrices A and E are not transposed. It it possible to also avoid the transposition for C and L, for the filter equation, but this is less important and more difficult to implement. (Some existing lower-level routines do not cover the transposed case.) Symmetry is used whenever possible. Common subexpressions of

[1] See http://www.slicot.org.

matrix products are evaluated only once, and the sequence of multiplications is optimized, depending on the n and m values. A new block algorithm is used for computing the matrix product MN, when the result is symmetric (e.g., when $M = \mathrm{op}(E)^{\,T}X$, and $N = GX\,\mathrm{op}(E)$).

The implemented algorithm solves either the generalized CARE (3.5) or standard CARE (3.6) using Newton's method with or without line search. The selection is made using another option. There is an option for solving related AREs with the minus sign replaced by a plus sign in front of the quadratic term. Moreover, instead of the symmetric matrix $G = BR^{-1}B^T$, the matrix B and the matrix R, or its Cholesky factor, may be given. The iteration is started by an initial matrix X_0, which can be omitted, if the zero matrix can be used. If X_0 is not stabilizing, and finding X_s is not required, Algorithm N will converge to another solution of CARE. Either the upper, or lower triangles, not both, of the symmetric matrices Q, G (or R), and X_0 need to be stored. Since the solution computed by a Newton algorithm generally depends on initialization, another option specifies if the stabilizing solution X_s is to be found. In this case, the initial matrix X_0 must be stabilizing, and a warning is issued if this property does not hold; moreover, if the computed X is not stabilizing, an error is issued. The optimal size of the real working array can be queried, by setting its length to -1. Then, the subroutine returns immediately, with the first entry of that array set to the optimal size. A maximum allowed number of iteration steps, $k_{\max}$, is specified on input, and the number of iteration steps performed, k_s, is returned on exit.

If $m \le cn$, where $c = 3/4$ (or $c = 3/5$, for the standard Newton solver), the algorithm is faster if a factorization $G = DD^T$ is used instead of G. Usually, the routine uses the Cholesky factorization of the matrix R, $R = L_r^T L_r$, and computes $D = BL_r^{-1}$. The standard theory assumes that R is positive definite. But the routine works also if this assumption does not hold numerically, by using the UDU^T or LDL^T factorization of R. In that case, the current implementation uses G, and not its factors, even if $m \le cn$. However, if $m > cn$, but the norm of G is too large, then its factor D is used during the iterations, in order to enhance the numerical accuracy, even if the efficiency somewhat diminishes.

The arrays holding the data matrices A and E are unchanged on exit. Array `B` stores either B or G. On exit, if B was given, and $m \le cn$, `B` returns the matrix $D = BL_r^{-1}$, if the Cholesky factor L_r can be computed. Otherwise, array `B` is unchanged on exit. Array `Q` stores matrix Q on entry and the computed solution X on exit. If matrix R or its Cholesky factor is given, it is stored in array `R`. On exit, `R` contains either the Cholesky factor, or the factors of the UDU^T or LDL^T factorization of R, if R is found to be numerically indefinite. In that case, the interchanges performed for the UDU^T or LDL^T factorization are stored in an auxiliary integer array.

The basic stopping criterion for the iterative process is stated in terms of a normalized residual, r_k, and a tolerance τ. If

$$r_k := r(X_k) := \|\mathcal{R}(X_k)\|_F / \max(1, \|X_k\|_F) \le \tau, \tag{3.7}$$

the iterative process is successfully terminated at iteration $k_s = k$. If $\tau \le 0$, a default tolerance is used, defined in terms of the Frobenius norms of the given matrices, and relative machine precision, ε_M, namely

$$\tau = \min\left(\varepsilon_M \sqrt{n} \big(\|E\|_F \, (2\, \|A\|_F + \|G\|_F \|E\|_F) + \|Q\|_F \big), \sqrt{\varepsilon_M} \right). \tag{3.8}$$

When G is given in factorized form (3.4), then $\|G\|_F$ in (3.8) is replaced by $\|D\|_F^2$. When E is identity, the factors involving its norm are omitted. The second operand of min in (3.8) was introduced to prevent deciding convergence too early for systems with very large norms for A, E, G, and/or Q.

The finally computed normalized residual is also returned. Moreover, approximate closed-loop system poles, as well as min(k_s, 50)+1 values of the residuals, normalized residuals, and Newton steps are returned in a working array.

Several approaches have been tried in order to reduce the number of iterations. One of them was to set $t_k = 1$ whenever $t_k \le \sqrt{\varepsilon_M}$. Often, but especially in the first iterations, the computed optimal steps t_k are too small, and the residual decreases too slowly. This is called *stagnation*, and remedies are used to escape stagnation. Specifically, the last computed k_B residuals are stored in the first k_B entries of an array RES. If $\|\hat{\mathcal{R}}(X_k + t_k N_k)\|_F > \tau_s \|\mathcal{R}(X_{k-k_B})\|_F > 0$, then $t_k = 1$ is used. Here, $\hat{\mathcal{R}}(X_k + t_k N_k)$ is an estimate of the residual obtained using (3.11). The current implementation uses $\tau_s = 0.9$ and sets $k_B = 2$, but values as large as $k_B = 10$ can be used by changing this parameter. The first k_B entries of array RES are reset to 0 whenever $t_k = 1$ is applied.

Pairs of symmetric matrices are stored economically, to reduce the workspace requirements, but preserving the two-dimensional array indexing, for efficiency. Specifically, the upper (or lower) triangle of X_k and the lower (upper) triangle of $\mathcal{R}(X_k)$ are concatenated along the main diagonals in a two-dimensional $n(n+1)$ array, and similarly for G and a copy of the matrix Q, if G is used. Array Q itself is also used for (temporarily) storing the residual matrix $\mathcal{R}(X_k)$, as well as the intermediate matrices X_k and the final solution.

3.2.2 Computation of the Newton Direction

The algorithm computes the initial residual matrix $\mathcal{R}(X_0)$ and the matrix $\text{op}(A_0)$, where $A_0 := \text{op}(A) \pm G X_0 \text{op}(E)$. If no initial matrix X_0 is given, we set $X_0 = 0$, $\mathcal{R}(X_0) = Q$ and $\text{op}(A_0) = A$.

At the beginning of the iteration k, $0 < k \le k_{\max}$, the algorithm decides to terminate or continue the computations, based on the current normalized residual $r(X_k)$. (At $k = 0$, the calculations continue, to allow improving a good initialization.) If $r(X_k) > \tau$, a standard (if $E = I_n$) or generalized (otherwise) Lyapunov equation

$$\text{op}(A_k)^T N_k \, \text{op}(E) + \text{op}(E)^T N_k \, \text{op}(A_k) = -\sigma \mathcal{R}(X_k), \tag{3.9}$$

is solved in N_k (the Newton direction), using SLICOT subroutines. The scalar $\sigma \leq 1$ is set by the Lyapunov solvers in order to prevent solution overflowing. Normally, $\sigma = 1$.

Another option is to scale the matrices A_k and E (if E is general) for solving the Lyapunov equations, and suitably update their solutions. Note that the LAPACK subroutines `DGEES` and `DGGES` [2], which are called by the SLICOT standard and generalized Lyapunov solvers, respectively, to compute the real Schur(-triangular) form, do not scale the coefficient matrices. Just column and row permutations are performed, to separate isolated eigenvalues. For some examples, the convergence was not achieved in a reasonable number of iterations. This difficulty was removed by the scaling included in the Newton code.

3.2.3 Computation of the Newton Step Size

The procedure for computing the optimal size of the Newton step (the line search) minimizes the Frobenius norm of the residual matrix along the Newton direction, N_k. Specifically, the optimal step size t_k is given by

$$t_k = \arg \min_t \|\mathcal{R}(X_k + tN_k)\|_F^2. \tag{3.10}$$

It is proved [4] that, in certain standard conditions, an optimal t_k exists, and it is in the "canonical" interval [0, 2]. Computationally, t_k is found as the argument of the minimal value in [0, 2] of a polynomial of order 4. Indeed,

$$\mathcal{R}(X_k + tN_k) = (1 - t)\mathcal{R}(X_k) - t^2 V_k, \tag{3.11}$$

where $V_k = \operatorname{op}(E)^T N_k G N_k \operatorname{op}(E)$. Therefore, the minimization problem (3.10) reduces to the minimization of the quartic polynomial [4]

$$f_k(t) = \operatorname{trace}(\mathcal{R}(X_k + tN_k)^2) = \alpha_k(1 - t)^2 - 2\beta_k(1 - t)t^2 + \gamma_k t^4, \tag{3.12}$$

where $\alpha_k = \operatorname{trace}(\mathcal{R}(X_k)^2)$, $\beta_k = \operatorname{trace}(\mathcal{R}(X_k)V_k)$, $\gamma_k = \operatorname{trace}(V_k^2)$.

In order to solve the minimization problem (3.10), a cubic polynomial (the derivative of $f_k(t)$) is set up, whose roots in [0, 2], if any, are candidates for the solution of the minimum residual problem. The roots of this cubic polynomial are computed by solving an equivalent 4-by-4 standard or generalized eigenproblem, following [16]. Depending on the magnitude of the polynomial coefficients, a matrix or matrix pencil is built, whose eigenvalues are the roots of the given polynomial, and they are computed using the QR and QZ algorithms.

A candidate solution should satisfy the following requirements: (i) it is real; (ii) it is in the interval [0, 2]; (iii) the second derivative of the cubic polynomial is positive.

If no solution is found, then t_k is set equal to 1. If two solutions are found, then t_k is set to the value corresponding to the minimum residual.

3.2.4 Convergence Tests and Updating the Current Iterate

The next action is to check if the line search stagnates and/or the standard Newton step is to be preferred. If $n > 1, k \leq 10, t_k < 0.5, \varepsilon_M^{1/4} < r_k < 1$, and $\|\hat{\mathcal{R}}(X_k + t_k N_k)\|_F \leq 10$, or $\|\hat{\mathcal{R}}(X_k + t_k N_k)\|_F > \tau_s \|\mathcal{R}(X_{k-k_B})\|_F$ (i.e., stagnation is detected), then a standard Newton step ($t_k = 1$) is used.

Another test is to check if updating X_k is meaningful. The updating is done if $t_k \|N_k\|_F > \varepsilon_M \|X_k\|_F$. If this is the case, set $X_{k+1} = X_k + t_k N_k$, and compute the updated matrices $\mathrm{op}(A_{k+1})$ and $\mathcal{R}(X_{k+1})$. Otherwise, the iterative process is terminated and a warning value is set, since no further improvement can be expected. Although the computation of the residual $\mathcal{R}(X_k + t_k N_k)$ can be efficiently performed by updating the residual $\mathcal{R}(X_k)$ via (3.11), the original data is used, since the updating formula (3.11) could suffer from severe numerical cancellation, and could compromise the accuracy of the intermediate results.

Then, $\|X_{k+1}\|_F$ and r_{k+1} are computed, and $k = k + 1$ is set. If the chosen step was not a Newton step, but the residual norm increased compared to the previous iteration, i.e., $\|\mathcal{R}(X_{k+1})\|_F \geq \|\mathcal{R}(X_k)\|_F$, but it is less than 1, and the normalized residual is less than $\varepsilon_M^{1/4}$, then the iterative process is terminated and a warning value is set. Otherwise, the iteration continues.

3.3 Numerical Results

This section presents some results of an extensive performance investigation of the solvers based on Newton's method. The numerical results have been obtained on an Intel Core i7-3820QM portable computer at 2.7 GHz, with 16 GB RAM, with the relative machine precision $\epsilon_M \approx 2.22 \times 10^{-16}$, using Windows 7 Professional (Service Pack 1) operating system (64 bit), Intel Visual Fortran Composer XE 2011 and MATLAB 8.0.0.783 (R2012b). The SLICOT-based MATLAB executable MEX-function has been built using MATLAB-provided optimized LAPACK and BLAS subroutines [2].

3.3.1 Randomly Generated Systems

A first set of tests refer to CAREs (3.5) with initial matrices E, A, B, L, Q, and R randomly generated from a uniform distribution in the (0, 1) interval, with n and m

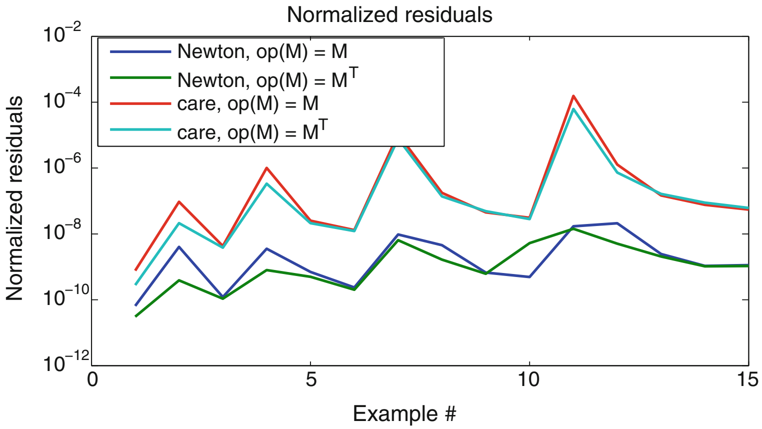


Fig. 3.1 The normalized residuals for random examples using Newton solver with line search and `care`; $n = 200 : 200 : 1,000$, $m = 200 : 200 : n$

set as $n = 200 : 200 : 1,000$, $m = 200 : 200 : n$ (in MATLAB notation). The generated matrix E was stabilized by subtracting $100 \cdot \|E\|_2$ from the diagonal. The generated matrices Q and R were modified by adding n and m, respectively, to the diagonal entries, and then each of them was symmetrized, by adding its transpose. The generated matrix L was divided by 100. We then used the MATLAB function `care` from the Control System Toolbox [22] with inputs A, B, Q, R, L, and E, and stabilized A using $A := A - BF$, where F is the feedback gain matrix returned by `care`. A new Riccati equation was solved using the modified A and the other matrices. This allowed us to set to zero the initial matrix X_0. For the Newton solvers, we removed the effect of L using (3.4). Fifteen CARE problems have been generated. For each CARE, various options have been tried (e.g., use either the upper or lower part of symmetric matrices, use the two values of $\text{op}(M)$, use either the matrices B and R, or the matrix G). The default tolerance, computed by the Newton solvers when the input value is non-positive, has been used.

Figure 3.1 presents the normalized residuals for the random examples solved using Newton solver with line search, and `care`. Figure 3.2 presents the CPU times (computed using the MATLAB pair functions `tic` and `toc`). The y-axis is scaled logarithmically, for better clarity, since the CPU times vary significantly. For the largest example, the run time for the modified Newton solver and $\text{op}(M) = M$ is about half the run time for `care`.

Similar results have been obtained with the standard Newton solver. Both Newton solvers were significantly more accurate (with one exception for the standard solver), and almost always faster than `care`. For this set of tests, the problems with $\text{op}(M) = M^T$ needed more iterations and CPU time for Newton solvers than those with $\text{op}(M) = M$, especially for the standard solver. Indeed, this solver was most often over 50 % faster than `care` for $\text{op}(M) = M$, but often over 20 % slower than `care` for $\text{op}(M) = M^T$. The Euclidean norm of the vectors of normalized residuals

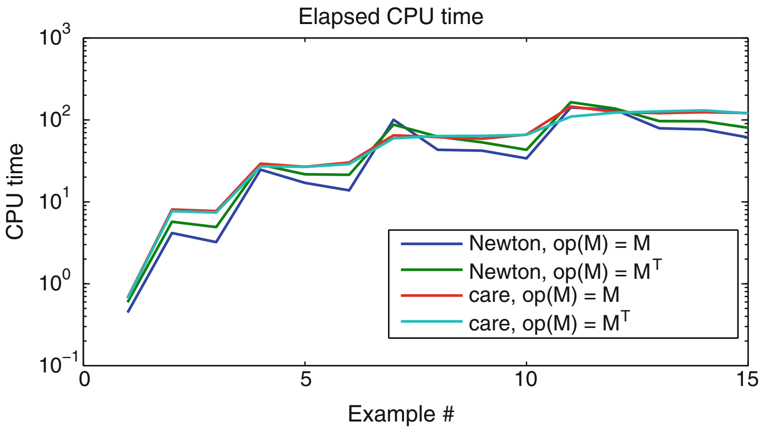


Fig. 3.2 The CPU times for random examples using Newton solver with line search and `care`; $n = 200 : 200 : 1,000$, $m = 200 : 200 : n$

Table 3.1 Normalized residuals 2-norms and mean number of iterations for random examples, $\mathrm{op}(M) = M$

	Line search	Standard Newton	`care`
$\|r_{1:15}\|_2$	2.98×10^{-8}	3.01×10^{-8}	1.55×10^{-4}
$\frac{1}{15}\sum_{i=1}^{15} k_s^i$	5.6	5.33	–

(one normalized residual for each example) and the mean number of iterations are shown in Table 3.1 for the case $\mathrm{op}(M) = M$.

We have also solved a problem with $n = m = 2,000$, built as described above. The modified Newton solver needed 4 iterations when $\mathrm{op}(M) = M$, and 7 iterations when $\mathrm{op}(M) = M^T$. The CPU times were about 793 and 1,360 s, and the normalized residuals were 8.41×10^{-9} and 4.4×10^{-9}, respectively. MATLAB `care` needed about 1,350 and 1,530 s, and the normalized residuals were 2.31×10^{-7} and 3.66×10^{-7}, respectively.

3.3.2 Systems from the COMPl$_e$ib Collection

Other tests have been performed for linear systems from the COMPl$_e$ib collection [21]. This collection contains 124 standard continuous-time examples (with $E = I_n$), with several variations, giving a total of 168 problems. All but 16 problems (for systems of order larger than 2,000, with matrices in sparse format) have been tried. The performance index matrices Q and R have been chosen as identity matrices of suitable sizes. The matrix L was always zero. Most often we used the default tolerance.

In a series of tests, we used $X_0 = 0$, if A is stable; otherwise, we tried to initialize the Newton solvers with a matrix computed using the algorithm in [15], and when this algorithm failed to deliver a stabilizing initialization, we used the solution provided by the MATLAB function `care`. A zero initialization was used for 44 stable examples. Stabilization algorithm was tried on 107 unstable systems, and succeeded for 91 examples. Failures occurred for 16 examples. With default tolerance, modified Newton solver improved the accuracy of the `care` solution for 15 examples. (Only the solution for example ROC5 could not be improved.) The function `care` failed to solve the Riccati equation for example REA4. This example has been excluded from our tests, because it is unstabilizable.

We tried both standard and modified Newton's method, with or without balancing the coefficient matrices of the Lyapunov equations. The modified solver needed more iterations than the standard solver for 10 examples only. The cumulative number of iterations with modified and standard solver for all 150 examples was 1,654 and 2,289, respectively, without balancing, and 1,657 and 2,279, respectively, with balancing. The mean number of iterations was about 11, for the modified solver, and 15.2, for the standard solver. We tried also to use the stabilization algorithm whenever possible, including for stable A matrices. Doing so, the total number of iterations without balancing was 1,796 and 2,208, respectively (1,784 and 2,207, with balancing). The largest number of iterations, 34, was applied for example CM5_IS, with order $n = 480$, and $m = 1$.

Figure 3.3 shows the normalized residuals for the COMPl$_e$ib examples. For clarity, only the results for modified Newton solver without balancing and for `care` are plotted. For the TL example, the normalized residual is 2.13×10^3 when using `care`, but 1.09×10^{-3} for the Newton solver (and 1.32×10^{-3}, using a stabilizing $X_0 \neq 0$). The matrices A and B of this example have norms of order 10^{14} and are poorly scaled (the minimum magnitude in A is of order 10^{-4}). Omitting example TL, the maximum normalized residual was of order 10^{-6} for the standard Newton solver, and of order 10^{-9} (10^{-10} with balancing) for the modified solver and `care`.

Figure 3.4 shows the relative residuals, computed in a similar manner with that used in `care`. The maximum value is 8.98×10^{-9} for the modified Newton solver (for example ROC5), 3.16×10^{-5} for `care` (for example TL), and 1 for the standard Newton solver (also for TL). Omitting TL, the maximum value was of order 10^{-7} for the standard Newton solver and of order 10^{-6} for `care`.

Similarly, Fig. 3.5 shows the elapsed CPU times. Although the modified Newton method was faster than `care` for 100 examples, out of 150, the sum of the CPU times was about 64 % larger than for `care`. The main reason is that, with the chosen initialization, some large examples (mainly, 15 examples in the HF2D class) required at least 19 iterations. The standard Newton solver was globally over 25 % slower than the solver with line search. The balancing option increased the CPU times by less than 4 % in both cases. When using stabilizing $X_0 \neq 0$, the speed-up of the modified Newton solver increased by about 30 %; the main contribution came from solving the CARE for example CM6 ($n = 960$, $m = 1$) in just one iteration, compared to 19 iterations needed when $X_0 = 0$ was used. (The stabilization algorithm failed for CM6, so X_0 was set to the `care` solution.) Clearly, a good initialization could significantly reduce the number of iterations.

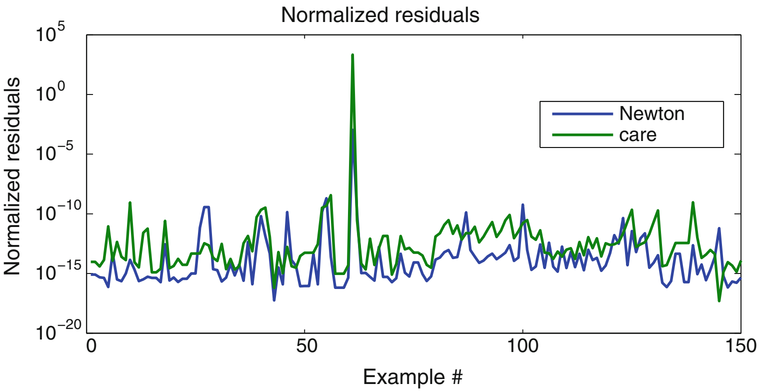


Fig. 3.3 The normalized residuals for examples from the COMPl$_e$ib collection, using Newton solver with line search without balancing and `care`

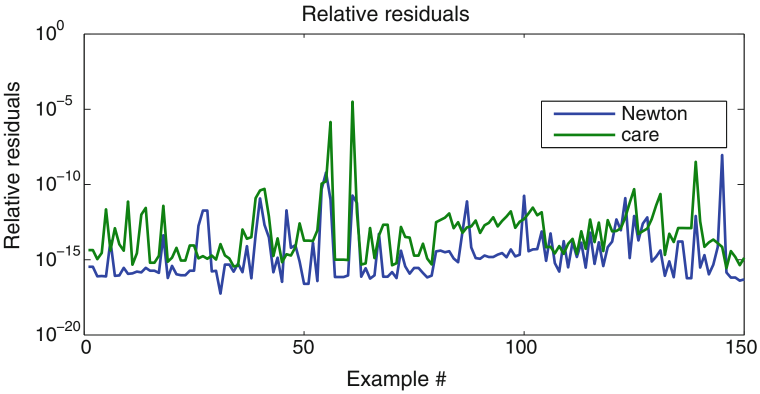


Fig. 3.4 The relative residuals for examples from the COMPl$_e$ib collection, using Newton solver with line search and `care`

Table 3.2 Summary of performance results for small tolerance τ and initialization by MATLAB `care`

τ	Total number of iterations	Sum of CPU times
10^{-12}	198	10.83
10^{-14}	257	23.79
ϵ_M	344	26.23

In another series of tests, we used the solution returned by `care` to initialize the Newton solvers for all COMPl$_e$ib examples, with values for the tolerance parameter τ set to 10^{-12}, 10^{-14}, and ϵ_M. For $\tau = 10^{-12}$, the behavior was identical with that for the default τ. In particular, example TL needed 11 iterations. For $\tau = 10^{-14}$, example TL needed 50 iterations without convergence (the tolerance being too small), one example needed 6 iterations, 2 examples needed 5 iterations, 3 examples needed 4 iterations, 14 examples needed 3 iterations, 9 examples needed 2 iterations, 119

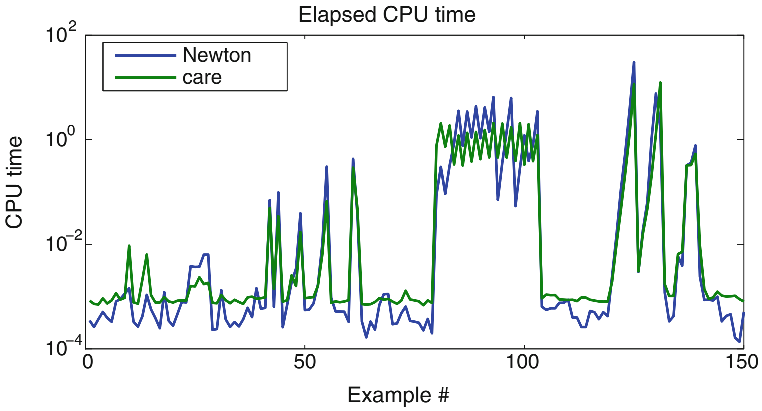


Fig. 3.5 The elapsed CPU time needed by the Newton solver with line search and MTLAB `care` for examples from the COMPl$_e$ib collection

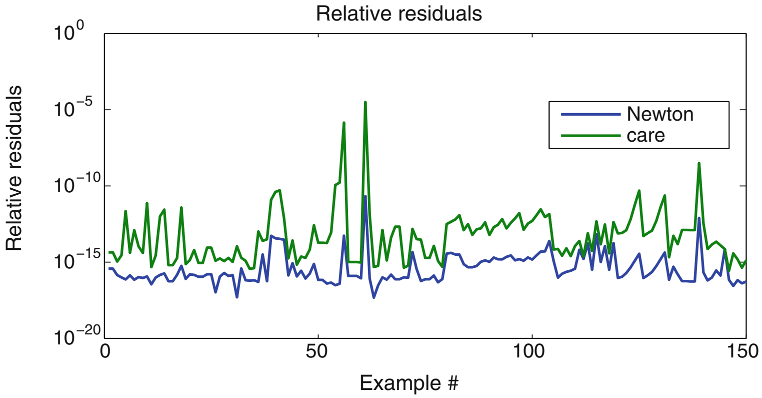


Fig. 3.6 The relative residuals for examples from the COMPl$_e$ib collection, using Newton solver with line search without balancing, initialization using `care` and tolerance relative machine precision.

examples needed one iteration, and ROC5 needed no iteration. For ROC5 example, the Newton solvers found that no improvement of X_0 is numerically possible, since the norm of the correction $t_0 N_0$, 3.07×10^{-14}, was too small compared to the norm of X_0, which is 1.41×10^4. (The normalized residual for X_0 was 4.88×10^{-18}.) Omitting TL, the maximum normalized residual reduced to 6.9×10^{-13} for $\tau = 10^{-12}$, and to about 3.5×10^{-13} for smaller values of τ. Performance results are summarized in Table 3.2.

Figure 3.6 shows the relative residuals for the Newton solver with line search, initialized by `care`, and with tolerance $\tau = \epsilon_M$. Except for ROC5, Newton solvers always succeeded to reduce the residuals, often by several orders of magnitude, compared to `care`. Figure 3.7 shows by a bar graph the size of this improvement.

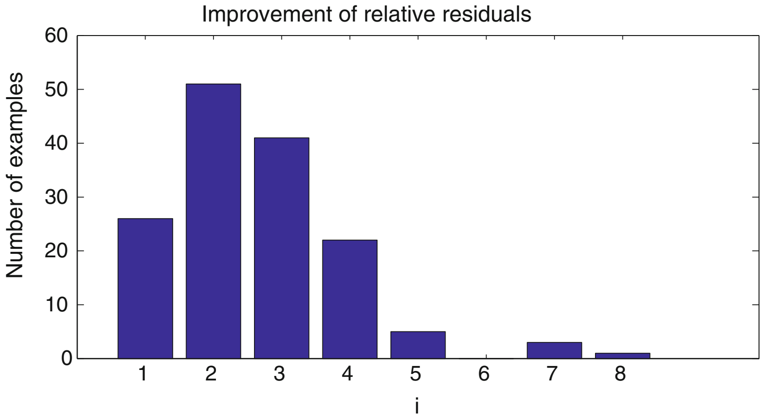


Fig. 3.7 Bar graph showing the improvement of the relative residuals for examples from the COMPl$_e$ib collection, using Newton solver with line search without balancing, initialization using `care` and tolerance relative machine precision. The height of the ith *vertical bar* indicates the number of examples for which the improvement was between i-1 and i orders of magnitude

Specifically, the improvement is of seven orders of magnitude for one example, six orders for three examples, four orders for five examples, etc. For 114 examples, the improvement is between one and three (inclusive) orders of magnitude.

3.4 Conclusions

Basic theory and improved algorithms for solving continuous-time algebraic Riccati equations using Newton's method with or without line search have been presented. Algorithmic details for the developed solvers, the main computational steps (finding the Newton direction, finding the Newton step size), and convergence tests are described. The usefulness of such solvers is demonstrated by the results of an extensive performance investigation of their numerical behavior, in comparison with the results obtained using the widely-used MATLAB function `care`. Randomly generated systems with orders till 1,000 (and even a system with order 2,000), as well as the systems from the large COMPl$_e$ib collection, are considered. The numerical results most often show significantly improved accuracy (measured in terms of normalized and relative residuals), and greater efficiency. The results strongly recommend the use of such algorithms, especially for improving, with little additional computing effort, the solutions computed by other solvers.

Acknowledgments Part of this work was done in a research stay at the Technical University (TU) Chemnitz, Germany, during November 1–December 20, 2005, with financial support from the German Science Foundation. The cooperation with Peter Benner from TU Chemnitz and Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany, is much acknowledged. Thanks are also addressed to Martin Slowik from Institut für Mathematik, TU Berlin, who worked out (till 2005) a preliminary version of the SLICOT codes for CAREs. The work has been recently resumed by the author. Finally, the continuing support from the NICONET e.V. is warmly acknowledged.

References

1. Anderson, B.D.O., Moore, J.B.: Linear Optimal Control. Prentice-Hall, Englewood Cliffs, New Jersey (1971)
2. Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: LAPACK Users' Guide, 3rd edn. Software Environments Tools, SIAM, Philadelphia (1999)
3. Arnold III, W.F., Laub, A.J.: Generalized eigenproblem algorithms and software for algebraic Riccati equations. Proc. IEEE **72**(12), 1746–1754 (1984)
4. Benner, P.: Contributions to the numerical solution of algebraic Riccati equations and related eigenvalue problems. In: Dissertation, Fakultät für Mathematik, Technische Universität Chemnitz-Zwickau, Germany (1997)
5. Benner, P., Byers, R.: An exact line search method for solving generalized continuous-time algebraic Riccati equations. IEEE Trans. Automat. Contr. **43**(1), 101–107 (1998)
6. Benner, P., Byers, R., Losse, P., Mehrmann, V., Xu, H.: Numerical Solution of Real Skew-Hamiltonian/Hamiltonian Eigenproblems. In: Technical report, Technische Universität Chemnitz (2007)
7. Benner, P., Byers, R., Mehrmann, V., Xu, H.: Numerical computation of deflating subspaces of skew Hamiltonian/Hamiltonian pencils. SIAM J. Matrix Anal. Appl. **24**(1), 165–190 (2002)
8. Benner, P., Kressner, D., Sima, V., Varga, A.: Die SLICOT-Toolboxen für Matlab. at–Automatisierungstechnik **58**(1), 15–25 (2010)
9. Benner, P., Mehrmann, V., Sima, V., Van Huffel, S., Varga, A.: SLICOT—a subroutine library in systems and control theory. In: Datta, B.N. (ed.) Applied and Computational Control, Signals, and Circuits, vol. 1, pp. 499–539. Birkhäuser, Boston (1999)
10. Benner, P., Sima, V.: Solving Algebraic Riccati Equations with SLICOT. In: 11th Mediterranean Conference on Control and Automation MED'03, 18–20 June 2003 Rhodes, Greece (2003)
11. Byers, R.: Solving the algebraic Riccati equation with the matrix sign function. Lin. Alg. Appl. **85**(1), 267–279 (1987)
12. Ciubotaru, B., Staroswiecki, M.: Comparative Study of Matrix Riccati Equation Solvers for Parametric Faults Accommodation. In: 10th European Control Conference, Budapest, Hungary, pp. 1371–1376 (2009)
13. Francis, B.A.: A Course in H_∞ Control Theory. In: Thoma, M., Wyner, A. (eds.) LNCIS, vol. 88. Springer-Verlag, Berlin (1987)
14. Gardiner, J.D., Laub, A.J.: A generalization of the matrix sign function solution for algebraic Riccati equations. Int. J. Control **44**, 823–832 (1986)
15. Hammarling, S.J.: Newton's Method for Solving the Algebraic Riccati Equation. In: Technical report DIIC 12/82, National Physics Laboratory, Teddington, U.K. (1982)
16. Jónsson, G.F., Vavasis, S.: Solving polynomials with small leading coefficients. SIAM J. Matrix Anal. Appl. **26**(2), 400–414 (2004)
17. Kleinman, D.L.: On an iterative technique for Riccati equation computations. IEEE Trans. Automat. Contr. AC, **13**, 114–115 (1968)
18. Lancaster, P., Rodman, L.: The Algebraic Riccati Equation. Oxford University Press, Oxford (1995)
19. Lanzon, A., Feng, Y., Anderson, B.D.O., Rotkowitz, M.: Computing the positive stabilizing solution to algebraic Riccati equations with an indefinite quadratic term via a recursive method. IEEE Trans. Automat. Contr. AC, **50**(10), 2280–2291 (2008)

20. Laub, A.J.: A Schur method for solving algebraic Riccati equations. IEEE Trans. Automat. Contr. AC, **24**(6), 913–921 (1979)
21. Leibfritz, F., Lipinski, W.: Description of the Benchmark Examples in COMPlib. In: Technical report, Department of Mathematics, University of Trier, Germany (2003)
22. MathWorks: Control System Toolbox$^{\text{TM}}$ User's Guide. Version 9.2 (Release 2011b). The Math Works, Inc. 3 Apple Hill Drive Natick, MA 01760-2098. http://www.mathworks.com
23. Mehrmann, V.: The Autonomous Linear Quadratic Control Problem. Theory and Numerical Solution. In: Thoma, M., Wyner, A. (eds.) LNCIS, vol. 163. Springer-Verlag, Berlin (1991)
24. Mehrmann, V., Tan, E.: Defect correction methods for the solution of algebraic Riccati equations. IEEE Trans. Automat. Contr. AC, **33**(7), 695–698 (1988)
25. Penzl, T.: Numerical solution of generalized Lyapunov equations. Adv. Comp. Math. **8**, 33–48 (1998)
26. Penzl, T.: LYAPACK Users Guide. In: Technical report SFB393/00-33, Technische Universität Chemnitz, Germany (2000)
27. Raines III, A.C., Watkins, D.S.: A Class of Hamiltonian-Symplectic Methods for Solving the Algebraic Riccati Equation. In: Technical report, Washington State University, Pullman (1992)
28. Roberts, J.: Linear model reduction and solution of the algebraic Riccati equation by the use of the sign function. Int. J. Control **32**, 667–687 (1980)
29. Sima, V.: An efficient Schur method to solve the stabilizing problem. IEEE Trans. Automat. Contr. AC, **26**(3), 724–725 (1981).
30. Sima, V.: Algorithms for Linear-Quadratic Optimization, Pure and Applied Mathematics: A Series of Monographs and Textbooks, Taft E.J., Nashed Z. (eds.), vol. 200. Marcel Dekker Inc, New York (1996)
31. Sima, V.: Computational Experience in Solving Algebraic Riccati Equations. In: 44th IEEE Conference on Decision and Control and European Control Conference ECC' 05, pp. 7982–7987. Omnipress (2005)
32. Sima, V.: Structure-preserving computation of stable deflating subspaces. In: Kayacan, E. (ed.) 10th IFAC Workshop "Adaptation and Learning in Control and Signal Processing" (ALCOSP 2010), IFAC-PapersOnLine, vol. 10, Part 1, http://www.ifac-papersonline.net/Detailed/46793.html (2010)
33. Sima, V.: Computational experience with structure-preserving Hamiltonian solvers in optimal control. In: Ferrier, J.L., Bernard, A., Gusikhin, O., Madani, K. (eds.) 8th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2011), vol. 1, pp. 91–96. SciTePress–Science and Technology Publications (2011)
34. Sima, V., Benner, P.: A SLICOT implementation of a modified Newton's method for algebraic Riccati equations. In: 14th Mediterranean Conference on Control and Automation MED'06. Omnipress, Ancona, Italy (2006)
35. Sima, V., Benner, P.: Experimental evaluation of new SLICOT solvers for linear matrix equations based on the matrix sign function. In: 2008 IEEE Multi-conference on Systems and Control; 9th IEEE International Symposium on Computer-aided Control Systems Design (CACSD), pp. 601–606. Omnipress (2008)
36. Van Dooren, P.: A generalized eigenvalue approach for solving Riccati equations. SIAM J. Sci. Stat. Comput. **2**(2), 121–135 (1981)
37. Van Huffel, S., Sima, V.: SLICOT and control systems numerical software packages. In: 2002 IEEE International Conference on Control Applications and IEEE International Symposium on Computer Aided Control System Design, CCA/CACSD 2002, pp. 39–44. Omnipress (2002)
38. Van Huffel, S., Sima, V., Varga, A., Hammarling, S., Delebecque, F.: High-performance numerical software for control. IEEE Control Syst. Mag. **24**(1), 60–76 (2004)
39. Varga, A.: A Schur method for Pole assignment. IEEE Trans. Automat. Contr. AC, **26**(2), 517–519 (1981)

Chapter 4
State Feedback Control with ANN Based Load Torque Feedforward for PMSM Fed by 3-Level NPC Inverter with Sinusoidal Output Voltage Waveform

Lech Grzesiak and Tomasz Tarczewski

Abstract The approach presented in this work focuses on the full state feedback algorithm designed to control the angular velocity of the PMSM and to provide true sine wave of the 3-level neutral point clamped inverter output voltages. Artificial neural network based feedforward path was introduced into control system in order to improve dynamic behaviour of the PMSM during load changing and to reduce the effect of load torque changes. It was shown that gains of the designed controller and feedforward path are non-stationary and depends on the angular velocity. The simulation results demonstrate the advantages of the proposed approach with comparison to state feedback control system without feedforward path.

Keywords State feedback controller · Artificial neural network · Load torque feedforward · 3-level neutral point clamped inverter · LC filter · Permanent magnet synchronous motor · Disturbance observer

4.1 Introduction

Artificial neural networks (ANN) have been playing an important role in a motion control systems. Thanks to the universal approximation property, ANNs are successfully used for: friction modeling and compensation [1], dead zone function estimation and compensation [2] as well as adaptive control [3].

L. Grzesiak (✉)
Institute of Control and Industrial Electronics,
Warsaw University of Technology, Warsaw, Poland
e-mail: l.grzesiak@isep.pw.edu.pl

T. Tarczewski
Institute of Physics, Faculty of Physics, Astronomy and Informatics,
Nicolaus Copernicus University, Torun, Poland
e-mail: ttarczewski@fizyka.umk.pl

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_4

The control performance of permanent magnet synchronous motor (PMSM) is influenced by an external load. This performance can be improved with the help of the feedforward compensation [4]. Although, load torque is non-measurable variable in a typical motion system, it can be estimated with the help of the disturbance observer [5]. Proper disturbance compensation by using the feedforward path requires suitable formula depends on control algorithm applied [6].

Performance of PMSM depends also on electromagnetic torque ripple [7]. When 3-level Neutral Point Clamped (NPC) true sine wave inverter with an output LC filter is used, torque ripple can be reduced with comparison to 2-level inverter [8]. Non-linear and non-stationary model of true sine wave inverter with an output LC filter causes, that the state feedback control is an attractive control method thanks to full vector component decoupling [9].

In this work control system with discrete state feedback controller for PMSM fed by true sine 3-level NPC inverter is presented. In order to reduce the effect of load torque changes and to improve the dynamic behaviour of PMSM during load variations, NN based non-stationary feedforward load torque path is introduced into control system. The described control structure can be used in industrial applications where load torque compensation and torque ripple suppression is needed.

A mathematical formula how to calculate an appropriate non-stationary gain values for a load torque feedforward path is depicted. Observed load torque is used as an input signal for the feedforward path. The structure and gain matrices of discrete full state feedback controller were selected in order to:

- control the angular velocity of the PMSM with respect to zero *d*-axis component of the current space vector,
- provide true sine wave of the input motor voltages in steady-state.

The paper is organized in seven sections. The next section briefly introduces the mathematical model of an electromechanical system. Section 4.3 describes discrete state feedback controller with internal input model. In Sect. 4.4, feedforward load torque compensation is introduced. Section 4.5 describes load torque observer. Simulation test results, obtained for designed control system, are reported in Sect. 4.6. Section 4.7 concludes the work. Appendix A contains the basic parameters of control system.

4.2 Mathematical Model of an Electromechanical System

Considered control system consists of: discrete state feedback controller with artificial neural network feedforward path, 3-level NPC inverter with an output LC filter, observer and PMSM. Block diagram of proposed control system was shown in Fig. 4.1.

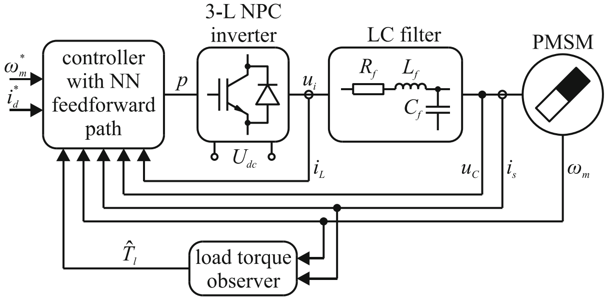


Fig. 4.1 Block diagram of proposed control system

4.2.1 Model of PMSM

In order to create mathematical model of PMSM, following assumptions are made [10, 11]: eddy current and hysteresis losses are negligible, saturation is neglected, the back *emf* is sinusoidal, magnetic symmetry occurs in the circuit. In an orthogonal *d*-*q* coordinate system that rotates at electrical velocity ω_k of the rotor, the expression of the voltage and flux equation takes the following form [10, 11]:

$$u_{Cd} = R_s i_{sd} + \frac{\mathrm{d}\psi_d}{\mathrm{d}t} - p\omega_m \psi_q \tag{4.1}$$

$$u_{Cq} = R_s i_{sq} + \frac{\mathrm{d}\psi_q}{\mathrm{d}t} + p\omega_m \psi_d \tag{4.2}$$

$$\psi_d = L_s i_{sd} + \psi_f \tag{4.3}$$

$$\psi_q = L_s i_{sq} \tag{4.4}$$

where $u_{Cd}, u_{Cq}, i_{sd}, i_{sq}, \Psi_d, \Psi_q$ are space vector components of voltages, currents and fluxes in d and q axis, R_s is resistance of the stator, L_s is inductance of the stator, Ψ_f is permanent magnetic flux linkage, p is the number of pole pairs, ω_m is rotor angular velocity.

Cross couplings between d and q axis as well as the product of an angular velocity and fluxes causes, that voltage Eqs. (4.1)–(4.2) are non-linear.

For a PMSM with a surface mounted magnets, the electromagnetic torque is proportional to the quadrature current and it can be expressed as follows [10, 11]:

$$T_e = \frac{3}{2} p\psi_f i_{sq} = K_t i_{sq} \tag{4.5}$$

where K_t is motor torque constant.

Finally, to complete mathematical model of the PMSM, the following equation of mechanical motion have been added [10, 11]:

$$\frac{\mathrm{d}\omega_m}{\mathrm{d}t} = \frac{1}{J_m}\left(T_e - B_m\omega_m - T_l\right) \tag{4.6}$$

where J_m is motor moment of inertia, T_l is load torque, B_m is viscous friction.

4.2.2 Model of Reactance Filter

Similarly to model of PMSM presented above, model of an output LC filter is described in an orthogonal *d*-*q* coordinate system. The expression of voltage and current equation takes the following form [9]:

$$u_{id} = R_f i_{Ld} + L_f \frac{\mathrm{d}i_{Ld}}{\mathrm{d}t} - L_f \omega_k i_{Lq} + u_{Cd} \tag{4.7}$$

$$u_{iq} = R_f i_{Lq} + L_f \frac{\mathrm{d}i_{Lq}}{\mathrm{d}t} + L_f \omega_k i_{Ld} + u_{Cq} \tag{4.8}$$

$$i_{Cd} = C_f \frac{\mathrm{d}u_{Cd}}{\mathrm{d}t} - C_f \omega_k u_{Cq} \tag{4.9}$$

$$i_{Cq} = C_f \frac{\mathrm{d}u_{Cq}}{\mathrm{d}t} + C_f \omega_k u_{Cd} \tag{4.10}$$

$$i_{Ld} = i_{Cd} + i_{sd} \tag{4.11}$$

$$i_{Lq} = i_{Cq} + i_{sq} \tag{4.12}$$

where u_{id}, u_{iq}, i_{Ld}, i_{Lq} are space vector components of filter input voltages and currents, i_{Cd}, i_{Cq} are space vector components of currents in filter capacitance, R_f is filter resistance, L_f is filter inductance, C_f is filter capacitance.

4.2.3 Model of Inverter

Static model of the 3-level NPC inverter can be used if inverter operates in a linear range, the switching frequency is much higher than the electrical time constant of PMSM and if dead time of IGBTs can be ignored. Model of the inverter can be described as follows [6]:

$$\begin{bmatrix} u_{id} \\ u_{iq} \end{bmatrix} = K_p \begin{bmatrix} u_{pd} \\ u_{pq} \end{bmatrix} \tag{4.13}$$

where u_{pd}, u_{pq} are space vector components of inverter control voltages, K_p is gain coefficient of inverter. Presented in [6] simulation as well as experimental test results show, that described model of the inverter does not introduce any significant error.

4.3 Discrete State Feedback Controller

Non-linear terms in Eqs. (4.1)–(4.2) as well as in Eqs. (4.7)–(4.10) cause that the state feedback control is an attractive approach to control described in a previous section electromechanical system.

4.3.1 State-Space Representation of the System

In order to design state feedback controller, model of electromechanical system (4.1)–(4.13) should be rewritten in a form of the state equation:

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{A}(\omega_k)\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{E}d \tag{4.14}$$

$$\boldsymbol{A}(\omega_k) = \begin{bmatrix} -a_1 & a_2 & -a_3 & 0 & 0 & 0 & 0 \\ -a_2 & -a_1 & 0 & -a_3 & 0 & 0 & 0 \\ a_4 & 0 & 0 & a_2 & -a_4 & 0 & 0 \\ 0 & a_4 & -a_2 & 0 & 0 & -a_4 & 0 \\ 0 & 0 & a_5 & 0 & -a_6 & a_2 & 0 \\ 0 & 0 & 0 & a_5 & -a_2 & -a_6 & -a_7 \\ 0 & 0 & 0 & 0 & 0 & a_8 & -a_9 \end{bmatrix}, \ \boldsymbol{x} = \begin{bmatrix} i_{Ld} \\ i_{Lq} \\ u_{Cd} \\ u_{Cq} \\ i_{sd} \\ i_{sq} \\ \omega_m \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} b_1 & 0 \\ 0 & b_1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$\boldsymbol{u} = \begin{bmatrix} u_{pd} \\ u_{pq} \end{bmatrix}, \ \boldsymbol{E}^{\mathrm{T}} = \begin{bmatrix} 0\,0\,0\,0\,0\,0 - \frac{1}{J_m} \end{bmatrix}, \ d = T_l, \ a_1 = \frac{R_f}{L_f}, \ a_2 = p\omega_m = \omega_k,$$

$$a_3 = \frac{1}{L_f}, \ a_4 = \frac{1}{C_f}, \ a_5 = \frac{1}{L_s}, \ a_6 = \frac{R_s}{L_s}, \ a_7 = \frac{p\psi_f}{L_s}, a_8 = \frac{K_t}{J_m}, \ a_9 = \frac{B_m}{J_m}, \ b_1 = \frac{K_p}{L_f}$$

4.3.2 An Internal Input Model

In the proposed control algorithm steady-state error of the angular velocity is caused by step variations of the reference velocity and the load torque. It could be eliminated by introducing an internal model of the reference input [6]. Field-oriented control strategy with zero d-axis component of the current space vector is the most popular in PMSM [11]. An internal model of the reference direct current has been added to ensure control strategy described above.

An augmented state equation, after introduction an internal input model and assumption, that external load torque T_l is omitted, takes the following form:

$$\frac{\mathrm{d}\boldsymbol{x}_i}{\mathrm{d}t} = \boldsymbol{A}_i(\omega_k)\boldsymbol{x}_i + \boldsymbol{B}_i\boldsymbol{u} + \boldsymbol{F}_i\boldsymbol{r}_i \tag{4.15}$$

where

$$\boldsymbol{A}_i(\omega_k) = \begin{bmatrix} -a_1 & a_2 & -a_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ -a_2 & -a_1 & 0 & -a_3 & 0 & 0 & 0 & 0 & 0 \\ a_4 & 0 & 0 & a_2 & -a_4 & 0 & 0 & 0 & 0 \\ 0 & a_4 & -a_2 & 0 & 0 & 0 & -a_4 & 0 & 0 \\ 0 & 0 & a_5 & 0 & -a_6 & 0 & a_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_5 & -a_2 & 0 & -a_6 & -a_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_8 & -a_9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \ \boldsymbol{x}_i = \begin{bmatrix} i_{Ld} \\ i_{Lq} \\ u_{Cd} \\ u_{Cq} \\ i_{sd} \\ e_i \\ i_{sq} \\ \omega_m \\ e_\omega \end{bmatrix}, \ \boldsymbol{B}_i = \begin{bmatrix} b_1 & 0 \\ 0 & b_1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$\boldsymbol{r}_i = \begin{bmatrix} i^*_{sd} \\ \omega^*_m \end{bmatrix}, \ \boldsymbol{F}_i^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

New state variable e_i introduced in an augmented state Eq. (4.15) corresponds to the integral of the direct current:

$$e_i(t) = \int_0^t [i_{sd}(\tau) - i^*_{sd}(\tau)]d\tau \tag{4.16}$$

where $i^*{}_{sd}$ is the reference value of the direct current. Similarly, state variable e_ω corresponds to the integral of the angular velocity error:

$$e_\omega(t) = \int_0^t [\omega_m(\tau) - \omega^*_m(\tau)]d\tau \tag{4.17}$$

where ω^*_m is the reference value of the angular velocity.

4.3.3 Non-stationary Discrete Controller

The control law for system described by an augmented state Eq. (4.15) can be computed from the following formula:

$$\boldsymbol{u}(t) = -\boldsymbol{K}(\omega_k)\boldsymbol{x}_i(t) = -\boldsymbol{K}_x(\omega_k)\boldsymbol{x}(t) - \boldsymbol{K}_i(\omega_k)e_i(t) - \boldsymbol{K}_\omega(\omega_k)e_\omega(t) \tag{4.18}$$

where $\boldsymbol{K}(\omega_k), \boldsymbol{K}_x(\omega_k), \boldsymbol{K}_i(\omega_k), \boldsymbol{K}_\omega(\omega_k)$ are non-stationary gain matrices of the state feedback controller.

In order to design discrete state feedback controller suitable to implement in a DSP system, the control law presented above must be rewritten in a discrete form:

$$\boldsymbol{u}(n) = -\boldsymbol{K}(\omega_k)\boldsymbol{x}_{\mathbf{i}}(n) = -\boldsymbol{K}_x(\omega_k)\boldsymbol{x}(n) - \boldsymbol{K}_i(\omega_k)e_i(n) - \boldsymbol{K}_\omega(\omega_k)e_\omega(n) \tag{4.19}$$

where n is an index of the discrete sampling time.

By using the backward Euler integration algorithm, discrete form of the state variables e_i and e_ω were obtained:

$$e_i(n) = e_i(n-1) + T_s[i_{sd}(n) - i_{sd}^*(n)] \tag{4.20}$$

$$e_\omega(n) = e_\omega(n-1) + T_s[\omega_m(n) - \omega_m^*(n)] \tag{4.21}$$

where T_s is the sampling interval.

The discrete linear-quadratic optimization method [12] was used to calculate gain coefficients of the state feedback controller at the operating points defined by the actual value of the angular velocity $\omega_k \in [-942; 942]$ rad/s. The Matlab Control System Toolbox has been used to calculate appropriate matrices of the controller:

$$\boldsymbol{K}_x(\omega_k) = \begin{bmatrix} k_{x1}(\omega_k) & k_{x2}(\omega_k) & k_{x3}(\omega_k) & k_{x4}(\omega_k) & k_{x5}(\omega_k) & k_{x6}(\omega_k) & k_{x7}(\omega_k) \\ k_{x8}(\omega_k) & k_{x9}(\omega_k) & k_{x10}(\omega_k) & k_{x11}(\omega_k) & k_{x12}(\omega_k) & k_{x13}(\omega_k) & k_{x14}(\omega_k) \end{bmatrix} \tag{4.22}$$

$$\boldsymbol{K}_i(\omega_k) = \begin{bmatrix} k_{e1}(\omega_k) \\ k_{e2}(\omega_k) \end{bmatrix}, \quad \boldsymbol{K}_\omega(\omega_k) = \begin{bmatrix} k_{\omega 1}(\omega_k) \\ k_{\omega 2}(\omega_k) \end{bmatrix} \tag{4.23}$$

In order to compute non-stationary gain values of the controller, the following penalty matrices has been assigned:

$$\begin{gathered} \boldsymbol{R}_i = \mathrm{diag}([r_{i1} \quad r_{i2}]), \\ \boldsymbol{Q}_i = \mathrm{diag}([q_{i1}\ q_{i2}\ q_{i3}\ q_{i4}\ q_{i5}\ q_{i6}\ q_{i7}\ q_{i8}\ q_{i9}]) \end{gathered} \tag{4.24}$$

where $r_{i1} = r_{i2} = 3 \times 10^{-1}, q_{i1} = q_{i2} = q_{i3} = q_{i4} = 1 \times 10^{-5}, q_{i5} = 5.7 \times 10^1, q_{i6} = 1 \times 10^7, q_{i7} = 7.6 \times 10^{-1}, q_{i8} = 1 \times 10^{-2}, q_{i9} = 1.64 \times 10^2$. Values of the gain matrices depicted above were selected manually in order to:

- provide zero steady-state angular velocity error for step angular velocity reference change as well as load torque step variations;
- achieve twice the rated current of PMSM (i_{sn} = 5.8 A) during the step change of the reference angular velocity from 0 to 70π rad/s with the rated load torque (T_{ln} = 8.8 Nm).

The assumptions presented above determine the maximum dynamics of the designed control system.

Matlab's *polyfit* and *polyval* commands were used to determine the mathematical functions that approximate dependencies between the controller's gain and the angular velocity.

Based on the simulation test results it was found that: coefficients $k_{x2}(\omega_k)$, $k_{x8}(\omega_k)$, $k_{x10}(\omega_k)$, $k_{x12}(\omega_k)$ and $k_{e2}(\omega_k)$ have the negligible impact of the control process and can be replaced by zeros; coefficients $k_{x1}(\omega_k)$, $k_{x3}(\omega_k)$, $k_{x5}(\omega_k)$, $k_{e1}(\omega_k)$, $k_{x9}(\omega_k)$, $k_{x11}(\omega_k)$, $k_{x13}(\omega_k)$, $k_{x14}(\omega_k)$ and $k_{\omega 2}(\omega_k)$ can be replaced by constant values (independent of the angular velocity). Constant gain coefficients were computed by using *mean* function implemented in the Matlab environment. Coefficients $k_{x4}(\omega_k)$, $k_{x6}(\omega_k)$, $k_{x7}(\omega_k)$, $k_{\omega 1}(\omega_k)$ should be implemented as the following linear functions:

$$k_{x4}(\omega_k) = 7.29 \times 10^{-7}\omega_k \tag{4.25}$$

$$k_{x6}(\omega_k) = 5.51 \times 10^{-5}\omega_k \tag{4.26}$$

$$k_{x7}(\omega_k) = -7.28 \times 10^{-6}\omega_k \tag{4.27}$$

$$k_{\omega 1}(\omega_k) = -6.81 \times 10^{-4}\omega_k \tag{4.28}$$

Finally, gain coefficients of the discrete state feedback controller calculated for the system with parameters given in Table 4.1 (see Appendix 1) and for penalty matrices (24) are as follows:

$$\boldsymbol{K}_x\,(\omega_k) = \begin{bmatrix} 0.13 & 0 & 0.0077 & k_{x4}(\omega_k) & 0.62 & k_{x6}(\omega_k) & k_{x7}(\omega_k) \\ 0 & 0.1 & 0 & 0.004 & 0 & 0.31 & 0.053 \end{bmatrix} \tag{4.29}$$

$$\boldsymbol{K}_i\,(\omega_k) = \begin{bmatrix} 298.76 \\ 0 \end{bmatrix}, \qquad \boldsymbol{K}_\omega\,(\omega_k) = \begin{bmatrix} k_{\omega 1}(\omega_k) \\ 5.71 \end{bmatrix} \tag{4.30}$$

4.4 Feedforward Load Torque Compensation

Dynamic properties of the discrete state feedback controller can be improved by using the disturbance signals [12]. In the designed control system, load torque can be used for a feedforward compensation.

4.4.1 Feedforward Computation

In order to introduce feedforward path, residual model of state Eq. (4.14) should be considered [9, 13]:

$$\frac{\mathrm{d}\tilde{\boldsymbol{x}}}{\mathrm{d}t} = \boldsymbol{A}(\omega_k)\tilde{\boldsymbol{x}} + \boldsymbol{B}\tilde{\boldsymbol{u}} \tag{4.31}$$

where

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_{ss}, \quad \tilde{\boldsymbol{u}} = \boldsymbol{u} - \boldsymbol{u}_{ss} \tag{4.32}$$

are deviations from the steady-state.

It can be seen, that residual model presented above is non-stationary due to the presence of ω_k in the state matrix. It was assumed that disturbance d remains constant for deviations from steady state, so it is not present in residual model (4.31).

The control law for the non-stationary residual model can be formulated as follows:

$$\boldsymbol{u} = -\boldsymbol{K}_x(\omega_k)\boldsymbol{x} + [\boldsymbol{K}_x(\omega_k)\ \boldsymbol{I}] \begin{bmatrix} \boldsymbol{x}_{ss} \\ \boldsymbol{u}_{ss} \end{bmatrix} \tag{4.33}$$

where $\boldsymbol{I}$ denotes an identity matrix with an appropriate dimension. The column vector from the right side of the control law (4.33) can be computed from the following form of the state equation in steady-state:

$$\begin{bmatrix} \boldsymbol{x}_{ss} \\ \boldsymbol{u}_{ss} \end{bmatrix} = -\boldsymbol{G}(\omega_k)^{-1}\boldsymbol{E}d \tag{4.34}$$

where

$$\boldsymbol{G}(\omega_k) = [\boldsymbol{A}(\omega_k)\boldsymbol{B}] \tag{4.35}$$

After substituting of (4.34) into (4.33), the control law can be rearranged as follows:

$$\boldsymbol{u} = -\boldsymbol{K}_x(\omega_k)\boldsymbol{x} - [\boldsymbol{K}_x(\omega_k)\boldsymbol{I}]\,\boldsymbol{G}(\omega_k)^{-1}\boldsymbol{E}d \tag{4.36}$$

Denoting the second component of the Eq. (4.36) as:

$$\boldsymbol{K}_d(\omega_k) = [\boldsymbol{K}_x(\omega_k)\boldsymbol{I}]\boldsymbol{G}(\omega_k)^{-1}\boldsymbol{E} \tag{4.37}$$

one can write the control law with the feedforward path:

$$\boldsymbol{u} = -\boldsymbol{K}_x(\omega_k)\boldsymbol{x} - \boldsymbol{K}_d(\omega_k)d \tag{4.38}$$

Finally, the discrete form of the control law with an internal input model of the reference signals and with the feedforward path takes the following form:

$$\boldsymbol{u}(n) = -\boldsymbol{K}_x(\omega_k)\boldsymbol{x}(n) - \boldsymbol{K}_i(\omega_k)e_i(n) - \boldsymbol{K}_\omega(\omega_k)e_\omega(n) - \boldsymbol{K}_d(\omega_k)d(n) \tag{4.39}$$

After evaluating Eq. (4.37), it was found that the relationships between the angular velocity ω_k and feedforward gain coefficients: $\boldsymbol{K}_d{}^T(\omega_k) = [k_{d1}(\omega_k)\ k_{d2}(\omega_k)]$ are nonlinear (Fig. 4.2).

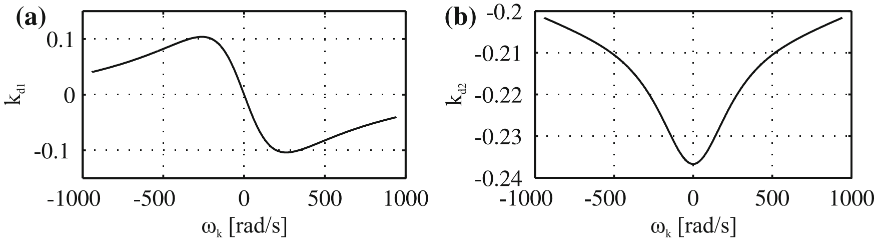


Fig. 4.2 Values of feedforward coefficients

4.4.2 Neural Network Approximation

Since artificial neural networks have an inherent capability of learning and approximating nonlinear functions [1], it is attractive to apply them to approximate nonlinear dependencies presented in Fig. 4.2.

It was found that feedforward coefficients can be successfully approximated with the help of the feedforward backpropagation artificial neural network. For a neural network with 7 neurons in the hidden layer and 2 neurons in the output layer, satisfactory level of approximation (mean square error less than 1×10^{-7}) was achieved after 417 epochs. Schematic diagram of the designed and trained in a Matlab environment neural network approximator is presented in Fig. 4.3, where n—normalization block, w$_i$—weight, b$_i$—bias, d—denormalization block. Sigmoidal functions were used as the activation functions in the hidden layer, while linear functions were used in the output layer.

Schematic diagram of state feedback controller described by (4.39) with artificial neural network based feedforward path depicted above is presented in Fig. 4.4.

4.5 Load Torque Observer

In order to design the control system with a feedforward load torque compensation, a non-measured load torque should be estimated with the help of the observer. The discrete state equation that describes the dynamics of the system takes the following form [5]:

$$\Delta \boldsymbol{x}_o(n) = \boldsymbol{A}_o \boldsymbol{x}_o(n) + \boldsymbol{B}_o u_o(n) \tag{4.40}$$

$$y_o(n) = \boldsymbol{C}_o \boldsymbol{x}_o(n) \tag{4.41}$$

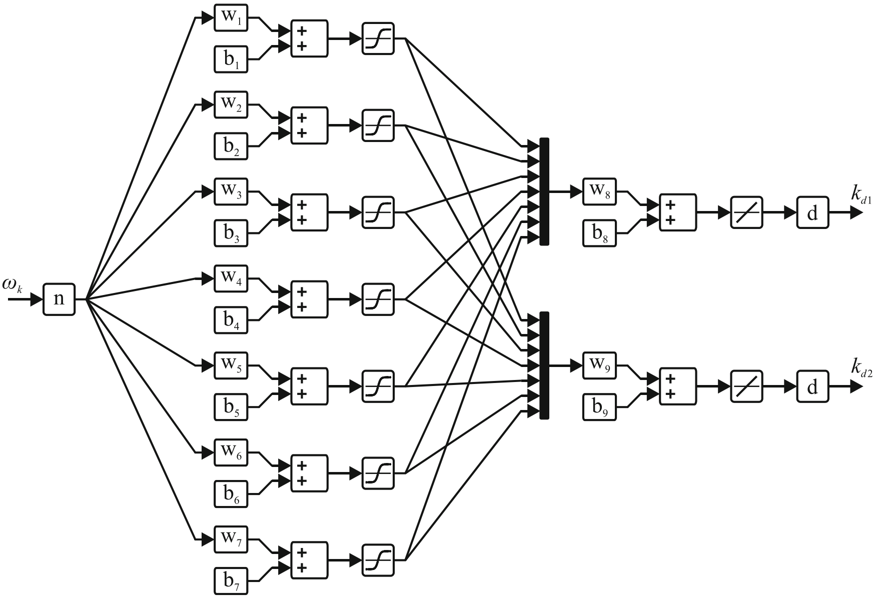


Fig. 4.3 Neural network based approximator

where

$$\Delta \boldsymbol{x}_o(n) = \frac{\boldsymbol{x}_o(n) - \boldsymbol{x}_o(n-1)}{T_s}, \ \boldsymbol{x}_o(n) = \begin{bmatrix} \omega_m(n) \\ T_l(n) \end{bmatrix}, \ \boldsymbol{B}_o = \begin{bmatrix} \frac{K_t}{J_m} \\ 0 \end{bmatrix},$$

$$\boldsymbol{A}_o = \begin{bmatrix} -\frac{B_m}{J_m} & -\frac{1}{J_m} \\ 0 & 0 \end{bmatrix}, \quad u_o(n) = i_{sq}(n), \ \boldsymbol{C}_o = [\, 1 \ 0 \,] \tag{4.42}$$

For system (4.42)–(4.42) the following equation of the discrete load torque observer can be formulated [14]:

$$\Delta\hat{\boldsymbol{x}}_o(n) = \boldsymbol{A}_o\hat{\boldsymbol{x}}_o(n) + \boldsymbol{B}_o u_o(n) + \boldsymbol{L}[y_o(n) - \boldsymbol{C}_o\hat{\boldsymbol{x}}_o(n)] \tag{4.43}$$

where

$$\hat{\boldsymbol{x}}_o(n) = \begin{bmatrix} \hat{\omega}_m(n) \\ \hat{T}_l(n) \end{bmatrix}, \quad \boldsymbol{L} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} \tag{4.44}$$

An observable values are depicted in $\hat{\boldsymbol{x}}_o$ while $\boldsymbol{L}$ is a gain matrix of the designed observer. Schematic diagram of implemented in Simulink discrete load torque observer is shown in Fig. 4.5.

The goal of the designed load torque observer is to provide an estimate $\hat{\boldsymbol{x}}_o$ so that $\hat{\boldsymbol{x}}_o \to \boldsymbol{x}_o$ for $t \to \infty$. Because system (4.43) is fully observable, we can find

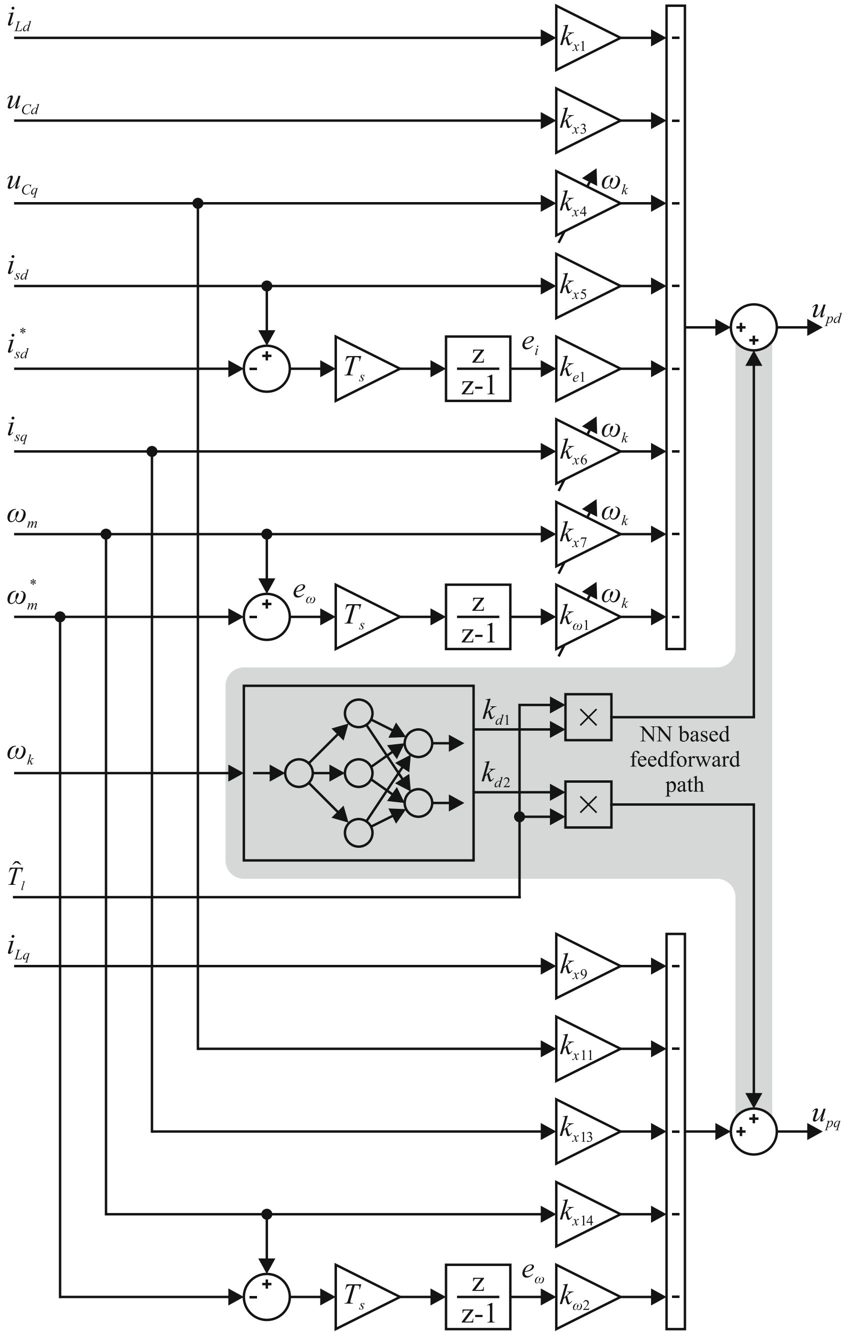


Fig. 4.4 Block diagram of the designed state feedback controller with NN based feedforward

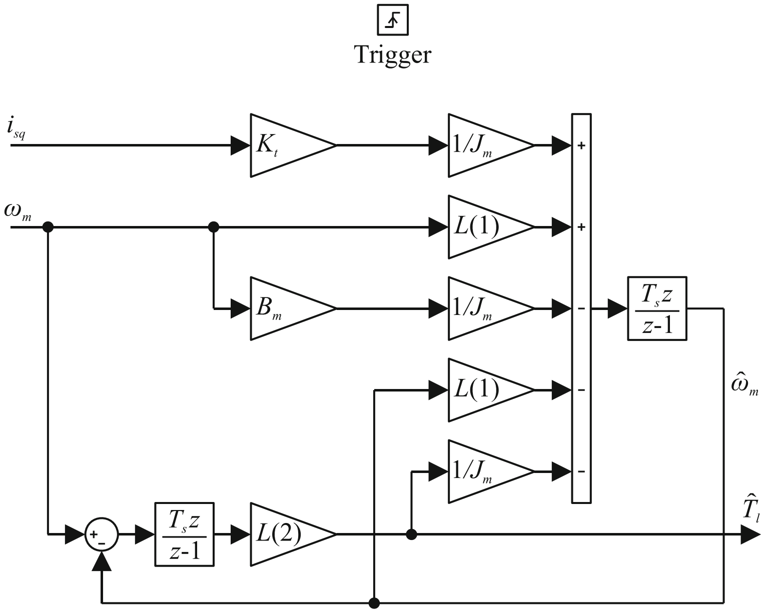


Fig. 4.5 Block diagram of the load torque observer

L matrix so that the tracking error is asymptotically stable. Therefore, the observer design process is reduced to finding the gain matrix ***L*** so that the roots of the system (4.43) characteristic equation lie in the left half-plane. Gain matrix of the load torque observer was determined with the help of Matlab's *place* formula. For the pole locations:

$$o_{1/2} = -3 \times 10^3 \pm 1 \times 10^3 i \tag{4.45}$$

that guarantee the proper dynamics of the observer, values of the gain matrix ***L*** are as follows:

$$l_1 = 6 \times 10^3, \quad l_2 = -6.2 \times 10^3. \tag{4.46}$$

4.6 Control System with Discrete State Feedback Controller and Load Torque Feedforward

The proposed control system was tested in the Matlab/Simulink environment with the help of the Plecs blockset. The results obtained for control system with neural network based load torque feedforward path were compared with the results achieved for the state feedback based control system without feedforward.

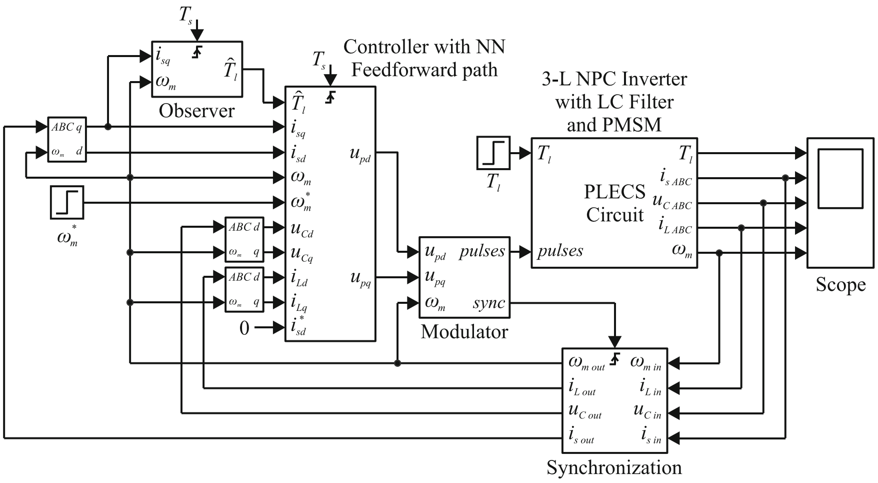


Fig. 4.6 Schematic diagram of the designed control system

4.6.1 Model of Proposed Control System

Schematic diagram of the designed control system was presented in Fig. 4.6.

Described in previous sections discrete state feedback controller as well as load torque observer were implemented in triggered subsystems in order to ensure proper generation of discrete control and estimate signals respectively. The sampling interval was set to T_s = 100 μs (the switching frequency is equal to f_s = 10 kHz). In order to realize measurements in a midpoint of the PWM pulse length, triggered synchronization block was used. Carrier-based sinusoidal PWM with level shifted triangular carriers modulation method was used to control switches in the 3-level NPC inverter [15]. Shown in Fig. 4.7 model of PMSM with 3-level NPC inverter as well as LC filter was implemented in the Plecs software.

For proper operation of the designed control system the resonance frequency of the LC filter (f_r = 1453 Hz) was set to be almost ten times higher than the rated frequency of the motor (f_m = 150 Hz) and almost seven times lower than the switching frequency [16].

4.6.2 Simulation Test Results

Simulation test results of the proposed control system were presented in Fig. 4.8.

Depicted in Fig. 4.8a the angular velocity step responses of the control system show, that by using state feedback controller with load torque feedforward path, improvement of the dynamics could be achieved during the transient caused by the load torque step change. It can be seen, that the angular velocity error caused by load

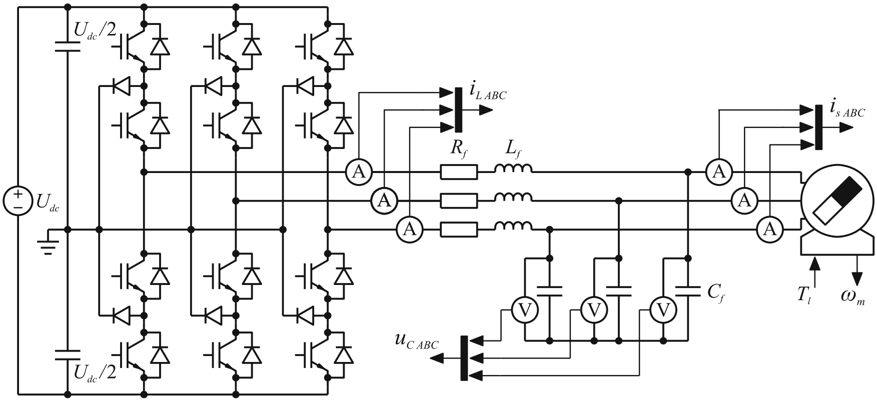


Fig. 4.7 Schematic diagram of the PMSM with 3-level NPC inverter and LC filter

torque step changes at $t = 20$ ms and at $t = 150$ ms is smaller, when feedforward path is used. The use of the load torque feedforward path minimize the dynamic error by the transient.

The proper operation of the load torque observer is presented in Fig. 4.8c. An actual value of the load torque is estimated with good dynamics and without steady-state error.

It can be seen from Fig. 4.8d, that the q-axis component of the current space vector is responsible for producing electromagnetic torque. PMSM operates with control strategy based on zero d-axis component of the current space vector.

By using of the LC filter, sinusoidal waveform of the input motor voltage can be obtained in a steady-state (Fig. 4.8f). In this case, electromagnetic torque ripple reduction can be achieved.

4.7 Conclusions

This paper presents discrete full state feedback non-stationary controller with neural network based non-stationary load torque feedforward path. A mathematical formula how to calculate an appropriate non-stationary gain values for a feedforward was presented.

Non-stationary state feedback control algorithm was used because model of the controlled plant (i.e. PMSM fed from 3-level NPC type inverter with output LC filter) is non-stationary and non-linear. Use of non-stationary controller causes, that linearization and decoupling process of the plant is not needed.

Discrete state feedback non-stationary controller was designed in order to control the angular velocity of the PMSM and to provide control strategy based on zero d-axis component of the current space vector as well as sinusoidal waveforms of the

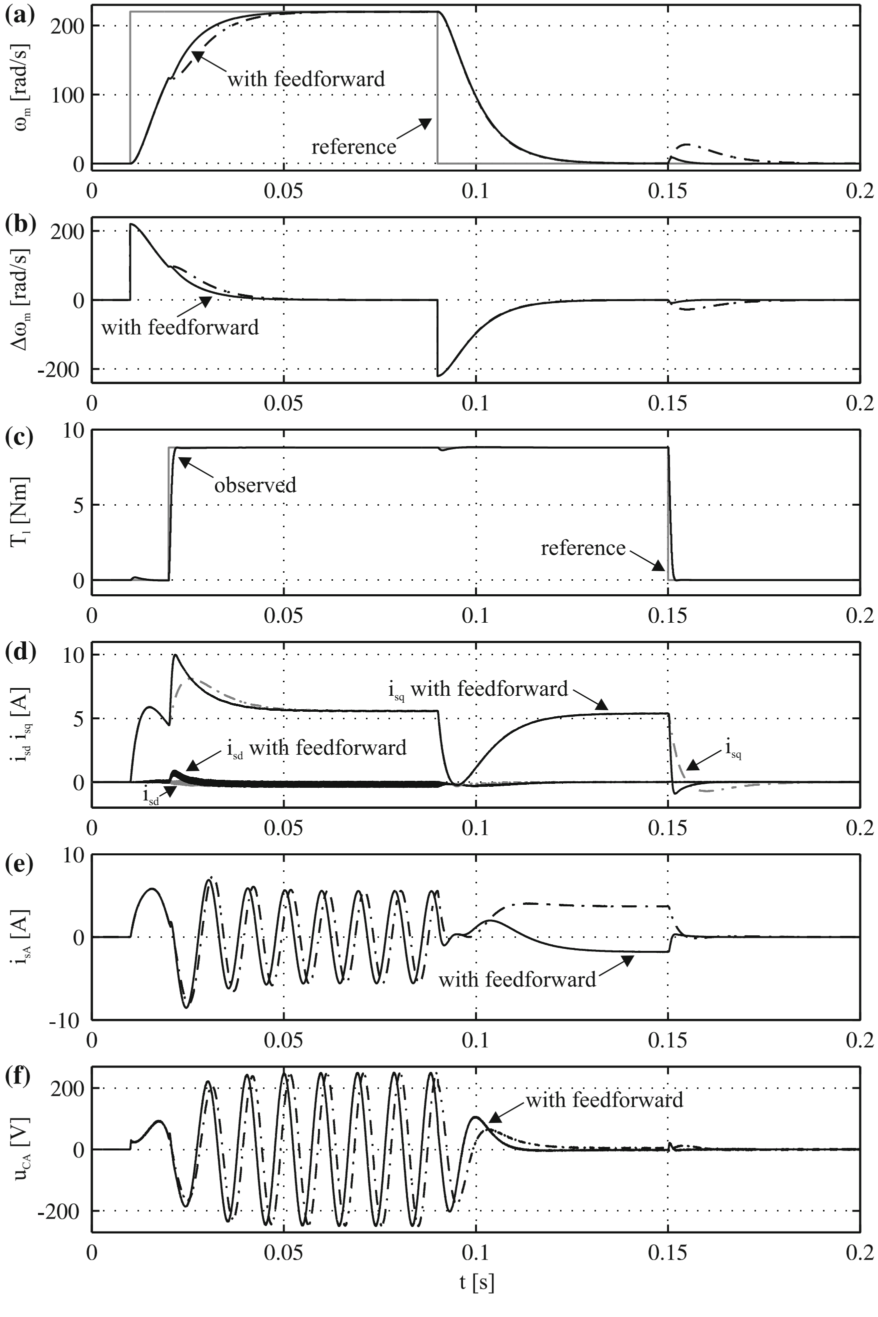


Fig. 4.8 Simulation test results

input motor voltages. Field-oriented control strategy with $i_{sd}{}^{*} = 0$ was realized by introduction of an internal input model into controller structure as well as by proper selection of the controller gain matrices.

Designed neural network approximator was successfully implemented in a control system with PMSM fed by 3-level NPC inverter with output LC filter. The observed load torque has been used as an input signal for the feedforward path. Proposed feedforward path significantly improves dynamic properties of the considered control system during load torque changing. The described control structure can be used in industrial applications where load torque compensation and torque ripple suppression is needed.

The proposed control algorithm was successfully tested in a Matlab environment. Experimental verification of the designed control algorithm with NN feedforward path is planned in the future.

4.8 Appendix: The Basic Parameters of the Control System

Table 4.1 The basic parameters of the control system

Parameter	Value	Unit
R_f	3×10^{-2}	Ω
L_f	2×10^{-3}	H
C_f	6×10^{-6}	F
R_s	1.05	Ω
L_s	9.5×10^{-3}	H
K_t	1.635	Nm/A
J_m	6.2×10^{-4}	kg m^2
B_m	1.4×10^{-3}	Nms/rad
K_p	291	
p	3	

Acknowledgments Research work financed by The National Science Centre (Poland) under Grant no 6636/B/T02/2011/40 (from 2011 to 2013).

References

1. Huang, S., Tan, K.K.: Intelligent friction modeling and compensation using neural network approximations. IEEE Trans. Ind. Electron. **59**(8), 3342–3349 (2012)
2. Selmic, R.R., Lewis, F.L.: Deadzone compensation in motion control systems using neural networks. IEEE Trans. Autom. Control **45**(4), 602–613 (2000)

3. Pajchrowski, T., Zawirski, K.: Adaptive neural speed controller for PMSM servodrive with variable parameters. In: 15th International Power Electronics and Motion Control Conference, pp. LS6b.3-1-LS6b.3-5, IEEE Press, New York (2012)
4. Iwasaki, M., Seki, K., Maeda, Y.: High-precision motion control techniques: a promising approach to improving motion performance. IEEE Ind. Electron. Mag. **6**(1), 32–40 (2012)
5. Mun-Soo, K., Song, D.-S., Lee, Y.-K., Won, T.-H., Park, H.-W., Jung, Y.-I., Lee, M.H., Lee, H.: A robust control of permanent magnet synchronous motor using load torque estimation. In: International Symposium on Industrial Electronics, vol. 2, pp. 1157–1162. IEEE Press, New York (2001)
6. Grzesiak, L.M., Tarczewski T.: PMSM servo-drive control system with a state feedback and a load torque feedforward compensation, COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, vol. 32, iss. 1, pp. 364–382, (2013).
7. Gulez, K., Adam, A.A., Pastaci, H.: Torque ripple and EMI noise minimization in PMSM using active filter topology and field-oriented control. IEEE Trans. Ind. Electron. **55**(1), 251–257 (2008)
8. Tarczewski, T., Grzesiak, L.M.: PMSM fed by 3-level NPC sinusoidal inverter with discrete state feedback controller. In: 15th European Conference on Power Electronics and Applications, pp. 1–9, IEEE Press, New York (2013)
9. Pawlikowski, A., Grzesiak, L.: Vector-controlled three-phase voltage source inverter producing a sinusoidal voltage for AC motor drives. In: The International Conference on "Computer as a Tool", pp. 1902–1909 (2007)
10. Pillay, P., Krishnan, R.: Modeling of permanent magnet motor drives. IEEE Trans. Ind. Electron. **35**(4), 537–541 (1988)
11. Zawirski, K.: Control of Permanent Magnet Synchronous Motor. Poznan University of Technology Publishers, Poznan (2005). (in Polish)
12. Tewari, A.: Modern Control Design with Matlab and Simulink. Wiley, Chichester (2002)
13. Lee, D.-C., Sul, S.-K., Park, M.-H.: High performance current regulator for a field-oriented controlled induction motor drive. IEEE Trans. Ind. Appl. **30**(5), 1247–1257 (1994)
14. Luenberger, D.: An introduction to observers. IEEE Trans. Autom. Control **16**(6), 596–602 (1971)
15. Rodriguez, J., Bernet, S., Steimer, P.K., Lizama, I.E.: A survey on neutral-point-clamped inverters. IEEE Trans. Ind. Electron. **57**(7), 2219–2230 (2010)
16. Steinke, J.K.: Use of an LC filter to achieve a motor-friendly performance of the PWM voltage source inverter. IEEE Trans. Energy Convers. **14**(3), 649–654 (1999)

Chapter 5
Adaptive Dynamic Programming-Based Control of an Ankle Joint Prosthesis

Anh Mai and Sesh Commuri

Abstract The potential of an adaptive dynamic programming (ADP)-based control strategy for learning the human gait dynamics in real-time and generating control torque for a prosthetic ankle joint is investigated in this paper. This is motivated by the desire for control strategies which can adapt in real-time to any gait variations in a noisy environment while optimizing some gait related performance indices. The overall amputated leg–prosthetic foot system is represented by a link-segment model with the kinematic patterns for the model are derived from human gait data. Then a learning-based control strategy including an ADP-based controller and augmented learning rules is implemented to generate torque which drives the prosthetic ankle joint along the designed kinematic patterns. Numerical results show that with the proposed learning rules, the ADP-based controller is able to maintain stable gait with robust tracking and reduced performance indices in spite of measurement/actuator noises and variations in walking speed. Promising results achieved in this paper serve as the starting point for the development of intelligent ankle prostheses, which is a challenge due to the lack of adequate mathematical models, the variations in the gait in response to the walking terrain, sensor noises and actuator noises, and unknown intent of users.

Keywords Adaptive dynamic programming · Biomechanics · Ankle joint prosthesis · Optimization · Learning

5.1 Introduction

Current ankle/foot prostheses are primarily passive devices whose performance cannot be adapted or optimized to meet the requirements of different users. Further, such devices cannot provide the rigidity, as well as the flexibility and power similar to that

A. Mai (✉) · S. Commuri
School of Electrical and Computer Engineering, The University of Oklahoma,
110 W. Boyd St., Devon Energy Hall 150, Norman, OK 73019-1102, USA
e-mail: anhmai@ou.edu

S. Commuri
e-mail: scommuri@ou.edu

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_5

of a human foot. The adverse consequences of wearing less functioning prosthetic feet include asymmetric gait, increased metabolic consumption, limited blood flow, instability, and pain. In the long-term, the amputees, especially ones with diabetes, might have to undergo hip replacement procedure and use wheel-chair on a daily basis.

The lack of an active prosthetic joint that can dynamically adapt to changing terrain and gait needs is a limiting factor in attaining adequate comfort and mobility in below-knee amputees. Powered ankle prostheses can adapt to some extent, but the rigidity and power required during the gait are usually varying depending on the activity pursued by the individual. Such unknown, varying requirement cannot be addressed through standard control techniques. One of the key steps in the development of these active prosthetic feet is the generation of adaptive torque profiles to drive the ankle joint in response to variations in the human locomotion. The design should also provide necessary energy return to significantly reduce the metabolic energy consumption during locomotion [1]. In an effort to achieve these goals, bionic feet such as Proprio Foot [2], BiOM [3], SPARKy [4], PPAMs [5] have been equipped with active components that can modify the dynamic characteristics of the prosthetic ankle joints. It is noted that the ankle joints currently available are typically controlled using classical control techniques. Once the controller is tuned, its parameters are usually fixed irrespective of any changes in gait. Adaptive control strategies can account for changes in gait. However, such adaptive strategies have to overcome the challenges due to lack of information on gait and interaction between the foot and the ground as well as the interaction between the prosthetic socket and the residual limb. In the absence of such information, optimization of the performance of the controller becomes a very challenging task and requires the use of new design strategies such as learning-based control.

Mathematical models and experimental data can be effectively combined to study normal and pathological gaits [6–8]. Figure 5.1 shows the diagram of the control-based approach which concentrates on generating suitable control signals to drive the model dynamics along desired trajectories obtained from the analysis of human gait [9]. In this framework, different methods of generating the joint torque can be analytically evaluated and the overall performance can be improved by feedback modification. Similarly, simulation frameworks which combine mathematical gait models and experimental data can be used to study the effect of prostheses on kinematic behaviors and other aspects of amputee locomotion [10, 11]. Such frameworks enable a quick evaluation of the performance of prosthetic devices under different operating conditions and extend the understanding of the prosthetic ankle-foot systems [12]. However, due to the complex interaction between the prosthetic feet and the ground and the unknown intent of amputees, it is not easy to guarantee efficient gait or robustness in performance. Therefore, a suitable control strategy that permits online adaptation to variations in gait while guaranteeing robust performance and improved efficiency has to be developed.

In this paper, performance and potential of an adaptive dynamic programming-based control structure, named Direct Neural Dynamic Programming (DNDP) [13], for control of an active prosthetic ankle joint is evaluated. DNDP has been shown

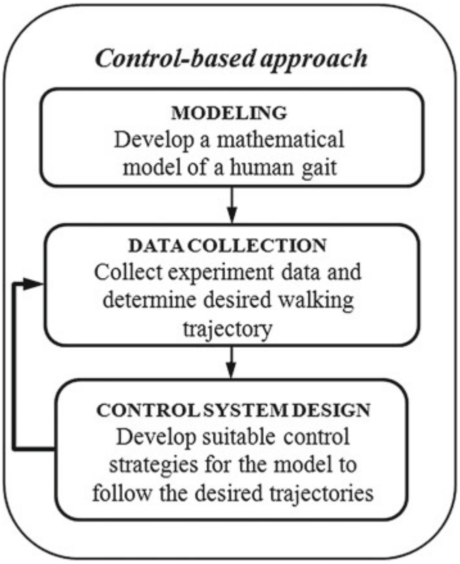

Control-based approach
MODELING
Develop a mathematical model of a human gait
DATA COLLECTION
Collect experiment data and determine desired walking trajectory
CONTROL SYSTEM DESIGN
Develop suitable control strategies for the model to follow the desired trajectories

Fig. 5.1 Control-based approach to the modeling and control of human gait

to be suitable for control of complex nonlinear systems with unknown dynamics and disturbances [14, 15]. Furthermore, this approach also tries to minimize the long-term cost function in the sense of Bellman‘s principle of optimality (dynamic programming). With these properties, DNDP appears to be a good candidate for a challenging task such as control of a prosthetic ankle. In order to apply this control technique, this paper addresses issues such as gait dynamics formulation, desired behaviors of the ankle joint during gait, control strategies, and long-term gait-related performance indices. In addition, augmented training rules are proposed to provide robustness against the external disturbances. This is the first attempt in applying such real-time adaptation scheme in learning the gait parameters and adjusting the control output to improve the gait efficiency. This will have enormous impact on the quality of life as well as the long-term health of people with below-knee amputation.

The rest of this paper is organized as follows. Section 5.2 describes the link-segment representation of the combined amputated leg—prosthetic foot system, viscoelastic model of the ground-foot interaction, and dynamics of the prosthetic ankle joint during gait. Section 5.3 presents the structure, learning algorithms, and implementation issues of the adaptive dynamic programming-based controller for the prosthetic ankle joint. Section 5.4 describes the numerical study for evaluation of the performance of the prosthetic ankle control structure. Section 5.5 concludes the paper and outlines future work.

5.2 Dynamical Models of the Gait

5.2.1 Link-Segment Representation of the Gait

The dynamic model in the sagittal plane of the residual limb of a unilateral below-knee amputee is considered in this study. This link-segment model includes 3 revolute joints: the hip joint connecting the biological thigh with the upper part of the human body; the knee joint connecting the biological thigh with the residual limb/artificial shank, and the prosthetic ankle joint connecting the artificial shank with the prosthetic foot. Actions of the human muscles and ligaments that control the hip and knee joints are represented by the torques at those biological joints. At the prosthetic ankle joint, an externally powered actuator generates a torque to manipulate the angular position of the ankle.

The kinematic and dynamic relationship of the link-segment model in Fig. 5.2 is obtained using the Euler-Lagrange formulation [16] following assumptions similar to those in Sects. 5.0.1 and 8.0.1 of [17]. The interaction between the residual limb and the socket to which the prosthetic foot is connected is not considered and the residual-biological-artificial shank is assumed to be rigid. The equations that govern the dynamics of the overall human-prosthetic system can be expressed as follows:

$$M\left(\theta\right)\ddot{\theta} + V\left(\theta,\dot{\theta}\right)\dot{\theta} + G\left(\theta\right) + F\left(\theta\right)\ddot{a}_H = \tau + DF_{GRF}, \tag{5.1}$$

where $\theta = \left[\theta_1\ \theta_2\ \theta_3\right]^T$ are joint angles (rad), $\dot{\theta} = \left[\dot{\theta}_1\ \dot{\theta}_2\ \dot{\theta}_3\right]^T$ are joint angular velocities (rad/s), and $\ddot{\theta} = \left[\ddot{\theta}_1\ \ddot{\theta}_2\ \ddot{\theta}_3\right]^T$ are joint angular accelerations (rad/s^2); $\ddot{a}_H = \left[\ddot{x}_H\ \ddot{z}_H\right]^T$ are the hip acceleration (m/s^2), $\tau = \left[\tau_1\ \tau_2\ \tau_3\right]^T$ are components of joint torques (Nm), and $F_{GRF} = \left[F_X\ F_Z\right]^T$ are horizontal and vertical components

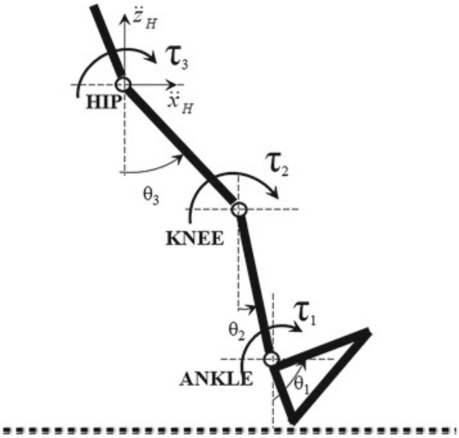


Fig. 5.2 Link-segment representation of the residual limb with a prosthetic ankle joint

of the ground reaction force (N). The nonlinear terms in (5.1) include the inertia matrix $M(\theta)$, the Coriolis and Centripetal term $V\left(\theta, \dot{\theta}\right)$, the gravity term $G(\theta)$, the coefficient matrix $F(\theta)$ that relates to the hip joint acceleration, and the Jacobian matrix $D(.)$ that transfers the effect of the ground reaction force onto the dynamics of each joint. Among these components, the ground reaction forces play a very important role and will be described in the subsequent section.

5.2.2 Ground Reaction Force

According to Winter [17], there are three forces acting on the link-segment model of the human gait: gravitational force, ground reaction force, and muscle and ligament forces. In the depicted gait model, the gravitational force is represented by the nonlinear term $G(\theta)$ whereas the force generated by the muscles and ligaments are replaced by the torque applied at the biological hip and knee joints. The ground reaction forces are generated during the gait as the result of interaction between the foot and the ground. Such reaction forces are then transferred up to the ankle, knee, and hip joints with the effect of altering the joint angular positions. Because the interaction between the foot and the ground is very complicated, it is very hard, if not impossible to exactly measure the ground reaction force without using carefully designed gait lab and force transducers [17]. On the other hand, the ground reaction force (GRF) cannot be ignored during the study of the human gait [7, 18]. Therefore, the following widely used model [6, 7] is selected as:

$$F_Z = k(z_{PEN})^e + Step\,(y, 0, 0, d_{\max}, c_{\max})\,\dot{z}_{PEN} \tag{5.2}$$

$$F_X = \mu F_Z sgn\,(\dot{x}_{COP})\,. \tag{5.3}$$

In this GRF model, F_Z and F_X are vertical and horizontal force components (N);$z_{PEN}, \dot{z}_{PEN}$ are the penetration (m) and penetration rate (m/s); k, e are spring coefficient (N/m) and spring exponent; $c_{\max}$ is the maximal damping coefficient (N/(m/s)); $d_{\max}$ is the maximal damping penetration (mm); μ is the friction coefficient; and $\dot{x}_{COP}$ is the horizontal velocity of the contact point with respect to the ground (m/s). Detailed descriptions of the parameters of this model can be found in [7].

It is noted that this ground reaction force model is more realistic than the rigid contact approach because it can describe the viscous-elastic behavior of the foot-ground interaction [19]. The penetration of the foot into the ground is modified from [20]. Because the ground reaction force can neither be measured exactly nor be ignored, it is treated as external disturbance to the gait dynamics during the evaluation of the control strategy.

5.2.3 Dynamics of the Prosthetic Ankle Joint During Gait

From the dynamics of the overall amputated leg—prosthetic foot system, dynamics of the prosthetic ankle joint during gait can be described as:

$$M_{11}\ddot{\theta}_1 + V_{11}\dot{\theta}_1 + G_1 + M_{12}\ddot{\theta}_2 + M_{13}\ddot{\theta}_3 + V_{12}\dot{\theta}_2 + V_{13}\dot{\theta}_3 + F_{11}\ddot{x}_H + F_{12}\ddot{z}_H \\ = \tau_1 + D_{11}F_X + D_{12}F_Z, \tag{5.4}$$

Terms in (5.4) represent dynamics of the prosthetic ankle joint as well as the interaction between the ankle joint and the biological knee joint, the biological hip joint, and the walking surface during gait. In the next section, a learning-based control strategy will be occupied to generate a torque τ_1 which compensates for these interactions and guarantees tracking of desired joint trajectories.

5.3 Control Structure of the Prosthetic Ankle Joint

5.3.1 Control of the Ankle Joint

The angular position of the prosthetic ankle joint can be controlled by an external actuator. This study focuses on a learning-based control strategy and training algorithms, therefore the dynamics of the actuator are not considered. The actuator at the prosthetic ankle joint is assumed to have access to only the actual ankle angle and angular velocity. Such quantities could be measured by using a rotational encoder and gyroscope mounted on the prosthetic foot. Therefore, the torque produced by an external actuator could be a function of the ankle angle, the ankle angular velocity, and the tracking errors between these quantities and their desired kinematic patterns as follows:

$$\tau_1 = f\left(\theta_1, \dot{\theta}_1, e_1, \dot{e}_1\right), \tag{5.5}$$

$$e_1 = (\theta_{1r} - \theta_1) \tag{5.6}$$

$$\dot{e}_1 = \left(\dot{\theta}_{1r} - \dot{\theta}_1\right) \tag{5.7}$$

where e_1 and $\dot{e}_1$ are tracking errors of the ankle angle and ankle angular velocity; $\theta_{1\mathrm{r}}$ and $\dot{\theta}_{1\mathrm{r}}$ are desired trajectories of the ankle angle and ankle angular velocity.

In order to reduce the system order, the filtered tracking error is defined as:

$$r_1 = \dot{e}_1 + \lambda_1 e_1, \tag{5.8}$$

in which $\lambda_1 > 0$ is the design parameter [21].

Then the dynamics of the ankle joint (5.4) can be written in term of the filtered tracking error as follows:

$$M_{11}\dot{r}_1 = -V_{11}r_1 + f_1(x) - \tau_1, \tag{5.9}$$

with the nonlinear term $f_1(x)$ is defined as:

$$\begin{aligned} f_1(x) = {} & M_{11}\left(\ddot{\theta}_{1r} + \lambda_1\dot{e}_1\right) + V_{11}\left(\dot{\theta}_{1r} + \lambda_1 e_1\right) + M_{12}\ddot{\theta}_2 + M_{13}\ddot{\theta}_3 \\ & + V_{12}\dot{\theta}_2 + V_{13}\dot{\theta}_3 + G_1 + F_{11}\ddot{x}_H + F_{12}\ddot{z}_H - D_{11}F_X - D_{12}F_Z. \end{aligned} \tag{5.10}$$

This nonlinear function, especially with the disturbances caused by the hip joint acceleration $(F_{11}\ddot{x}_H + F_{12}\ddot{z}_H)$ and the ground reaction torque $(D_{11}F_X + D_{12}F_Z)$, is unknown and difficult to compute. The nonlinearity of this function is further increased in multi-step gait due to the fact that the ground reaction torque only affects the gait dynamics during the stance phases when the residual limb is in contact with the ground. However, this external torque is not present during the swing phases. In order to overcome these difficulties, this nonlinear function will be learnt online and approximated by a multi-layer neural network in an adaptive dynamic programming framework which will be described in the next section.

Finally, assumed that the nonlinear function $f_1(x)$ is approximated by $\hat{f}_1(x)$, the control signal is generated as:

$$\tau_1 = \hat{f}_1(x) + K_{v1}r_1, \tag{5.11}$$

with $K_{v1}r_1$ is a Proportional-Derivative (PD) control term, and r_1 is the filtered tracking error defined in (5.8).

5.3.2 DNDP-Based Control Structure

The DNDP-based control structure comprises of two neural networks: critic network and action network. The critic network is responsible for approximating of the long-term cost function which satisfies the Bellman's principle of optimality. The action network is responsible for generating a control signal which leads to the optimization of the approximated long-term cost (or output of the critic network). Figure 5.3 presents the two-network configuration of the DNDP-based control. The next section will provide detailed information about elements in Fig. 5.3.

Critic Network. In general adaptive dynamic programming framework, the long-term cost is represented as the weighted sum of the short-term (instantaneous) cost as follows:

$$L(t) = S(t+1) + \alpha S(t+2) + \alpha^2 S(t+3) + \cdots = S(t+1) + \alpha L(t+1), \tag{5.12}$$

with α is the discount factor.

Because the critic network generates $J(t)$ as an approximation of the long-term cost function $L(t)$, the backpropagation error is defined as:

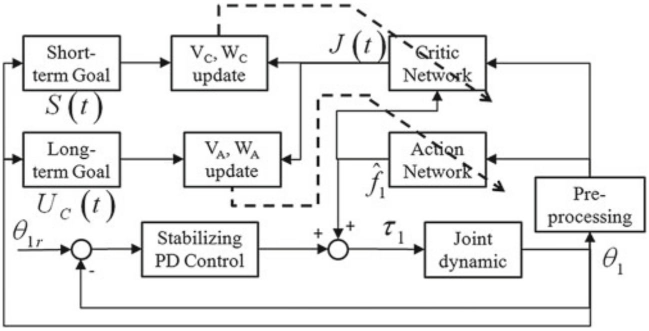


Fig. 5.3 DNDP-based control of the prosthetic ankle joint

$$e_C(t) = \underbrace{[J(t-1) - S(t)]}_{TARGET} - \underbrace{\alpha J(t)}_{\substack{CURRENT\\OUTCOME}}, \tag{5.13}$$

where $S(t)$ is the instantaneous cost at time t (short-term cost).

Inputs to the critic network are:

$$x_C = \left[e_1 \; \dot{e}_1 \; \theta_1 \; \dot{\theta}_1 \; \hat{f}_1 \; (x_A) \right]^T, \tag{5.14}$$

and the critic network output is the approximation of the long-term cost function defined in Eq. (5.12):

$$J = \widehat{W}_C^T \hat{\sigma}_C \left(\hat{V}_C^T x_C \right) = \sum_{i=1}^{L_C} \widehat{W}_C^T(1,i) \hat{\sigma}_C \left(\sum_{j=1}^{N_C} \hat{V}_C^T(i,j) x_C(j,1) \right), \tag{5.15}$$

with L_C is the number of nodes in the hidden layer, and $N_C = 5$ is the number of inputs to the critic network.

Weights of the critic network are trained as follows:

$$\dot{\widehat{W}}_C = \alpha F_C e_C \hat{\sigma}_C - k_C F_C \|e_C\|_2 \widehat{W}_C \tag{5.16}$$

$$\dot{\hat{V}}_C = \alpha G_C e_C x_C \widehat{W}_C^T \hat{\sigma}'_C - k_C G_C \|e_C\|_2 \widehat{V}_C, \tag{5.17}$$

in which α is the discount factor, F_C, G_C, k_C are design parameters, and $\hat{\sigma}'_C$ is the Jacobian matrix defined as:

$$\hat{\sigma}'_C = \frac{\partial \hat{\sigma}_C(\hat{V}_C^T x_C)}{\partial(\hat{V}_C^T x_C)}.$$

Action Network. In general, the action network is responsible for generating a control action which results in the optimization of the approximated long-term cost function, i.e. the output of the critic network. Therefore, the backpropagation error of the action network is given as follows:

$$e_A(t)\ \underbrace{U_C(t)}_{TARGET} - \underbrace{J(t)}_{\substack{CURRENT\\ OUTCOME}}\quad, \tag{5.18}$$

where $U_C(t)$ is an ultimate control goal, or the target for the long-term cost approximate $J(t)$.

Inputs to the action network are:

$$x_A = [e_1 \quad \dot{e}_1 \quad \theta_1 \quad \dot{\theta}_1]^T, \tag{5.19}$$

and structure of the action network is as follows:

$$\hat{f}_1 = \widehat{W}_A^T \hat{\sigma}_A\left(\hat{V}_A^T x_A\right) = \sum_{i=1}^{L_A} \widehat{W}_A^T(1,i)\hat{\sigma}_A\left(\sum_{j=1}^{N_A} \hat{V}_A^T(i,j)x_A(j,1)\right), \tag{5.20}$$

in which L_A is the number of nodes in the hidden layer, and $N_A = 4$ is the number of inputs to the action network.

Weights of the action network are trained as follows:

$$\dot{\widehat{W}}_A = F_A e_A \hat{\sigma}_A \hat{V}_{CA} \hat{\sigma}_C' \widehat{W}_C - F_A \hat{\sigma}_A' \hat{V}_A^T x_A r_1 - k_A F_A \|e_A\|_2 \widehat{W}_A \tag{5.21}$$

$$\dot{\hat{V}}_A = G_A e_A x_A \hat{V}_{CA} \hat{\sigma}_C' \widehat{W}_C \widehat{W}_A^T \hat{\sigma}_A' - k_A G_A \|e_A\|_2 \hat{V}_A, \tag{5.22}$$

in which $\hat{V}_{CA}$ is contained in $\hat{V}_C$ to map from $\hat{f}_1(x_A)$ to the hidden node output, F_A, G_A, k_A are design parameters, and $\hat{\sigma}_A'$ is the Jacobian matrix defined as:

$$\hat{\sigma}_A' = \frac{\partial \hat{\sigma}_A(\hat{V}_A^T x_A)}{\partial(\hat{V}_A^T x_A)}.$$

Compared to the weight updating rules in [13], it is noted that the last terms in (5.16), (5.17), (5.21), and (5.22) provide robustness against the disturbances generated by the hip acceleration and external ground reaction torque during the gait. Finally, the DNDP-based control is given as in (5.11).

Short-term Performance Index (Cost) and Ultimate Control Goal. The short-term (or instantaneous) cost is calculated as follows:

$$S(t) = -\frac{1}{2}\left(\frac{\theta_{1r}(t) - \theta_1(t)}{\theta_{1M}}\right)^2 - \frac{1}{2}\left(\frac{\dot{\theta}_{1r}(t) - \dot{\theta}_1(t)}{\dot{\theta}_{1M}}\right)^2, \tag{5.23}$$

where $\left\{\theta_1(t), \dot{\theta}_1(t)\right\}$, $\left\{\theta_{1r}(t), \dot{\theta}_{1r}(t)\right\}$ and $\left\{\theta_{1M}(t), \dot{\theta}_{1M}(t)\right\}$ are actual, desired, and maximal values of the ankle joint angular position and velocity. This selection relates to the gait efficiency in the way that if the prosthetic ankle joint can perform as closed as possible to the biological ankle, then the hip and knee joints do not have to modify their behaviors to compensate for gait degradation. Consequently, amputees can walk with no or little unnecessary extra effort/energy consumption. As the result, the overall human-prosthetic system can perform a normal gait.

The desire for optimization of the gait efficiency can be interpreted as the optimization of the approximated long-term cost function $J(t)$. Since the short-term cost function $S(t)$ is defined in term of tracking errors as in (5.23), the ultimate control goal for $J(t)$ is selected as $U_C(t) = 0$.

5.4 Numerical Study

Performance of the DNDP-based control structure is evaluated in an overall amputated leg—prosthetic foot system with the presence of measurement/actuator noises and variations in walking speed.

5.4.1 Simulation Setup

In this section, the proposed control structure will be evaluated in a framework which mimics the human gait. In this framework, desired joint trajectories are generated based on human gait data, and the ideal computed torque control algorithms generate the required torques at the hip and knee joints.

Kinematic Pattern Generation. In order to study the effectiveness of the DNDP-based control strategy, the behavior of the overall human-prosthetic system under different gait conditions has to be investigated. Different gaits are represented by kinematic patterns of angular positions, velocities, and accelerations of the joints. These quantities are obtained from the gait lab database [22] from real human subjects which are widely used as a reference for studying of human gait and humanoid robots. From the gait lab database, the analytical forms of the desired joint trajectories in time domain are used to perform multi-step simulation of the model.

The desired joint trajectories of the hip, knee, and ankle joints, and the vertical Cartesian trajectory of the hip joint are approximated by five-term Fourier series as in Eq. (5.24). The horizontal Cartesian trajectory of the hip joint is approximated by the sum of a first order polynomial (linear) and five-term Fourier series as in Eq. (5.25). From these analytical forms, the first and second order derivatives can be calculated without introducing any discontinuities in the kinematic patterns.

$$\{\theta_{3r}(t), \theta_{2r}(t), \theta_{1r}(t), z_{Hr}(t)\} = a_0 + \sum_{k=1}^{5} [a_i \cos(k\omega t) + b_i \sin(k\omega t)] \quad (5.24)$$

$$x_{Hr}(t) = k_0 t + m_0 + c_0 + \sum_{k=1}^{5} [c_i \cos(k\omega t) + d_i \sin(k\omega t)] \quad (5.25)$$

Kinematic data collected from human subjects during walking with different cadences (natural, fast, slow) in the gait lab [22] is converted to represent the kinematic patterns for the human-prosthetic dynamic model in corresponding gaits with normal, fast, and slow walking speed.

Control of the Hip and Knees Joints. For the biological hip and knee joints, it is assumed that below-knee amputees are able to adjust their muscle activities to generate enough torques to manipulate these joints and maintain normal gait despite possible control efforts at the prosthetic ankle joint. For that reason, ideal computed torque control is applied at the hip and knee joints. These ideal torques are computed assuming that the (noisy) joint angles, angular velocities, and angular accelerations, as well as the nonlinear terms in (5.1) are known. Such control inputs have the same structure as the ideal computed torque control for robot manipulators [21].

Equations (5.26) and (5.27) describe the ideal computed torque control applied at the biological hip and knee joints during simulation of the model, in which $e_i = (\theta_{ir} - \theta_i)$ and $\dot{e}_i = (\dot{\theta}_{ir} - \dot{\theta}_i)$ are tracking errors, $\ddot{\theta}_{ir}$ is a desired angular acceleration of each joint, $K_{Di} > 0$, $K_{Pi} > 0$ are design parameters. Control parameters for the ideal computed torque controllers at the biological hip and knee joints are $K_{Pi} = 10$ and $K_{Di} = 5$.

$$\begin{aligned}\tau_2 = & M_{21}\left(\ddot{\theta}_{1r} + K_{D1}\dot{e}_1 + K_{P1}e_1\right) + M_{22}\left(\ddot{\theta}_{2r} + K_{D2}\dot{e}_2 + K_{P2}e_2\right) \\ & + M_{23}\left(\ddot{\theta}_{3r} + K_{D3}\dot{e}_3 + K_{P3}e_3\right) + V_{21}\dot{\theta}_1 + V_{22}\dot{\theta}_2 + V_{23}\dot{\theta}_3 \\ & + G_2 + F_{21}\ddot{x}_H + F_{22}\ddot{z}_H - D_{21}F_X - D_{22}F_Z\end{aligned} \quad (5.26)$$

$$\begin{aligned}\tau_3 = & M_{31}\left(\ddot{\theta}_{1r} + K_{D1}\dot{e}_1 + K_{P1}e_1\right) + M_{32}\left(\ddot{\theta}_{2r} + K_{D2}\dot{e}_2 + K_{P2}e_2\right) \\ & + M_{33}\left(\ddot{\theta}_{3r} + K_{D3}\dot{e}_3 + K_{P3}e_3\right) + V_{31}\dot{\theta}_1 + V_{32}\dot{\theta}_2 + V_{33}\dot{\theta}_3 \\ & + G_3 + F_{31}\ddot{x}_H + F_{32}\ddot{z}_H - D_{31}F_X - D_{32}F_Z.\end{aligned} \quad (5.27)$$

Control Parameters for the Prosthetic Ankle Joint. For the prosthetic ankle joint, the DNDP-based control is generated by an action network with 4 nodes in the input layer, 8 nodes in the hidden layer, and 1 node in the output layer. The critic network has 5 nodes in the input layer, 10 nodes in the hidden layer, and 1 node in the output layer. Both networks use sigmoid activation functions and are fully connected with randomly initialized weights in the range $[-1, 1]$. Other design parameters include the discount factor $\alpha = 0.95$ and PD control with $K_{V1} = 5$ and $\lambda_1 = 10$. The unknown nonlinear function $f_1(x)$ is approximated by $\hat{f}_1(x)$ in (5.20). The critic network and action network weights are updated using (5.16)–(5.17) and (5.21)–

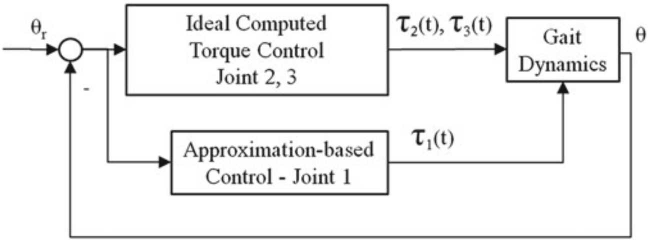

θ_r
-
Ideal Computed Torque Control Joint 2, 3
$\tau_2(t), \tau_3(t)$
Gait Dynamics
θ
Approximation-based Control - Joint 1
$\tau_1(t)$

Fig. 5.4 Control structure with ideal computed torque control at hip and knee joints, and approximation-based control at the prosthetic ankle joint

(5.22), respectively. Equation (5.23) is used to calculate the short-term cost at each time step. Figure 5.4 shows the structure of the controllers used in this study.

5.4.2 Simulation Results

Ideal Condition. In this ideal condition, the model is simulated during a gait including 20 steps of normal speed without any measurement and actuator noises. The tracking performance of the ankle joint and DNDP-based torque for 5 steps are shown in Fig. 5.5. It is observed that both the ankle position and angular velocity can follow their desired trajectories with small errors. As expected, the DNDP-based ankle torque generated during simulation of the model is very similar to the biological ankle torque calculated from human subjects during gait lab testing [22].

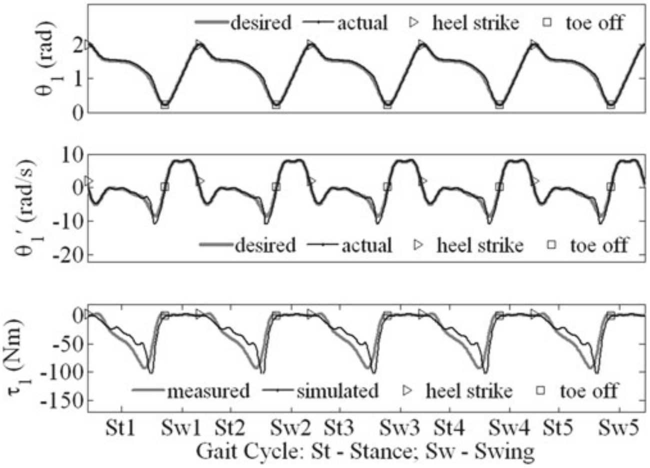


Fig. 5.5 Tracking performance of the DNDP-based control during normal speed under ideal conditions

Table 5.1 Long-term cost after 20 steps of normal walking speed with increasing measurement/actuator noises

Noise	PD	FLNN	DNDP
2 % Measurement noise	0.715	0.239	0.075
5 % Measurement noise	3.96	2.003	0.118
5 % Measurement noise and 2 % actuator noise	3.961	2.079	0.120
5 % Measurement noise and 5 % actuator noise	3.966	2.336	0.130

Effect of Measurement and Actuator Noises. Uniformly distributed measurement noises are added to the ankle position and angular velocity. Torque output generated for the ankle joint is also added with uniformly distributed actuator noise as follows:

$$\begin{aligned}\theta_1 &= \theta_1 + \rho\theta_1\\ \dot{\theta}_1 &= \dot{\theta}_1 + \rho\dot{\theta}_1\\ \tau_1 &= \tau_1 + \rho\tau_1,\end{aligned}$$

where ρ is in the range $[-2\,\%\,, 2\,\%\,]$ (or $[-5\,\%\,, 5\,\%\,]$). The model is simulated with 20 steps of normal walking speed and increasing measurement and/or actuator noises (see Table 5.1).

For the comparison purpose, the simulation is repeated with other types of control at the ankle joint including Proportional-Derivative control (PD) as:

$$\tau_1 = K_{V1}r_1 = \ K_{P1}e_1 + K_{D1}\dot{e}_1, \tag{5.28}$$

and direct Feedback Linearization-based multi-layer Neural Network control (FLNN):

$$\tau_1 = \hat{f}_{1,FLNN}(X) + K_{V1}r_1 - \nu.$$

in which ν is the robustifying term to compensate for approximation errors and unknown disturbances. It is noted that the approximation of $\hat{f}_{1,FNNN}(x)$ bases on backpropagation of the tracking errors and does not involve the optimization of the approximated long-term cost $J\,(t)$ as used in (5.18). Ideal computed torque controls are still used at the hip and knee joints.

The average long-term cost function is reported in Table 5.1. It can be seen that as the measurement/actuator noises increase, the DNDP-based control outperforms other control methods by producing robust tracking performance with lower long-term cost.

Effect of Variations in Walking Speed. Control configurations similar to previous simulation scenarios are repeated here to evaluate the performance of the DNDP-based control in the presence of variations in walking speed. The model is simulated with 5 % measurement noise, 5 % actuator noise, and 4 different walking setups (see Table 5.2).

Table 5.2 Long-term cost with 5 % measurement noise, 5 % actuator noise, and combinations of different walking speeds

Number of steps	PD	FLNN	DNDP
10 Normal + 10 fast	2.140	0.567	0.100
10 Normal + 10 slow	3.910	1.915	0.106
10 Normal + 5 fast + 5 slow	2.233	0.461	0.082
10 Normal + 5 slow + 5 fast	2.206	0.490	0.084

Again, despite the variations in walking speed, the DNDP-based control is still able to provide lower long-term cost compared to other control strategies.

5.5 Conclusions

This paper evaluates the performance and potential of a model-free adaptive dynamic programming-based controller for a prosthetic ankle joint. Issues such as gait dynamics formulation, desired ankle joint behaviors, control strategies, augmented training rules, and long-term gait-related efficiency were addressed in order to implement the DNDP-based control approach. Simulation scenarios indicate that with the DNDP-based control, the prosthetic ankle joint is able to provide stable, robust, and optimized performance during gait. Results of this study serve as a starting point for the development of intelligent ankle prostheses. Future works include hardware realization of the DNDP-based control strategy on an actual prosthetic foot, and adaptive determination of gait using biological feedback from amputees.

References

1. Versluys, R., Beyl, P., Damme, M.V., Desomer, A., Ham, R.V., Lefeber, D.: Prosthetic feet: state-of-the-art review and the important of mimicking human ankle-foot biomechanics. Disability Rehabil Assistive Technol **4**, 65–75 (2009)
2. Össur. http://www.ossur.com/?PageID=12704
3. BIOM. http://www.biom.com/
4. Hitt, J., Sugar, T., Holgate, M., Bellman, R., Hollander, K.: Robotic transtibial prosthesis with biomechanical energy regeneration. Int. J. Ind. Robot. **36**, 441–447 (2009)
5. Versluys, R., Desomer, A., Lenaerts, G., Damme, M.V., Beyl, P., Perre, G.V.d., Peeraer, L., Lefeber, D.: A pneumatically powered below-knee prosthesis: design specifications and first experiments with an amputee. In: 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechanics, pp. 372–377. Scottsdale, AZ, USA (2008)
6. Millard, M., McPhee, J., Kubica, E.: Multi-step forward dynamic gait simulation. Comput. Methods Appl. Sci. **12**, 25–43 (2008)
7. Peasgood, M., Kubica, E., McPhee, J.: Stabilization of a dynamic walking gait simulation. ASME J. Comput. Nonlinear Dyn. **2**, 65–72 (2007)
8. Thelen, D.G., Anderson, C.: Using computed muscle control to generate forward dynamic simulations of human walking from experiment data. J. Biomech. **39**, 1107–1115 (2006)

9. Xiang, Y., Arora, J.S., Abdel-Malek, K.: Physics-based modeling and simulation of human walking: a review of optimization-based and other approaches. Med. Bioeng. Appl. **42**, 1–23 (2010)
10. Pejhan, S., Farahmand, F., Parnianpour, M.: Design optimization of an above-knee prosthesis based on the kinematics of gait. In: 30th Annual International IEEE EMBS Conference, Vancouver, British Columbia, Canada (2008)
11. Brugger, P., Schemiedmayer, H.-B.: Simulating prosthetic gait—lessons to learn. Proc. Appl. Math. Mech. **3**, 64–67 (2003)
12. Hansen, H.: Scientific methods to determine functional performance of prosthetic ankle-foot systems. J. Prosthet. Orthot. **17**, 23–29 (2005)
13. Si, J., Wang, Y.-T.: On-line learning control by association and reinforcement. IEEE Trans. Neural Networks **12**, 264–276 (2001)
14. Lu, C., Si, J., Xie, X.: Direct heuristic dynamic programming for damping oscillations in a large power system. IEEE Trans. Syst. Man Cybern. Part B Cybern. **38**, 1008–1013 (2008)
15. Enns, R., Si, J.: Helicopter trimming and tracking control using direct neural dynamic programming. IEEE Trans. Neural Networks **14**, 929–939 (2003)
16. Amirouche, L.: Computational Methods in Multibody Dynamics. Prentice Hall, Englewood Cliffs, NJ (1992)
17. Winter, A.: Biomechanics and Motor Control of Human Movement. Wiley, Hoboken, NJ (2009)
18. Wojtyra, M.: Multibody simulation model of human walking. Mech. Based Des. Stuct. Mach. **31**, 357–379 (2003)
19. Bruneau, O., Ouezdou, B.: Compliant contact of walking robot feet. In: 3rd ECPD International Conference on Advanced Robotics, Bremen, Germany (1997)
20. Marhefka, D.W., Orin, D.E.: A compliant contact model with nonlinear damping for simulation of robotic systems. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **29**, 566–572 (1999)
21. Lewis, F.L., Jagannathan, S., Yesilderek, A.: Neural network control of robot manipulators and nonlinear systems. Taylor & Francis, London, UK (1999)
22. Winter, D.A.: The biomechanics and motor control of human gait: normal, elderly and pathological. University of Waterloo Press, Waterloo, Ontario, Canada (1991)

Chapter 6
An Agent Based Layered Decision Process for Vehicle Platoon Control

Baudouin Dafflon, Franck Gechter, Pablo Gruer and Abder Koukam

Abstract Vehicle platoon systems can be considered as good alternative solutions to traffic problems encountered in urban environment. Indeed, they allow to propose new public transportation models based on size adaptable trains. In other fields, such as military operation theaters and agricultural fields, platoon systems can also solve specific problems. The goal of this paper is to propose an agent based decision process that allow to handle the control of vehicles platoon. The decision process is composed of 5 interconnected layers each dealing with one specific aspect of the decision process including the perception interpretation, the spatial configuration choice and the control of the vehicles. This model has been tested in simulation by taking into account several geometrical configurations.

Keywords Multi-Agent model · Platoon control

6.1 Introduction

Platoons can be defined as a set of autonomous vehicles evolving on a particular environment while maintaining a particular geometric configuration. Platoon control consists in determining the behaviour of each one of the vehicles during platoon evolution, in order to maintain the configuration and adapt it to changes in the terrain (presence of unanticipated obstacles, decreasing of available surface, …).

B. Dafflon (✉) · F. Gechter · P. Gruer · A. Koukam
Laboratoire Systèmes et Transports, Université de Technologie de Belfort-Montbéliard, Belfort, France
e-mail: baudouin.dafflon@utbm.fr
http://set.utbm.fr

F. Gechter
e-mail: franck.gechter@utbm.fr

P. Gruer
e-mail: pablo.gruer@utbm.fr

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_6

Several international projects, past or present, address platoon control. Among them, we can cite PATH [1], SARTRE [2], CRISTAL,[1] SAFEPLATOON [3].[2] Most of them deal with column platoons as unique configuration and situate in well defined environment such as highways where the curve radius are high and where the speed can be considered to be constant most of the time. However, other application domains, placed in different kinds of environments could benefit from platoons composed of different types of vehicles. Among those application domains we can mension transportation and maintenance operations in urban areas, labouring and harvesting in agricultural areas and military operation theatres. Those cases are subject to diverse, more stringent constraints. As an example, in urban areas with a column configuration, the lateral error must be highly limited in order to avoid collisions. In echelon or line configurations, environment related constraints are more influential.

In this paper, we propose an multi-agent based approach for the multi-configuration platoon control problem. The approach can be considered as self-organizing, because platoon configuration emerges from the behaviour of each vehicle, strictly based on local vehicle's perceptions. The platoon's configuration is determined locally by assigning to any platoon vehicle another neighbouring platoon vehicle, considered as local leader. This proposal is structured as a multi-layer decision process dealing first with the interpretation of the perception data, then integrating the choice of the local leader depending on the intended spatial configuration and finally producing a kinematic decision that has to be performed by the vehicle. In this approach, each vehicle is considered as an agent which bases on its perceptions and on the local intended configuration to make the correct decision. The proposed approach integrates obstacle avoidance abilities which is based on a multi-agent filtering method similar as the one developed in [4].

The paper is structured as follow: after a short reminder about vocabulary, a state of the art on platoon systems is proposed. Then, the multi agent system applied to platoon control is described. Finally, a conclusion and some considerations on future work directions are presented.

6.2 Definitions

The literature introduces a rich terminology in related to the platoon domain. In this section, we intend to present the vocabulary used along the work, in order to avoid ambiguities.

[1] http://projet-cristal.net/

[2] http://web.utbm.fr/safeplatoon

6.2.1 Leaders

Vehicles with a distinctive role within the platoon are frequently qualified as leaders. We distinguish two kind of leader roles.

Global Leader: The global leader is the reference vehicle of the entire platoon. It can be fully autonomous, applying a path-following algorithm, or driven by a human operator. The global leader determines the reference trajectory for the convoy.

Local Leader: The local leader notion is tied to local, self-organizing approaches, and corresponds to the vehicle taken as a reference by a follower vehicle. Each vehicle in the platoon has a local leader, but this role can be assigned dynamically during platoon operation. Generally, the local leader is taken among the closest vehicles in the follower perception field.

Virtual Leader: The notion of virtual leader is tied to the mechanisms involved in our methods. The principle developed in this paper, is to be able to transform any spatial configuration into a local column configuration of a local leader and its follower, and to apply a well defined interaction model, to determine follower's behaviour.

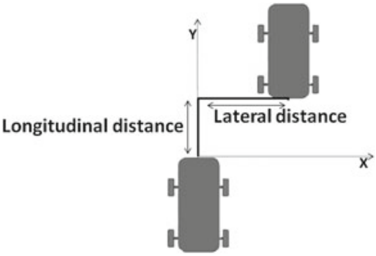
6.2.2 Geometry

A platoon configuration geometry is defined by means of two distances, lateral and the longitudinal (cf. Fig. 6.1). The **Lateral distance** represents the lateral spacing between two neighbour vehicles. The **Longitudinal distance** represents the spacing between two neighbour vehicles, in the direction of the move.

Configuring a platoon formation bases on the definition of both lateral and longitudinal distance. Depending on the values of these, several platoon configuration can then be defined, among which we can mention:

Column Configuration. This configuration, represents the most frequently studied form of platoon where vehicles organize as train (cf. Fig. 6.2). In this configuration, lateral distance should remain as small as possible (for curved trajectories this means mono-trace displacement). Column configurations have been foreseen as mostly dedicated for the transport of passengers in urban or highway transportation systems.

Fig. 6.1 Lateral and longitudinal distances



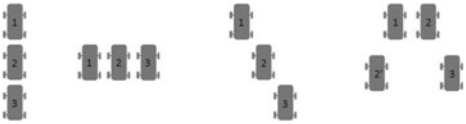


Fig. 6.2 Configurations of platoon. From *left to right* column, line, echelon, wedge

Line Configuration. In this configuration, vehicles are placed side by side (cf. Fig. 6.2). The longitudinal distance must be null. This configuration can be applied to agricultural tasks, such as tilling.

Echelon Configuration. Vehicles are in an inclined column configuration, each is offset from the preceding by a lateral distance (cf. Fig. 6.2). In this configuration, lateral and longitudinal nominal distances have a specified, non null value. This configuration can be dedicated to agricultural or military applications.

Arbitrary Configuration and **Wedge Configuration**. Arbitrary configurations can hold many geometrical forms, produced by the combinations of two or three of the configurations described above. These configurations are mostly used in military environments. In wedge configuration for instance, leader vehicle is followed by echelons of vehicles placed to the right and the left forming an inverted "V" formation (cf. Fig. 6.2).

6.3 State of the Art

Platoon control approaches can be sorted into two main categories: local approaches base on a local reference frame. On the other hand, global approaches base on a global frame, common to every vehicle in the platoon.

6.3.1 Local Approaches

Local approaches are based on a local reference frame which generally anchors in the follower vehicle. In this frame, the position and the orientation of the local leader can be determined in the local frame. Consequently, the local approach can be considered as the regulation of both lateral and longitudinal distances between the follower and its local leader.

Among the proposals corresponding to this approach, we can mention:

- System based on an automatic control mechanism: [5] proposes a mechanism for local control based on a PID (Proportional, Integral, Derivative) controller. The measurement acquisition is performed by linear cameras or by a range finder sensor [6]. The control reference is decomposed on one hand in a longitudinal reference and on the other hand in a lateral reference.
- Control based on undumped impedances: [7] presents a model of light links where vehicles are treated as particles subjected to physical forces.

- Control model based on a double impedance control: in [8] an impedance control model is used as a model of the immaterial link between vehicles.
- Control based on a virtual mechanical link: In these approaches, each vehicle in the convoy is designed as an intelligent system able to perceive its environment and maintaining a pre-set distance with the preceding vehicle. The platoon system is the result of direct interaction between each vehicle and its predecessor. This approach is based on an interaction model inspired by the physics. Indeed, the interaction between two successive vehicles virtual link shown by a mass-spring type. Among these one can cite [4].

6.3.2 Global Approaches

Each vehicle determines its control references depending on a global shared reference. Platoon trajectory is determined by the global leader and expressed as a series of trajectory points situated in the global reference frame, known by every vehicle. The sharing of the trajectory points implies vehicle to vehicle communication capabilities and high performance global localisation devices aimed at determining the position of each vehicle in the global reference frame. Platoon control can be considered as the regulation of both lateral and longitudinal distance between each vehicle and the reference trajectory.

Among the proposals corresponding to this approach, we can mention:

- An optimal controller is proposed in [9] to serve on a constant set of inter-vehicle distance in columns moving at high speed.
- In [10], the formation is characterized by a set of generalized coordinates describing the position and the orientation. The resulting shape and evolution of the column is based on the laws of mechanics.

Relatively to military platoons enumerated, these can fit the global/local classification. Unit center referenced and leader referenced approaches can be considered as global (the control can be centralized or decentralized). By contrast, Neighbour-referenced technique is local and decentralized.

6.4 Multi Layers Decision Process

6.4.1 Global Overview

Our proposal is based on a multi-layer systems for decision making. This architecture provides a set of 5 plug and play units (cf. Fig 6.3). The principle developed is based on the transformation of any spatial configuration into a column configuration which uses a well defined interaction model. Then, we use a virtual leader, the position and

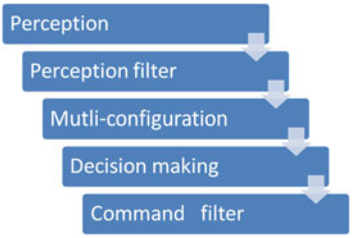


Fig. 6.3 Multi-layer architecture

the orientation of which correspond to the transformation of the local leader position and orientation taking into account platoon spatial desired configuration.

Each step of the process is independent and has got specific parameters and can be defined as follow: **Perception** is an abstraction of sensors. The input data are the raw information from the sensors. A parser is used to convert them in workable data set for further decision-making process.

In **perception filter** block a filtering policy is applied. The goal of this unit is to sort out the perceived elements between obstacles and other convoy vehicles. Among these vehicles a local leader is chosen using specific strategy.

The **multi-configuration** unit applies the geometrical transformations required to change local leader position and orientation into virtual leader ones.

The **decision-making** block corresponds to the application of the interaction model and the computing of the command law to be applied to the vehicle.

The last step in this process is an **command filtering** which can corresponds to driving assistance filtering such as obstacle avoidance or to the introduction of a kinematic model for the smooth command for instance.

6.4.2 Detailed Description

This section aims at describing layers one by one following the introduction description and Fig. 6.3.

6.4.2.1 Perception

The perception takes as an input the raw data of the sensor associated to object detection algorithm. To be able to have reliable detection, sensors have to cover the surrounded environment of the vehicle as shown in Fig. 6.4. The output of the perception unit is a list of localized object in the vehicle reference frame. Various kind of sensors can be used to performed this process such as stereo cameras, sonars belt, laser range finders ...

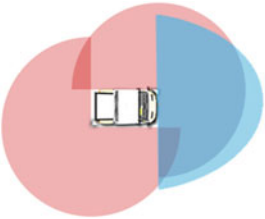


Fig. 6.4 Sensors ideal coverage area

6.4.2.2 Perception Filter

The goal of this unit is to analyse the output of the perception in order to remove noise in the data and to sort out detected objects. A Kalman filter has been chosen to filter the noise. A Kalman filter is a prediction/correction based filter which uses a transition model for the prediction and an observation model for the correction. As for the sorting of the objects a simple strategy has been used. the aim is to provide a classification of the environment in 3 class.

1. object member of convoy
2. object moving in same direction but not member of convoy
3. object moving perilously.

This classification is based on dynamic study of perceived object.

Then, the final step of this filtering unit is a geometrical transformation aimed at expressing the coordinates of the detected objects into the vehicle reference frame. (Initially, their coordinates are expressed into the sensors reference frames)…

6.4.2.3 Multi-configuration

As exposed before, each vehicle in the platoon can be seen as an agent that acts based only on its perceptions. For each agent, we define a leader among its neighbours in the platoon (see Sect. 6.4.2). The agent computes its references based on the position of its leader by trying to maintain the desired lateral and longitudinal spacing and the correct orientation. Column formation platoon control functions are now well known and expose reliable properties. Consequently, it as been decided to base our approach on this elementary function. So, the key step is to translate leader local position in vehicle agent reference frame in order to be able to use column platoon function and integrates desired lateral and longitudinal distances. As exposed in Sect. 6.2, there are several types of configuration that can be grouped into several families (echelon, line, column). The multi configuration proposal is based on modification of perception by introduction of virtual leader vehicle. Indeed, as detailed in Fig. 6.5, every formations are possible with the introduction of a virtual leader.

In order to determine position of virtual leader, several parameters are used: geometrics aspects, perception and configuration order. Indeed, a translation of la and

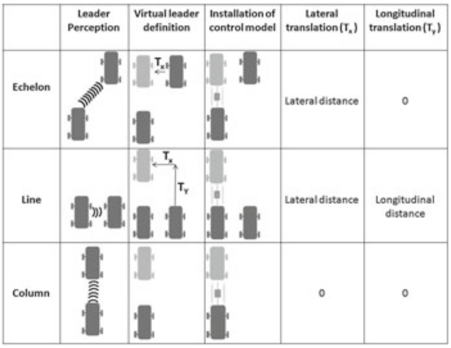


Fig. 6.5 Formations tab

	Leader Perception	Virtual leader definition	Installation of control model	Lateral translation (T_x)	Longitudinal translation (T_y)
Echelon		T_x		Lateral distance	0
Line		T_x T_y		Lateral distance	Longitudinal distance
Column				0	0

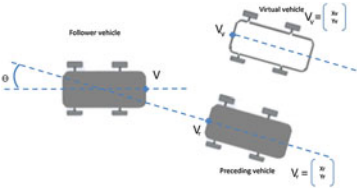

Virtual vehicle
$V_v = \begin{pmatrix} x_v \\ y_v \end{pmatrix}$
V_v
Follower vehicle
θ
V
V_r
Preceding vehicle
$V_r = \begin{pmatrix} x_r \\ y_r \end{pmatrix}$

Fig. 6.6 Virtual leader in multi configuration

lo is made in the referential of local leader. Position of virtual leader is defined by a 2D Transformation (Fig. 6.6) could be describe by following equation:

$$\begin{cases} x_v = x_r + l_a \cos(\theta) + l_o \sin(\theta) \\ y_v = y_r - l_a \sin(\theta) + l_o \cos(\theta) \end{cases} \tag{6.1}$$

where

- (x_v, y_v) virtual leader position
- (x_r, y_r) real leader position
- (l_a, l_o) lateral and longitudinal distances
- θ angle between cars.

6.4.2.4 Interaction Model

A model of interaction inspired by physics has been chosen. This model, detailed in [11] has got two springs and a damper placed between the local leader and the follower vehicle (cf. Fig. 6.7). This model is virtual and unidirectional. Indeed to better control the interaction, the local leader vehicle is not affected by generated forces.

The parameter of this model take into account the vehicles characteristics. This optimisation allow to keep a low latency and keep a real-time reactivity. A compositional verification of this system has been made in [12] to prove non collision event.

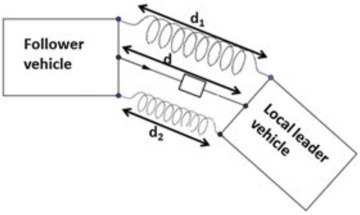


Fig. 6.7 Interaction model

6.4.2.5 Command Filter

The command proposed by the interaction model must be adapted to the environment in order to deal with the presence of obstacles for instance. Several filters can be considered. The most simple is an emergency stop filter: in case of obstacles in a neighbourhood too close, the command is set to 0. Another filter presented in previous work [4] and aimed at computing obstacle avoidance strategies can be apply. This obstacle avoidance device is based on a multi-agent system where the observation of an agents population lead to a modification of the vehicle command.

These filters require data from the perception. This is why a link is possible to bring directly sensors data from perception unit (or perception filter) to command filter as shown in the Fig. 6.8.

6.4.3 Flexible and Adaptive System

As seen above, our proposal is based on layers. Each module takes, as an input, the output of preceding block. Thus, provided that the input and the output are correct, layers can be changed using other algorithms or strategies. Moreover, it is possible to combined some inputs or to combine in cascade several blocks of the same type. Finally, these changes in blocks can be performed in runtime.

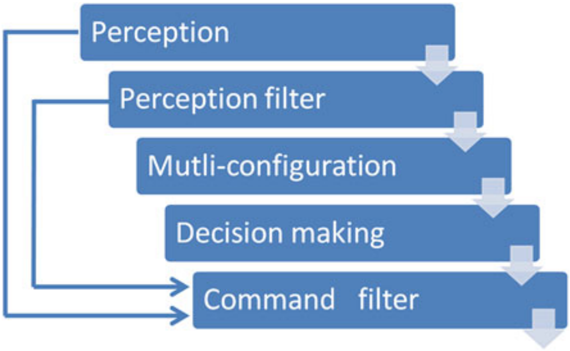
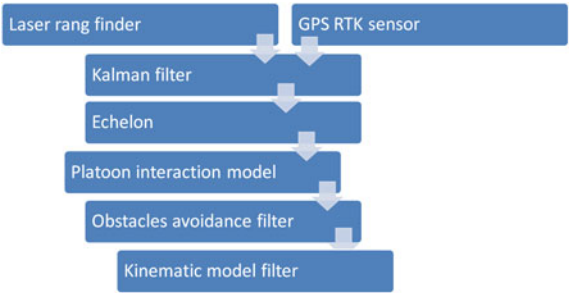


Fig. 6.8 Command filter input

Fig. 6.9 Input combined



Here are two examples to illustrate this modularity:

1. Input combined and association in cascade (Fig. 6.9)
2. Interchanged blocks (cf. Fig. 6.10)

6.5 Experimental Results

6.5.1 Global Overview

6.5.1.1 Vehicles Description

Vehicles used in the simulation represent (graphically and physically) the experimental laboratory's vehicles. They satisfy the physical constraints and share the same characteristics. In simulation, they are equipped with two 270° virtual laser range finder, replica of LMS SICK 200 and with GPS-RTK simulation required to follow and study trajectories. Vehicles have the following characteristics: 1.8 m width, 3.05 m length, max steering angle = 30° and max speed = 12 m/s.

6.5.1.2 Simulator

To assess the quality of our approach, simulations have been done using VIVUS simulator [13], VIVUS is a vehicle simulator based on PhysX for real physical behaviour developed by the SeT[3] laboratory.

This software can simulates behaviours for each vehicle such as perception with laser range finder or cameras, physical reaction between elements (wheels, car's parts, . . .), . . . Physical reaction are computed using the same physical law as real world (collision, gravity, . . .) and considering the peculiarity of the environment (friction with soil, materials of soils and walls, . . .). VIVUS has already been used to test

[3] http://set.utbm.fr/

various intelligent vehicle algorithms such as linear platoon control [4, 11], obstacle avoidance and driving assistance [14, 15], and intelligent crossroads simulations in [16].

6.5.1.3 Metrics and Analysis

Two informations are measured during experimentations:

- Lateral distance measures the spacing between the horizontal axes of two neighbour vehicles. In cases of column platoon, this distance should be null.
- Longitudinal distance represents the inter-vehicle distance between two neighbour vehicles.

Lateral and longitudinal distances are recorded by VIVUS simulator and analysed offline by matlab scripts. thanks to these measures, we will explore:—General behaviour of the convoy route with GPS—Lateral and longitudinal distances a function of time—Lateral and longitudinal distances a function of speed.

6.5.1.4 Test Area

Simulations were performed on a 3D geo-localized model of the city of Belfort (France). Two different trajectories have been chosen. In the first trajectory, vehicles have to follow a circle around 25 m of radius. The simulations are done many times (ie: 500 iterations) for the purpose a statistical studies about experimentations.

6.5.2 *Experimental Results*

Experimentations are made following two scenarios. The first simulates a convoy of vehicles in the column whereas the second simulates a echelon convoy.

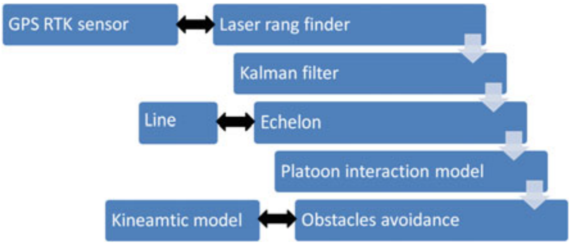


Fig. 6.10 Association in cascade

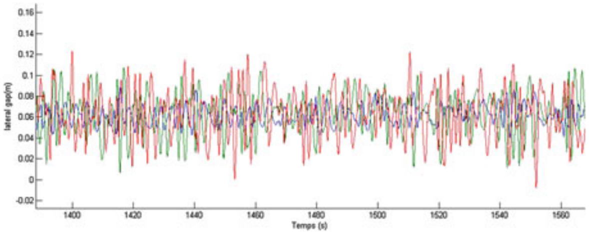


Fig. 6.11 Lateral gap in platoon convoy

6.5.2.1 5 Vehicle Platoon Convoy in Column

1. 5 Vehicle Platoon Convoy in Line:
   In Fig. 6.11, only the lateral gaps from the first, third, fourth and last follower are presented. We can see that the error between the measurement and instruction is amplified depending on the position of the vehicle in the convoy. The last follower error is between 0.07 and 0.55 m, while the first follower error is between 0.12 and 0.01 m. The difference between the measurement and setpoint comes from amplification errors due to the local approach. It is interesting to compare these measures to the width of a tire (about 0.2 m).
   In Fig. 6.12, only the longitudinal gaps from the first, third and last follower are represented. We can see that the error between the measurement and set point is amplified. The error of first follower is between 5.7 and 6.7 m whereas the last follower is between 5.2 and 6.4 m. The difference between the measurement and the reference is from an extension of the springs in the present interaction model.
2. 5 Vehicle Platoon Convoy in Echelon:
   Figure 6.13 shows the lateral deviation follower 1 and 5. We notice that the echelon formation does not lead additional oscillation as compared to a column formation. The virtual vehicle seems a reliable alternative for the inter-distance

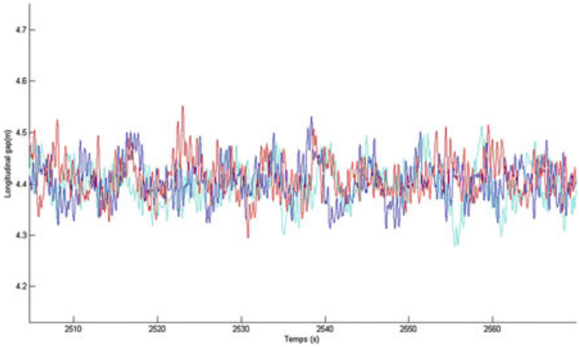


Fig. 6.12 Longitudinal gap in column platoon convoy

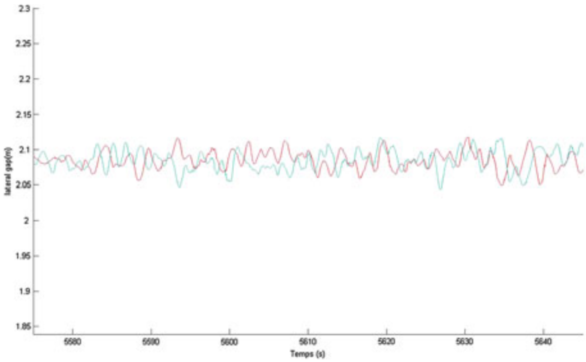


Fig. 6.13 Lateral gap in echelon platoon convoy

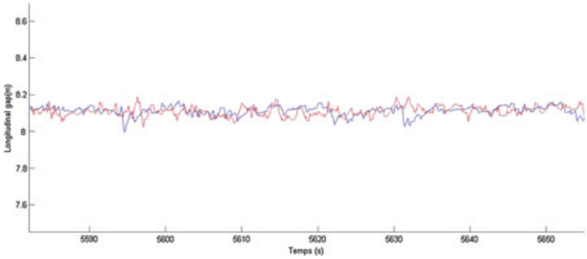


Fig. 6.14 Longitudinal gap in echelon platoon convoy

management of a multi-lateral configuration.

Figure 6.14 shows the longitudinal inter-distance measured on the train. We can notice that the error between the measurement and set point spreads by amplifying. The inter-distance of first follower is between 5.18 and 5.2 m while the last follower is between 5.01 and 5.22 m.

6.5.2.2 5 Vehicle Platoon Convoy in Circle

1. 5 Vehicle Platoon Convoy in Column:

   As shown in Fig. 6.15 the differences between first and last follower, we can see besides the residual oscillations that the differences between vehicles are of the same order as in previous experiment. From the first follower to the last, the longitudinal gap is in a range from 5.87 to 6.04 m.

   Figure 6.16 represents the lateral error from the first follower to the last. Thus, while the measurement of the first Follower are in the same order as before, the last follower describes an lateral error between −0.1 and 0.2 m.
2. 5 Vehicle Platoon Convoy in Echelon:

   Figure 6.17 describes the evolution in time of the lateral deviations. it may be noted that the position in the convoy does not increase the oscillations. Indeed,

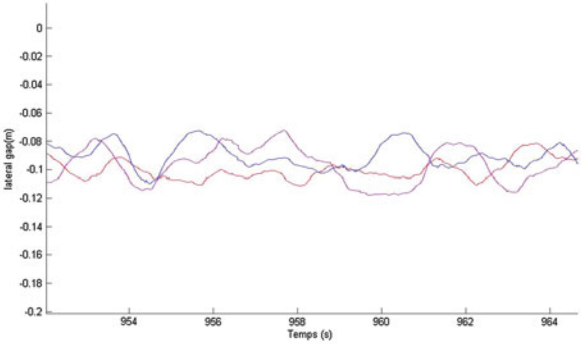


Fig. 6.15 Lateral gap in column platoon convoy

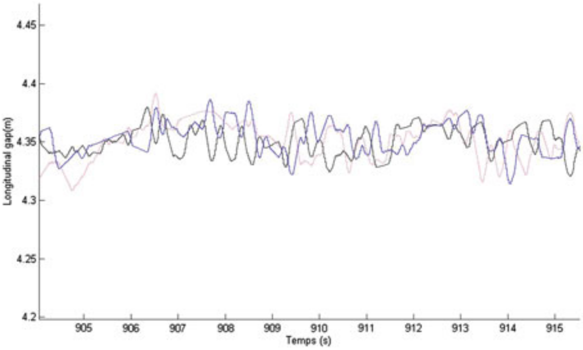


Fig. 6.16 Longitudinal gap in column platoon convoy

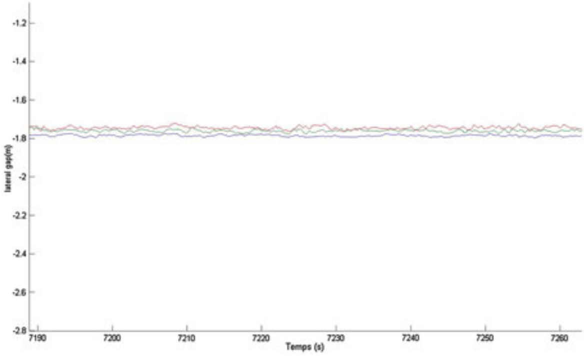


Fig. 6.17 Lateral gap in echelon platoon convoy

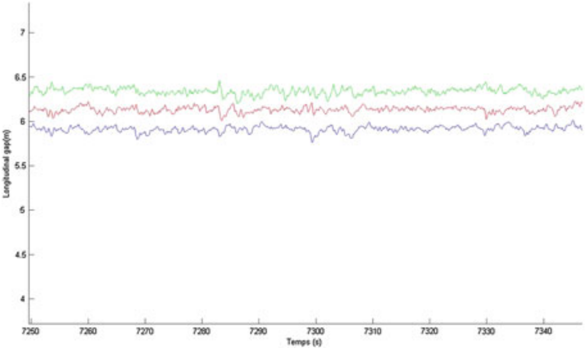


Fig. 6.18 Longitudinal gap in echelon platoon convoy

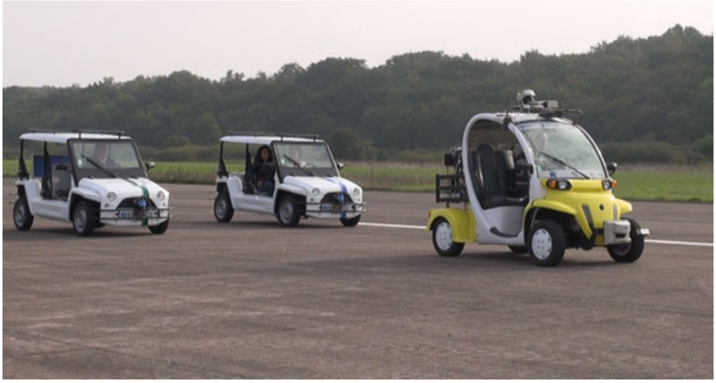


Fig. 6.19 Demonstration for SafePlatoon: 1 leader 2 follower in echelon configuration

with a circular path, the error is comprised between 0.09 and 0.11 m (less than the width of a tire). It is interesting to note that the train keeps the same stability with 3–5 vehicles.

Figure 6.18 shows that the most the vehicle is near the center, the more the error on the longitudinal set point is large. The reasons of this result are the links between anticipation, speed elongation and virtual vehicle transformation.

6.6 Conclusion

The paper presents a agent approach for platoon system through a generic and modular decision process for autonomous vehicle in platoon system. In this model, different layer is are proposed and combined to define behaviour. Each layer allow facilitate the processing done in next step. The main advantages proposed by this system is a run time and self adaptation to environment. This solution was successfully tested in simulation and results obtained are encouraging to test using real laboratory vehicles and real sensors.

This model was used in a demonstration in the project safeplaton (Fig. 6.19).

In order to continue this research, we are now working on generic and emerging perception filter to adapt perception to any sensors.

Those works are done with the support of the French ANR (National Research Agency) through the ANR-VTT *SafePlatoon*.[4] project (ANR-10-VPTT-011)

References

1. Hedrick, J., Tomizuka, M., Varaiya, P.: Control issues in automated highway systems. IEEE Control Syst. **14**(6), 21–32 (1994)
2. Chan, E., Gilhead, P., Jelinek, P., Krejci, P.: Sartre cooperative control of fully automated platoon vehicles.In: 18th World Congress on Intelligent Transport Systems (2011)
3. Cartade, P., Lenain, R., Thuilot, B., Berducat, M.: Formation control algorithm for a fleet of mobile robots.In: 3rd International Conference on Machine Control and Guidance (2002)
4. Gechter, F., Contet, J.M., Gruer, P., Koukam, A.: A reactive agent based vehicle platoon algorithm with integrated obstacle avoidance ability. Self-Adaptive and Self-Organizing Systems (2010)
5. Daviet, P., Parent, M.: Longitudinal and lateral servoing of vehicles in a platoon. In: IEEE Intel ligent Vehicles Symposium, pp. 41–46 (1996)
6. Riess, R.: Qualification d'un tlmtre balayage laser pour la robotique mobile: Intgration et exprimentations. Master's thesis (2000)
7. Gehrig, S., Stein, F.: Elastic bands to enhance vehicle following. In: IEEE Conference on Intelligent Transportation Systems, pp. 597–602 (2001)
8. Yi, S.Y., Chong, K.T.: Impedance control for a vehicle platoon system. Mechatronics (2005)
9. Levine, W., Athans., M.: On the optimal error regulation of a string of moving vehicles. In: IEEE Transactions on Automatic Control (1966)
10. Caicedo, R., Valasek, J., Junkins, J.: Preliminary results of one-dimensional vehicle formation control using structural analogy. In: Control Conference (2003)
11. El-Zaher, M., Gechter, F., Gruer, P., Hajjar, M.: A new linear platoon model based on reactive multi-agent systems. In: The 23rd IEEE International Conference on Tools with Artificial Intelligence ICTAI, IEEE Computer Society (2011)
12. El-Zaher, M., Contet, J., Gruer, P., Gechter, F.: Towards a compositional verification approach for multi-agent systems : application to platoon system. In: 1st International workshop on Verification and Validation of multi-agent models for complex systems (V2CS) (2011)
13. Lamotte, O., Galland, S., Contet, J.M., Gechter, F.: Submicroscopic and physics simulation of autonomous and intelligent vehicles in virtual reality. In: International Conference on, Advances in System Simulation, pp. 28–33 (2010)
14. Gechter, F., Contet, J., Gruer, P., Koukam, A.: Car driving assistance using organization measurement of reactive multi-agent system. Procedia CS 1(1): 317–325 (2010)
15. Dafflon, B., Contet, J.M., Gechter, F., Gruer, P.: Toward a reactive agent based parking assistance system. In: The 24rd IEEE International Conference on Tools with Artificial Intelligence ICTAI, IEEE Computer Society (2012)
16. Daffon, B., Gechter, F., Contet, J., Abbas-Turki, A., Gruer, P.: Intelligent crossroads for vehicle platoons reconfiguration. In: International Conference on Adaptive and Intelligent Systems (2011)

[4] http://safeplatoon.utbm.fr

Chapter 7
Tyre Footprint Geometric Form Reconstruction by Fibre-Optic Sensor's Data in the Vehicle Weight-in-Motion Estimation Problem

Alexander Grakovski, Alexey Pilipovecs, Igor Kabashkin and Elmars Petersons

Abstract The problem of measuring road vehicle's weight-in-motion (WIM) is important for overload enforcement, road maintenance planning and cargo fleet managing, control of the legal use of the transport infrastructure, road surface protection from the early destruction and for the safety on the roads. The fibre-optic sensors (FOS) functionality is based on the changes in the transparency of the optical cable due to the deformation of the optical fibre under the weight of the crossing vehicle. It is necessary for WIM measurements to estimate the impact area of a wheel on the working surface of the sensor called tyre footprint. Recorded signals from a truck passing over a group of FOS with various speeds and known weight are used as an input data. The results of the several laboratory and field experiments with FOS, e.g. load characteristics according to the temperature, contact surface width and loading speed impact, are provided here. The method of decomposition of input signal into symmetric and asymmetric components provides the chance to approximate geometric size of tyre surface footprint as well as calculate the weight on each wheel separately. The examples of the estimation of a truck tyre surface footprint using FOS signals, some sources of errors and limitations of possible application for WIM are discussed in this article.

Keywords Transport telematics · Weigh-in-motion · Fibre-optic sensor · Tyre footprint

7.1 Introduction

The worldwide problems and costs associated with the road vehicles overloaded axles are being tackled with the introduction of the new weigh-in-motion (WIM) technologies. WIM offers a fast and accurate measurement of the actual weights of the trucks when entering and leaving the road infrastructure facilities. Unlike the static weighbridges, WIM systems are capable of measuring vehicles traveling at a

A. Grakovski (✉) · A. Pilipovecs · I. Kabashkin · E. Petersons
Transport and Telecommunication Institute,Riga, Latvia
e-mail: avg@tsi.lv

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_7

reduced or normal traffic speeds and do not require the vehicle to come to a stop. This makes the weighing process more efficient, and in the case of the commercial vehicle allows the trucks under the weight limit to bypass the enforcement.

There are four major types of sensors that have been used today for a number of applications comprising the traffic data collection and overloaded truck enforcement: piezoelectric sensors, bending plates, load cells and fibre-optic sensors [4, 9]. The fibre-optic sensors (FOS), whose working principle is based on the change of the optical signal parameters due to the optic fibre deformation under the weight of the crossing road vehicle [1, 3, 6], have gained popularity in the last decade.

Analysis of the WIM current trends indicates that optical sensors are more reliable and durable in comparison to the strain gauge and piezoelectric sensors. Currently the two FOS types based on two main principles are used:

- Bragg grating (the change of diffraction in a channel under deformations);
- The fibre optical properties (transparency, frequency, phase and polarization) change during the deformations.

A lot of recent investigations are devoted to the peculiarities of the construction and applications of the sensors, using different physical properties. The data presented in this publication have been received using SENSOR LINE PUR experimental sensors [8] based on the change of the transparency (the intensity of the light signal) during the deformation.

7.2 Axles Weighing-in-Motion Principles

The fibre optic weight sensor is the cable consisting of a photoconductive polymer fibres coated with a thin light-reflective layer (Fig. 7.1b). A light conductor is created in such a way that the light cannot escape. If one directs a beam of light to one end of the cable, it will come out from the other end and in this case the cable can be twisted in any manner. To measure the force acting on the cable, the amplitude technology is more appropriated for the measurements based on measuring of the optical path intensity, which changes while pushing on the light conductor along its points.

At these points the deflection of a light conductor and reflective coating occurs, that is why the conditions of light reflection inside are changed, and some of it escapes. The greater the load the less light comes from the second end of the light conductor. Therefore the sensor has the unusual characteristic for those, familiar with the strain gauges: the greater the load the lower the output is. Apart from the fact that it is reversed and in addition to this it is non-linear.

In order to avoid the inaccuracy of zero load level we need to exclude the high frequency components from the voltage signal at the output of the sensor's transducer by filtering, as well as to recalculate the voltage signal U(t) (Fig. 7.1c) into the relative visibility losses signal V(t) (Fig. 7.1d), directly related to the weight pressure on the FOS surface. It can be done by the transformation (7.1):

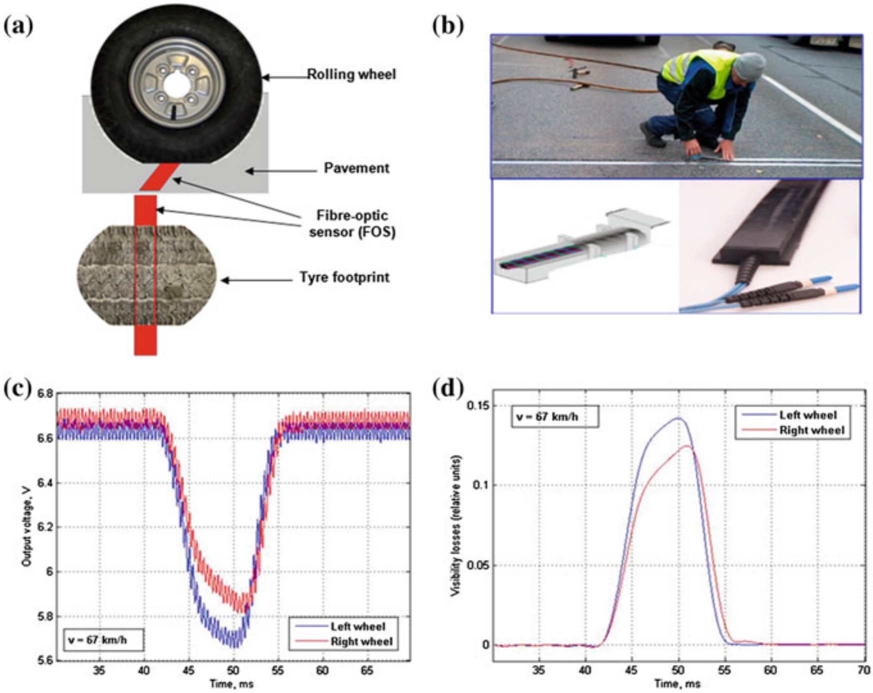


Fig. 7.1 **a** Fibre optic sensor's position against the wheel and tyre footprint. **b** SENSOR LINE PUR installation and construction of the sensor. **c** FOS output voltage. **d** Visibility losses as the function of pressure after pre-processing (filtering)

$$V(t) = \frac{U_0 - U(t)}{U_0} \tag{7.1}$$

where U_0 is the voltage of sensor's output with zero load. The signal transformation to the relative visibility losses signal V(t) gives the possibility to compare signals for different measurements in different conditions.

Fibre optic load-measuring cables are placed in the gap across the road, filled with resilient rubber (Fig. 7.1b). The gap width is 30 mm. Since the sensor width is smaller than the tyre footprint on the surface, the sensor takes only part of the axle weight. Two methods are used in the existing systems to calculate the total weight of the axle [3, 4]: the Basic Method and the Area Method. The following formula is used to calculate the total weight of the axis using the Basic Method:

$$\boldsymbol{W_{ha}} = \boldsymbol{A_t} \cdot \boldsymbol{P_t} \tag{7.2}$$

where W_{ha}—weight on half-axle, A_t—area of the tyre footprint, $P_t \sim V(t)$—air pressure inside the tyre and, according to Newton's 3rd law, it is proportional to the axle weight.

As we can see the exact values of the formula factors are unknown. The area of the tyre footprint is calculated roughly by the length of the output voltage impulse, which, in its turn, depends on the vehicle speed. The Area Method uses the assumption that the area under the recorded impulse curve line, in other words—the integral, characterizes the load on the axle. To calculate the integral, the curve line is approximated by the trapezoid. In this case the smaller the integral—the greater the load. This method does not require knowing the tyre pressure, but it requires the time-consuming on-site calibration. Also, it has to be kept in mind that the time of the tyre crossing the sensor is too small to get an electrical signal of high quality for its further mathematical processing. We use the Area method only for the tyre footprint area definition in (7.2), but the pressure is measured from the signal amplitude.

7.3 Experimental Vehicle Parameters

There was the set of measurement experiments with the roadside FOS sensors on April, 2012 in Riga, Latvia. Loaded truck (Fig. 7.2) was preliminary weighed on the weighbridge with the accuracy <1 %.

Reference weights of the separate axles are given in the Table 7.1. The output signals from FOS sensors for truck speeds 70 and 90 km/h are demonstrated on Fig. 7.3. It is evident that the signals for the different speeds have been changing by amplitude and the proportion of amplitudes does not fit the axle weights (Fig. 7.3).

The reason of this behaviour may be explained by FOS properties such as weight (pressure) distribution along the sensor length as well as sensor non-linearity and temperature dependence.

Fig. 7.2 Experimental truck "Volvo FH12" with full load 36900 kg

Table 7.1 The reference static axle weights

Date: 20.04.2012 (air temperature +12 °C)					
Reference axle weight (tons)	7.296	12.619	5.509	5.641	5.844

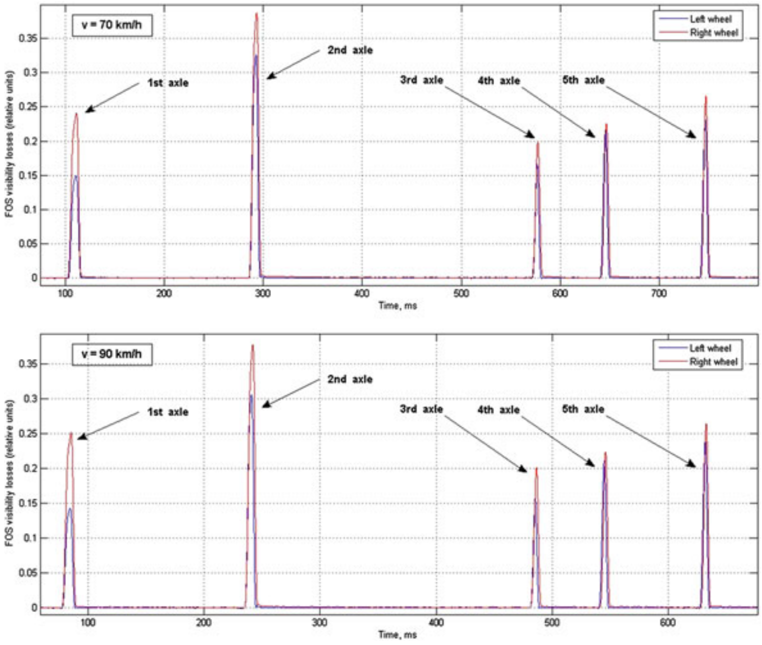


Fig. 7.3 Examples of FOS signals of experimental truck for vehicle speeds 70 and 90 km/h respectively

7.4 Fibre-Optic Sensor Properties

Fibre-optic sensor [8] output light intensity changes due to the applied external vertical force were measured using of the optical interface SL MA-110 that was developed by SensorLine GmbH [8]. Laboratory experiments with varying parameters (temperature, steel plate width and load speed) were made at Latvian University Institute of Polymer Mechanics with electronically controlled compression machine.

The first experiment examined the load characteristic according to the temperature change: FOS was placed into the tube of the soft thermal insulation material in which chilled carbon dioxide was circulating. The load from a compression machine was applied to the sensor through the tube and a 200 × 200 mm square steel plate (Fig. 7.4a). It was found during this experiment that the optical response of the FOS was changing due to the warming. And it is important to notice, that no pressure was applied (Fig. 7.4b).

FOS is permanently installed in the road surface, therefore environment temperature changes affect characteristics of the protective housing rubber (stiffness) and the medium where the light propagates. These changes introduce nonlinear distortions which together with externally applied pressure on a FOS are displayed at Fig. 7.4c. Relations between the load characteristics at the different temperatures

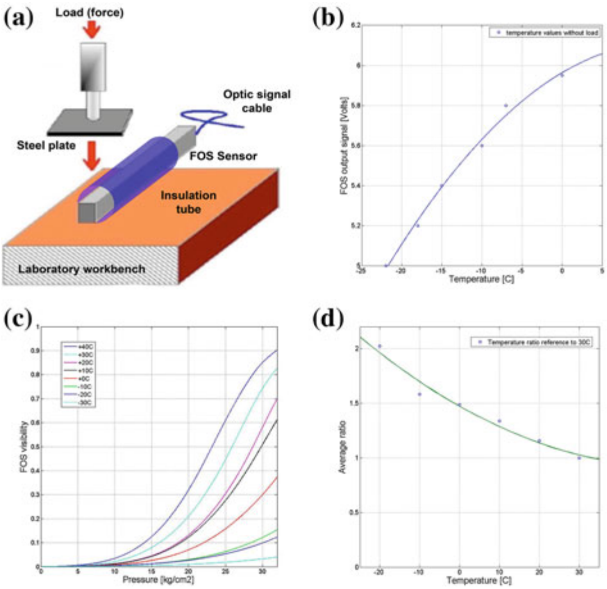


Fig. 7.4 **a** Experimental laboratory equipment scheme. **b** FOS temperature dependence without applying load. **c** FOS load characteristics at different temperatures, and **d** fitted model of FOS various temperature load characteristic ratio values relative to 30 °C

are displayed in the Fig. 7.4d. These relations can be conditionally described by polynomial approximation model:

$$\boldsymbol{LC}_{\mathrm{T[C]}} = \boldsymbol{LC}_{30\mathrm{[C]}} \cdot (\boldsymbol{a}_2 \cdot \boldsymbol{t}^2 + \boldsymbol{a}_1 \cdot \boldsymbol{t} + \boldsymbol{a}_0), \tag{7.3}$$

where $LC_{\mathrm{T[C]}}$ is desired load characteristic at T °C, $\mathrm{LC}_{30[C]}$ is load characteristic at 30 °C and $\mathrm{a}_{2,1,0}$ are coefficients of least square optimisation calculated from Fig. 7.4d.

In the real environment tyre footprint width may vary depending on tyre size and inflation pressure, which will result in the different force redistribution. The second experiment shows this dependence (Fig. 7.5a), these measurements were made at constant temperature 14 °C and constant loading speed 20 mm/s.

Relations between the load characteristics obtained using the different steel plates are displayed in Fig. 7.5b. These relations can be conditionally described by exponential approximation model:

$$\boldsymbol{LC}_{\boldsymbol{W[mm]}} = \boldsymbol{LC}_{\boldsymbol{200[mm]}} \cdot \boldsymbol{a}_1 (1 - \boldsymbol{e}^{\boldsymbol{a}_0/\boldsymbol{W}}), \tag{7.4}$$

where $\mathrm{LC}_{\mathrm{W[mm]}}$ is desired load characteristic with W mm wide plate, $\mathrm{LC}_{200\mathrm{[mm]}}$ is load characteristic with 200 mm wide plate and $\mathrm{a}_{0,1}$ are coefficients of least square optimisation calculated from Fig. 7.5b.

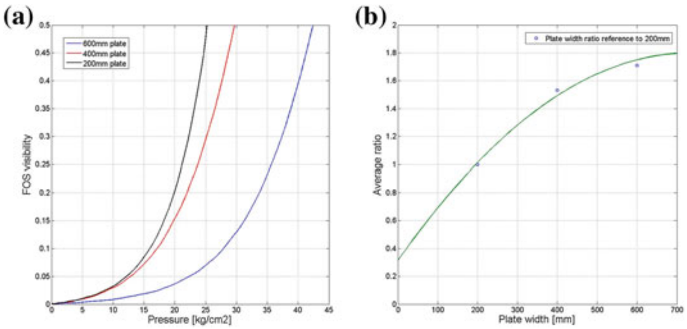


Fig. 7.5 **a** FOS load characteristics with the different steel plate widths, and **b** fitted model of FOS various steel plate width load characteristic ratio values relative to 200 × 200 mm steel plate

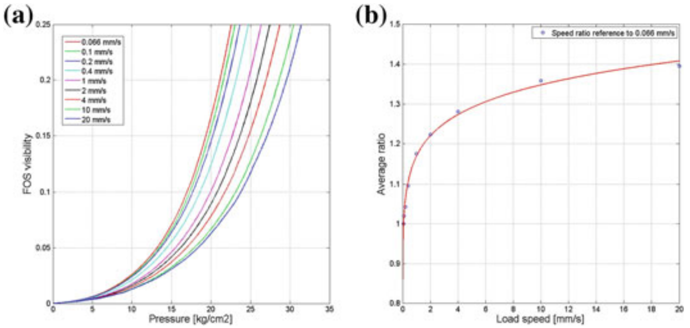


Fig. 7.6 **a** FOS load characteristics at different load speeds, and **b** fitted model of FOS various load speed load characteristic ratio values relative to 0.066 mm/s load speed

The vehicles are crossing the FOS at the different speeds and the sensor reaction is different due to its inertia properties. Therefore the third experiment was dedicated to study FOS output signal dependence on applied force at the different speeds (Fig. 7.6a): these measurements were made at the constant temperature 17 °C and the steel plate size 200 mm. Relations between the load characteristics at the different applied speeds are displayed in Fig. 7.6b. These relations can be conditionally described by power approximation model:

$$\boldsymbol{LC_{S[mm/s]}} = \boldsymbol{LC_{0.066[mm/s]}}(\boldsymbol{a}_1 + \boldsymbol{S}^{\boldsymbol{a}_0}), \tag{7.5}$$

where $\mathrm{LC}_{\mathrm{S[mm/s]}}$ is desired load characteristic at S mm/s, $\mathrm{LC}_{0.066[\mathrm{mm/s}]}$ is load characteristic at 0.066 mm/s and $\mathrm{a}_{0,1}$ are coefficients of least square optimisation calculated from Fig. 7.6b.

7.5 Tyre Footprint and Weight Estimation

As it is clearly seen from the expression (7.2), the A$_t$ area of the tyre footprint should be known to calculate the axle weight by the registered FOS signal [1]. The form of the signal is non-symmetric and sufficiently distorted by the rolling process of the wheel on the road surface (see Fig. 7.7a).

One of the possible explanations of the signal waveform distortion is the idea about the common interaction of two factors [2]: vertical dead weight gravity of conditionally immovable wheel (according to wheel geometry it must be symmetric, see Fig. 7.7b), and the force of friction (it depends on the pavement and tyre properties, wheel's speed and weight, and its expected waveform is asymmetric, see Fig. 7.7c). The problem of the decomposition of the non-symmetric signal in 2 parts (symmetric and asymmetric) can be solved by the polynomial approximation using the least square method and the further grouping of members with even and odd powers separately or by standard even-odd decomposition of the signal on finite window [5, 10].

On the other hand, by assumption that the vehicle moves uniformly and all forces maximally compensate each other, we can accept that the friction force is as minimal as possible (rolling friction only without sliding friction). It is possible to minimize

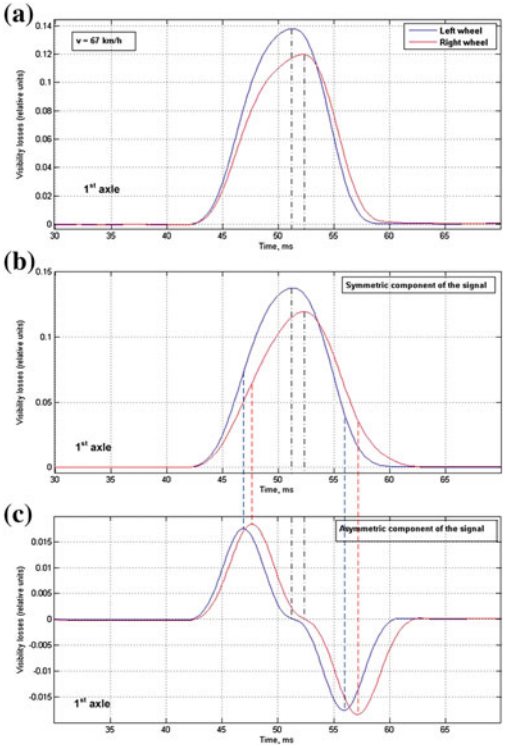


Fig. 7.7 **a** FOS output signal in the form of visibility losses (formula 7.1). **b** Approximated vertical weight component (symmetric), and (c) approximated asymmetric component depending on horizontal velocity and friction [2]

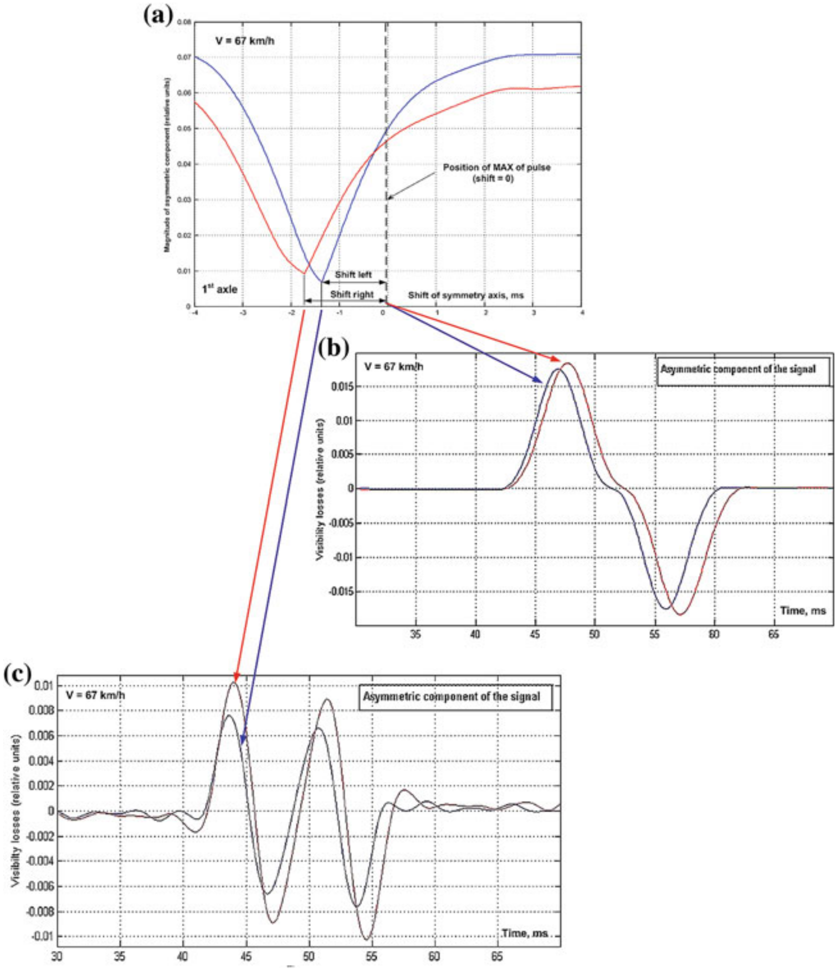


Fig. 7.8 Symmetry axis shift optimisation: **a** approximated asymmetric component magnitude as the function on symmetry axis shift from the maximum of the pulse, **b** asymmetric (friction) component waveform if the shift is equal 0, **c** the same if the shift responds to minimum of friction force

the friction component magnitude moving the axis of symmetry before the pulse (see Fig. 7.8). This dependence is obtained by calculation of the maximum (magnitude) of asymmetric component of the signal changing the shift between maximum of initial pulse form and location of symmetry axis. It will be minimal at the position conditionally named as the "mass centre" of the pulse (see Fig. 7.9a).

The waveform of the friction component on Fig. 7.9b sufficiently differs from the same on Fig. 7.7c. Two maximums and two minimums clearly locate the

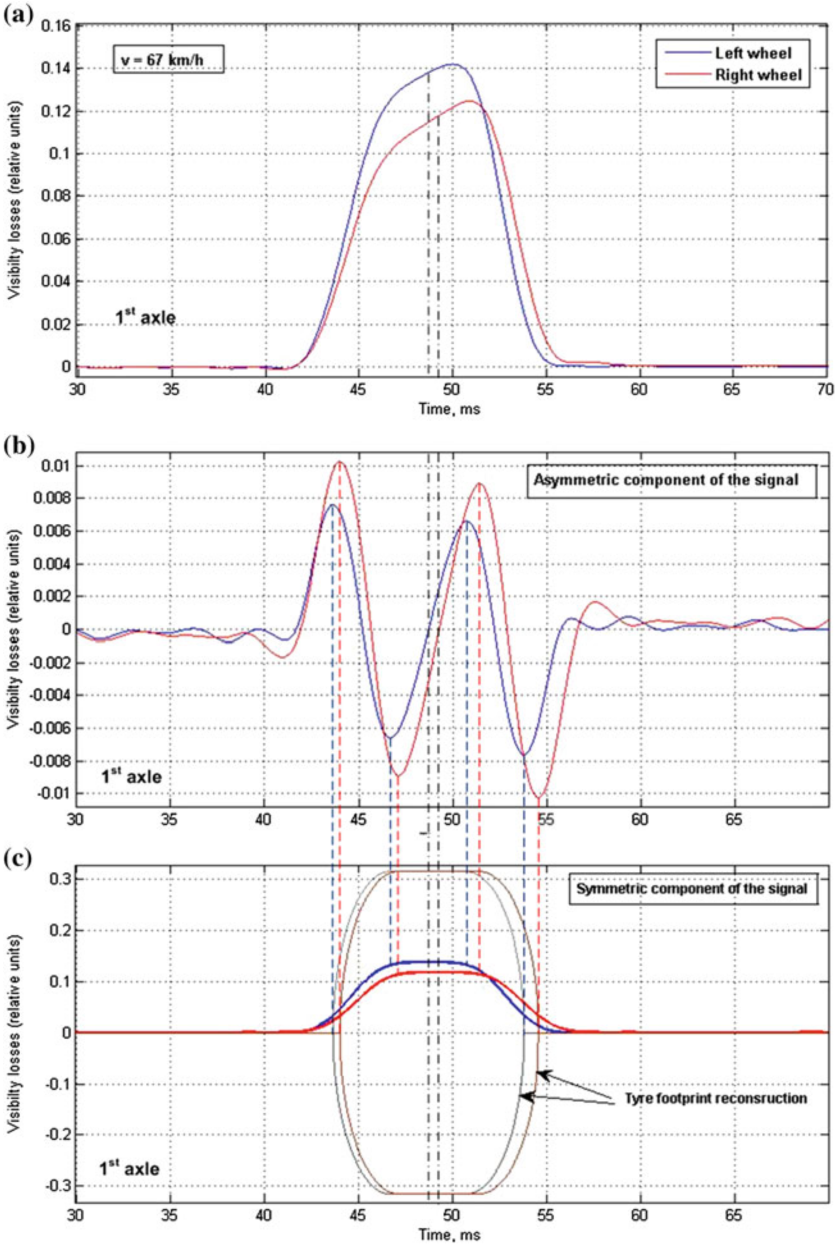
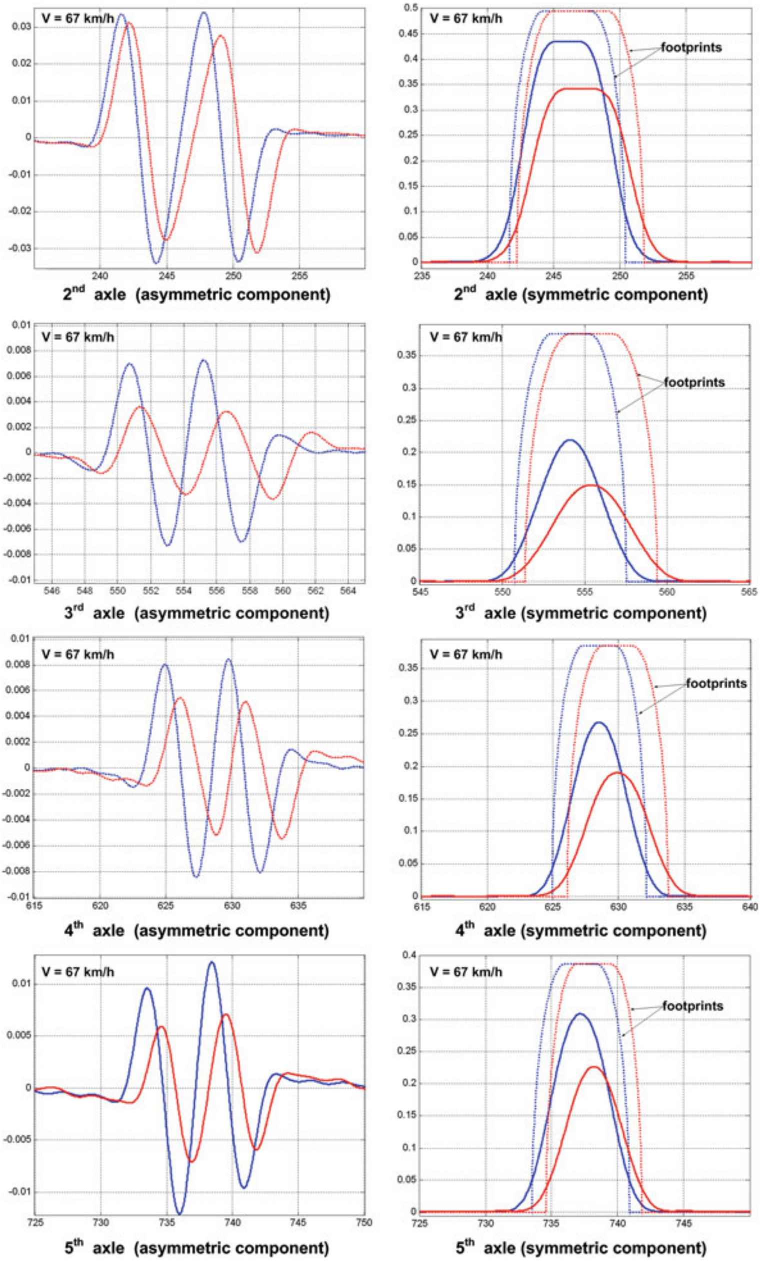


Fig. 7.9 **a** FOS output signal in the form of visibility losses in dimensionless units. **b** Approximated vertical weight component (symmetric). **c** Approximated asymmetric component and tire footprint reconstruction in case of minimal friction condition



2nd axle (asymmetric component)

2nd axle (symmetric component)

3rd axle (asymmetric component)

3rd axle (symmetric component)

4th axle (asymmetric component)

4th axle (symmetric component)

5th axle (asymmetric component)

5th axle (symmetric component)

Fig. 7.10 The results of axle's signals decomposition on symmetric (from *right*) and asymmetric (from *left*) components for 2nd, 3rd, 4th and 5th axle and the form of footprint reconstruction respectively

Table 7.2 The results of the axle weight estimation and errors for the different measurements at the speeds 10–90 km/h (two sets of measurements per speed bin)

Date: 20.04.2012 (air temperature +12 °C)						
	Etalon axle's weight (tons)	7.296	12.619	5.509	5.641	5.844
Speed: 10 km/h						
No	Parameter:	1st axle	2nd axle	3rd axle	4th axle	5th axle
1	Axle's weight (tons)	7.3392	12.7267	4.7930	5.0592	5.8173
	Error (%)	0.589	0.855	−12.994	−10.317	−0.453
2	Axle's weight (tons)	7.2190	14.0050	4.6014	5.1721	6.5539
	Error (%)	−1.058	10.985	−16.472	−8.316	12.152
Speed: 20 km/h						
1	Axle's weight (tons)	7.1375	12.9337	5.3205	6.2553	6.5012
	Error (%)	−2.176	2.496	−3.417	10.885	11.251
2	Axle's weight (tons)	7.2929	13.5535	4.4927	4.8372	5.6539
	Error (%)	−0.046	7.408	−18.445	−14.253	−3.248
Speed: 50 km/h						
1	Axle's weight (tons)	7.5824	11.9901	5.3806	5.4392	5.5499
	Error (%)	3.921	−4.982	−2.327	−3.582	−5.029
2	Axle's weight (tons)	7.4452	12.4781	5.1152	5.3775	5.6148
	Error (%)	2.042	−1.114	−7.144	−4.676	−3.918
Speed: 70 km/h						
1	Axle's weight (tons)	7.5129	12.1787	5.2473	5.6684	5.6978
	Error (%)	2.969	−3.487	−4.746	0.482	−2.497
2	Axle's weight (tons)	7.4754	12.4669	5.2623	5.7398	5.8544
	Error (%)	2.455	−1.204	−4.474	1.747	0.183
Speed: 90 km/h						
1	Axle's weight (tons)	7.3327	12.8335	5.0597	5.3737	5.6183
	Error (%)	0.499	1.702	−8.152	−4.744	−3.859
2	Axle's weight (tons)	7.5990	12.0713	5.1834	5.6570	5.8492
	Error (%)	4.149	−4.338	−5.907	0.278	0.094

characteristic points for the tire footprint estimation in the elliptic approximation (see Fig. 7.9c).

Now the problem of the tyre footprint area estimation may be solved. Multiplying the impulse length by the speed of the wheel we can calculate the length of the footprint. In the considered examples (Figs. 7.7 and 7.9) it is $L_{left} = 0.1905$ m and $L_{right} = 0.1976$ m. It agrees with another data for the wheel R22.5 and tyre width of 315 mm for the 1st axle.

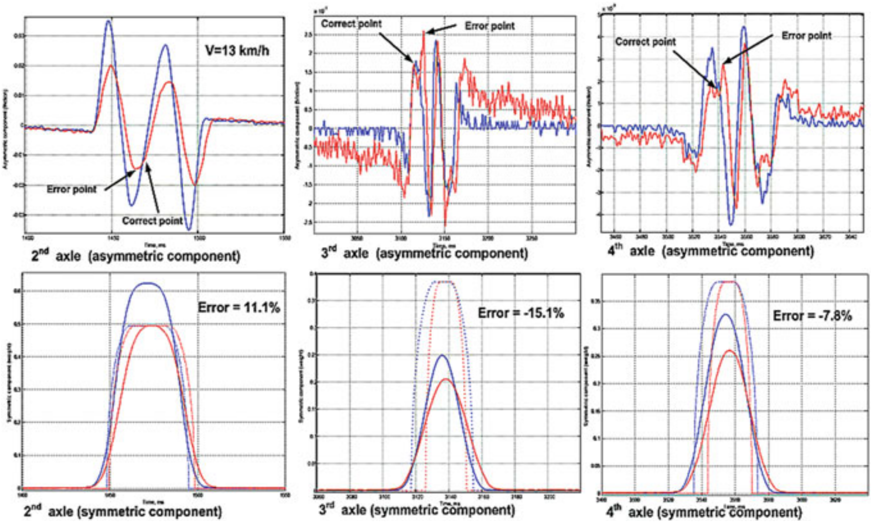
Applying above mentioned approach to another vehicle's axles and wheels we can estimate the length and form of each tyre footprint. Of course, the width of 2nd axle's tyre we consider as double (double wheels for Volvo FH12 vehicle's 2nd (motor) axle).

Full data of footprint lengths for experimental cargo vehicle (see Figs. 7.2, 7.9, 7.10 and Table 7.1) are the following: left side wheels lengths are L_{L12345} = {0.1905 0.1652 0.1278 0.1354 0.1379} m, and right side wheels lengths are L_{R12345} = {0.1976 0.1799 0.1509 0.1449 0.1356} m. The difference between the lengths of each wheel can be explained by the fact that the 2nd axle has the double-wheel, the trailer tyres but (axles No 3–5) width is 385 mm.

Applying the sensors signal processing algorithm together with the of estimation of the tyre footprint dynamic area (on the sensor's surface) and approximation of the nonlinear characteristics of the FOS (Figs. 7.4, 7.5 and 7.6) for the suitable range of temperatures, it is possible to estimate the following weights of axles (Table 7.2).

As it can be seen from the Table 7.2, the most preferred for the measurements are the velocity ranges from 50 km/h and above: the measurement errors of the axle loads do not exceed 10 %, which is consistent with the problem of the pre-selection of the overloaded vehicles. The dynamics of vehicle breaking or acceleration is more visible at the low velocities (see Fig. 7.11). It can seriously distort the waveform of asymmetric friction component and change characteristic points for tyre footprint estimation. The source of the relatively big measurement errors at the low velocities are the distortion and footprint area reconstruction errors because of the vertical oscillations of the dynamic motion of the vehicle, whose amplitudes are smaller at the higher speeds.

Taking into account the properties of each individual sensor, the calibration of FOS should be conducted twice: at first in the laboratory (load characteristics in the temperature range from −20 to + 30 °C), and, secondly, after installing the sensor in the road surface using the vehicles with allowed but known weight.



2nd axle (asymmetric component)

3rd axle (asymmetric component)

4th axle (asymmetric component)

2nd axle (symmetric component)

3rd axle (symmetric component)

4th axle (symmetric component)

Fig. 7.11 The results of axle's signals transformation on symmetric (gravity) and asymmetric (friction) components for 2nd, 3rd, and 4th axle contained the errors of footprint form reconstruction due to dynamic oscillations in the waveform of asymmetric component

7.6 Conclusions

Fibre-optic sensors (FOS) are mainly used as the vehicle detectors because of the complicated dependence of a set of factors (sensor's surface temperature, area of impact (vehicle's tyre width), the speed of loading, and vehicle velocity). The set of input parameters made relatively problematic the task of weigh-in-motion using FOS. The results of the present research demonstrate that the factors mostly impacting the FOS measurement accuracy can be investigated and included into the axle weight calculations.

An idea to normalize the FOS output voltage by the sensor visibility losses (changing from 1 to 0) parameter helps to avoid the influence of the static voltage source instability as well as the conditions of sensor installation into the pavement. Each instantaneous measured value of the rolling wheel is independent here from the output voltage for the unloaded sensor. Also, the static part of the temperature dependence is compensated by this way.

A novice approach to decompose each wheel response into the gravity (symmetric) and the rolling friction (asymmetric) components near the "mass center" of the pulse, leads to the possibility of tyre footprint area estimation and weight calculation based on mixed Basic and Area method. Preliminary results of the proposed method for WIM using FOS demonstrates the accuracy of measurements are in range of less than 10 % of the measured weight. It is sufficient for the problem of overloaded vehicles pre-selection.

The experimental results show that the range of the vehicles velocity from 50 to 90 km/h seems more appropriate for WIM based on fibre-optic sensors. From the authors point of view, using the additional signal processing efforts, it is possible to achieve the consistent accuracy level not only at the high speeds (above 50 km/h), but also at the low speeds (10–50 km/h). We mean B+(7) according to COST 323 for the high speeds and D2 according to OIML R134 for the low speeds [7].

Acknowledgments This research was granted by ERDF funding, project "Fiber Optic Sensor Applications for Automatic Measurement of the Weight of Vehicles in Motion: Research and Development (2010–2013)", No. 2010/0280/2DP/2.1.1.1.0/10/APIA/VIAA/094, 19.12.2010.

References

1. Batenko, A., Grakovski, A., Kabashkin, I., Petersons, E., Sikerzhicki, Y.: Weight-in-motion (WIM) measurements by fiber optic sensor: problems and solutions. Transp. Telecommun. **12**(4), 27–33 (2011)
2. Krasnitsky, Y.: Transient response of a small-buried seismic sensor. Comput. Model. New Technol. **16**(4), 33–39 (2012)
3. Malla, R.B., Sen, A., Garrick, N.W.: A special fiber optic sensor for measuring wheel loads of vehicles on highways. Sensors **8**, 2551–2568 (2008)
4. McCall, B., Vodrazka Jr, W.: States' successful practices weigh-in-motion handbook. Federal Highway Administration, Washington, DC (1997)

5. Mesco, A.: Digital filtering: applications in geophysical exploration for oil. Academiai Kiado, Budapest (1984)
6. Mimbela, L.-E.Y., Pate, J., Copeland, S., Kent, P. M., Hamrick, J.: Applications of fiber optic sensors in weigh-in-motion (WIM) systems for monitoring truck weights on pavements and structures. Final report on research project, 158 p. New Mexico State University, Las Cruces (2003)
7. O'Brien, E. J., Jacob, B.: European specification on vehicle weigh-in-motion of road vehicles. In: Proceedings of the 2nd European Conference on Weigh-in-Motion of Road Vehicles, 171–183. Office for Official Publications of the European Communities, Luxembourg (1998)
8. SENSORLINE GmbH.: SPT short feeder spliceless fiber optic traffic sensor: product description. Retrieved 7 Jan 2011, from http://sensorline.de/home/pages/downloads.php (2010) (Sensor Line)
9. Teral, S.R.: Fiber optic weigh-in-motion: looking back and ahead. Opt Eng **3326**, 129–137 (1998)
10. Vinay, K.I., Proakis, J.G.: Digital signal processing using MATLAB. Thomson Learning (2006)

Part II
Robotics and Automation

Chapter 8
Sliding Mode Control with Integral Action for Slip Suppression of Electric Vehicles

Tohru Kawabe

Abstract This paper proposes a new control method based on the SMC (sliding mode control) with integral action for traction control of EVs (electric vehicles). The proposed method is the combination of the SMC and the integral action, and it's able to improve the maneuverability, the stability and the low energy consumption of EVs. The effectiveness of the method is demonstrated via numerical examples.

Keywords Integral action · Sliding mode control · Slip suppression · Electric vehicle

8.1 Introduction

Electrical vehicles (EV) have received much attention in recent years as a countermeasure to global warming and for being Eco friendly [1–4]. EVs are automobiles which are propelled by electric motors, using electrical energy stored in batteries or another energy storage devices. Electric motors have several advantages over (internal-combustion engines) ICEs:

(a) Energy Efficient.
Electric motors convert 75 % of the chemical energy from the batteries to power the wheels—ICEs only convert 20 % of the energy stored in gasoline.

(b) Environmentally Friendly.
EVs emit no tailpipe pollutants, although the power plant producing the electricity may emit them. Electricity from wind-, solar-, or hydro-powered plants causes no air pollutants.

(c) Performance Benefits.
Electric motors provide quiet, smooth operation and stronger acceleration and require less maintenance than ICEs.

T. Kawabe (✉)
Faculty of Engineering, Information and Systems, University of Tsukuba,
Tsukuba 305-8573, Japan
e-mail: kawabe@cs.tsukuba.ac.jp
http://tk2.cs.tsukuba.ac.jp/~kawabe

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_8

(d) Reduce Energy Dependence.
Electricity is a domestic energy source.

The travel distance per charge for EV has been increased through battery improvements and using regeneration brakes, and attention has been focused on improving motor performance. The following facts are viewed as relatively easy ways to improve maneuverability and stability of EVs.

1. The input/output response is faster than for gasoline/diesel engines.
2. The torque generated in the wheels can be detected relatively accurately
3. Vehicles can be made smaller by using multiple motors placed into the wheels.

Much research has been done on the stability of general automobiles, for example, Anti-lock-Braking Systems (ABS), Traction-Control-Systems (TCS), and Electric-Stability-Control (ESC) [5] as well as Vehicle-Stability-Assist (VSA) [6] and All-Wheel-Control (AWC) [7]. What all of these have in common is that they maintain a suitable tire grip margin and reduce drive force loss to stabilize the vehicle behavior and improve drive performance. With gasoline/diesel engines, however, the response time from accelerator input until the drive force is transmitted to the wheels is slow and it is difficult to accurately determine the drive torque, which limits the vehicle's control performance.

This paper deals with the traction control of EV for slip suppression during driving. Conventional gasoline/diesel vehicles are equipped with a TCS, which requires expensive sensors and additional equipment, but, as mentioned above, EV have a fast torque response and the motor characteristics can be used to accurately determine the torque, which makes it relatively easy and inexpensive to realize high-performance traction control. This is expected to improve the maneuverability and stability of EV. Increasing use of EV in the future, EVs are expected to be equipped with TCS as standard. It's, therefore, important to research and development to achieve high-performance EV traction control.

When the vehicle is starting off or accelerating, particularly on a slippery or wet road surface, the wheel spins easily, which causes unstable driving situation and large waste of energy. Therefore, it's important to keep the optimal driving force in all driving situation for motion stability and saving energy. During acceleration, the driving force of wheel directly depends on the friction coefficient between road and tire, which is in accordance with the wheel slip and road conditions. For this reason, it becomes possible to give the adequate driving force by controlling the wheel traction.

Several methods have been proposed for the traction control by using slip ratio of EVs [8–10], such as the method based on Model Following Control (MFC) in [11] and Model Predictive PID method (MP-PID) in [12]. Both of these methods show good performances under the nominal conditions where the situation, for example, mass of vehicle, road condition, and so on, is not changed. To meet the high performance even variation happened in the conditions, it is significant to construct the robust method against the situation changing. About this point, Sliding Mode Control (SMC) has been performed good robustness for the systems with uncertainties or nonlinearities.

However, for slip ratio control with the conventional SMC, the control performance will get degradation due to the chattering which always occurs because of switching the control inputs due to the structure of SMC. To overcome such disadvantages of conventional SMC method, new SMC method with introducing the integral term to the design of the sliding surface in order to get better control performance and save more energy for slip suppression of EVs with changing the mass of vehicle and road condition is proposed. The numerical examples show the effectiveness of the proposed method.

8.2 Preliminaries of SMC

Consider the single input nonlinear system [13]

$$x^{(n)} = f(x) + b(x)u \tag{8.1}$$

where $x = \left[x\ \dot{x} \ldots x^{(n-1)}\right]^T$ is the state vector and u is the control input. In general, the function $f(x)$ and the control gain $b(x)$ are not exactly known but the extents of the imprecision on $f(x)$ and $b(x)$ are known by their upper bounds. The control problem is to seek a solution that is robust to uncertainties in $f(x)$ and $b(x)$. Firstly, we defined a time-varying surface $s(x;t)$ in the state space $R^{(n)}$ by

$$s(x;t) = \left(\frac{d}{dt} + \alpha\right)^{n-1} \tilde{x} = 0, \quad \alpha > 0 \tag{8.2}$$

where $\tilde{x} = x - x^* = \left[\tilde{x}\ \dot{\tilde{x}} \ldots \tilde{x}^{(n-1)}\right]^T$ is the error between the output state x and the desired state x^*. The problem of tracking $x = x^*$ is equivalent to remaining on the surface s for all $t > 0$. When $s = 0$, that is to say, the output state reaches the surface which represents the error is zero. Here, $s = 0$ is called sliding surface. On this surface the error will converge to zero exponentially. When $\dot{s} = 0$, the state is controlled to slide on the sliding surface, which is described that the system is in sliding mode.

The SMC law contains two parts, the equivalent control u_{eq} and the hitting control u_{ht}, which is defined as follows,

$$u = u_{eq} + u_{ht}. \tag{8.3}$$

u_{eq} can be interpreted as the continuous control law which would maintain $\dot{s} = 0$ when the dynamics are exactly known. When the dynamics are not exactly known, such as the uncertainties occur in the system or the state of system is off the sliding surface, u_{ht} acts to bring the state back to the sliding surface and keeps it in sliding mode. Generally, u_{ht} uses a discontinuous function to realize the switching action on sliding surface.

For choosing the control input u, it is necessary to consider the sliding condition [14], which is defined as

$$\frac{1}{2}\frac{d}{dt}s^2 \leq -\eta|s| \tag{8.4}$$

where $\eta > 0$. From Eq. (8.4) , s^2 shows that the squared "distance" to the sliding surface, which decreases along all system trajectories. Particularly, once the states reach the surface, the system trajectories remain on the surface. In other words, satisfying the sliding condition makes the trajectories reach the surface in finite time, and once on the manifold, it cannot leave it. Furthermore, Eq. (8.4) also implies that some dynamic uncertainties can be tolerated while still keeping the surface an invariant set.

To realize the concept of SMC, we always need to follow two steps:

[**Step 1.**] Design a sliding surface s which is invariant of the controlled dynamics.
[**Step 2.**] Choose the control input u which drives the states to the sliding surface in sliding mode in finite time.

8.3 Electric Vehicle Dynamics

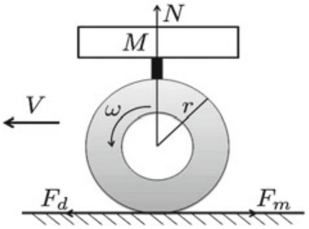
As a first step toward practical application, this paper restricts the vehicle motion to the longitudinal direction and uses direct motors for each wheel to simplify the one-wheel model to which the drive force is applied. In addition, braking was not considered this time with the subject of the study being limited to only when driving.

From Fig. 8.1, the vehicle dynamical equations are expressed as Eqs. (8.5–8.7).

$$M\frac{dV}{dt} = F_d(\lambda) - F_a - \frac{T_r}{r} \tag{8.5}$$

$$J\frac{d\omega}{dt} = T_m - rF_d(\lambda) - T_r \tag{8.6}$$

Fig. 8.1 One-wheel car model



$$F_m = \frac{T_m}{r} \tag{8.7}$$

$$F_d = \mu(c, \lambda)N \tag{8.8}$$

where M is the vehicle weight, V is the vehicle body velocity, F_d is the driving force, J is the wheel inertial moment, F_a is the resisting force from air resistance and other factors on the vehicle body, T_r is the frictional force against the tire rotation, ω is the wheel angular velocity, T_m is the motor torque, F_m is the motor torque force conversion value, r is the wheel radius, and λ is the slip ratio. The slip ratio is defined by (8.9) from the wheel velocity (V_ω) and vehicle body velocity (V).

$$\lambda = \begin{cases} \dfrac{V_\omega - V}{V_\omega} & \text{(accelerating)} \\ \dfrac{V - V_\omega}{V} & \text{(braking)} \end{cases} \tag{8.9}$$

λ during accelerating can be shown by (8.10) from Fig. 8.1.

$$\lambda = \frac{r\omega - V}{r\omega} \tag{8.10}$$

The frictional forces that are generated between the road surface and the tires are the force generated in the longitudinal direction of the tires and the lateral force acting perpendicularly to the vehicle direction of travel, and both of these are expressed as a function of λ. The frictional force generated in the tire longitudinal direction is expressed as μ, and the relationship between μ and λ is shown by (8.11) below, which is a formula called the Magic-Formula [15] and which was approximated from the data obtained from testing.

$$\mu(\lambda) = -c_{road} \times 1.1 \times (\mathrm{e}^{-35\lambda} - \mathrm{e}^{-0.35\lambda}) \tag{8.11}$$

where c_{road} is the coefficient used to determine the road condition and was found from testing to be approximately $c_{road} = 0.8$ for general asphalt roads, approximately $c_{road} = 0.5$ for general wet asphalt, and approximately $c_{road} = 0.12$ for icy roads. For the various road conditions ($0 < c < 1$), the $\mu - \lambda$ surface is shown in Fig. 8.2. It shows how the friction coefficient μ increases with slip ratio λ ($0.1 < \lambda < 0.2$) where it attains the maximum value of the friction coefficient. As defined in (8.8), the driving force also reaches the maximum value corresponding to the friction coefficient. However, the friction coefficient decreases to the minimum value where the wheel is completely skidding. Therefore, to attain the maximum value of driving force for slip suppression, it should be controlled the optimal value of slip ratio. the optimal value of λ is derived as follows.

Choose the function $\mu_c(\lambda)$ defined as

$$\mu_c(\lambda) = -1.1 \times (\mathrm{e}^{-35\lambda} - \mathrm{e}^{-0.35\lambda}). \tag{8.12}$$

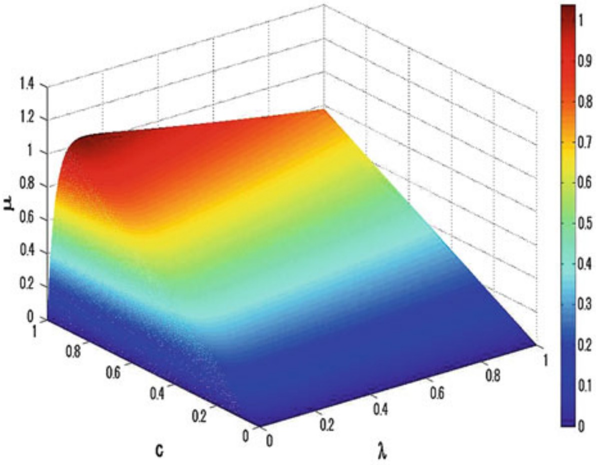


Fig. 8.2 λ-μ surface for road conditions

By using (8.11) and (8.12) can be rewritten as

$$\mu(c, \lambda) = c_{road} \cdot \mu_c(\lambda). \tag{8.13}$$

Evaluating the values of λ which maximize $\mu(c, \lambda)$ for different $c(c > 0)$, means to seek the value of λ where the maximum value of the function $\mu_c(\lambda)$ can be obtained. Then let

$$\frac{d}{d\lambda}\mu_c(\lambda) = 0 \tag{8.14}$$

and solving equation (8.14) gives

$$\lambda = \frac{\log 100}{35 - 0.35} \approx 0.13. \tag{8.15}$$

Thus, for the different road conditions, when $\lambda \approx 0.13$ is satisfied, the maximum driving force can be gained. Namely, from (8.11) and Fig. 8.2, we find that regardless of the road condition (value of c), the $\lambda - \mu$ surface attains the largest value of μ when λ is the optimal value 0.13.

8.4 Sliding Mode Control with Integral Action Method for Slip Suppression

In this section, for slip suppression of EVs, the proposed control strategy based on SMC with introducing the integral term is explained. Without loss of generality, one wheel model mentioned above is used for design of the control laws. The system dynamics can be written as

$$\dot{\lambda} = f + bT_m \tag{8.16}$$

where $\lambda \in R$ is the state of system representing the slip ratio of driven wheel which is defined as Eq. (8.10) for the case of acceleration. T_m is the control input.

Differentiating Eq. (8.10) with respect to time

$$\dot{\lambda} = \frac{-\dot{V} + (1-\lambda)\dot{V_w}}{V_w} \tag{8.17}$$

and substituting Eqs. (8.5), (8.6) and (8.8) into Eq. (8.17), the following equations can be attained,

$$f = -\frac{g}{V_w}\left[1 + (1-\lambda)\frac{r^2 M}{J_w}\right]\mu(c, \lambda) \tag{8.18}$$

$$b = \frac{(1-\lambda)r}{J_w V_w}. \tag{8.19}$$

The control objective is to control the value of the slip ratio to the constant reference value λ^*.

Actually, the mass of vehicle often changes with the number of passengers and the weight of luggage. Besides, the vehicle has to always travel on many kinds of road surfaces. As a result, the controller needs to perform much robustly with the uncertainties happened in the mass of vehicle and road surface conditions which are represented by M and c respectively. The ranges of variation of M and c are set as

$$M_{min} \leq M \leq M_{max} \tag{8.20}$$

$$c_{min} \leq c \leq c_{max}. \tag{8.21}$$

Consider the system Eq. (8.16), the nonlinear function f is not exactly known, but it can be estimated as $\hat{f}$. The estimation error on f is assumed to be bounded by a known function $F = F(\lambda)$,

$$\left|\hat{f} - f\right| \leq F. \tag{8.22}$$

The uncertainty in f is due to the parameter M and c. Accordingly, by using Eq. (8.18) the estimation of f can be defined as

$$\hat{f} = -\frac{g}{V_w}\left[1 + (1-\lambda)\,\frac{r^2\hat{M}}{J_w}\mu\left(\hat{c},\lambda\right)\right] \tag{8.23}$$

where $\hat{M}$ is the estimated value of M and $\hat{c}$ is estimated for c.

Here, we define the estimated values of these parameters respectively by using the arithmetic mean of the value of the bounds as

$$\hat{M} = \tfrac{M_{min}+M_{max}}{2} \tag{8.24}$$

$$\hat{c} = \tfrac{c_{min}+c_{max}}{2}. \tag{8.25}$$

From these definitions, the error in estimation can be given by

$$\begin{aligned}\left|f-\hat{f}\right| \leq \frac{g}{|V_w|}\Big\{&\left|\mu\left(c_{max},\lambda\right)-\mu\left(\hat{c},\lambda\right)\right| \\ &+(1-\lambda)\frac{r^2}{J_w}\left|M_{max}\mu(c_{max},\lambda)-\hat{M}\mu(\hat{c},\lambda)\right|\Big\}.\end{aligned} \tag{8.26}$$

Then, let

$$\begin{aligned}F = \frac{g}{|V_w|}\Big\{&\left|\mu(c_{max},\lambda)-\mu(\hat{c},\lambda)\right| \\ &+(1-\lambda)\frac{r^2}{J_w}\left|M_{max}\mu(c_{max},\lambda)-\hat{M}\mu(\hat{c},\lambda)\right|\Big\}.\end{aligned} \tag{8.27}$$

8.4.1 Design of Sliding Surface

Letting $\tilde{\lambda}$ be the variable of interest, then the order of system is assumed to be one. The sliding function of conventional SMC can be given by

$$s_c(\lambda,t) = \tilde{\lambda} \tag{8.28}$$

where $\tilde{\lambda}$ is the error between the actual slip ratio and the reference value, which is defined as $\tilde{\lambda} = \lambda - \lambda^*$.

By adding an integral item to the sliding function s_c, the new sliding function s can be designed as

$$s(\lambda, t) = \tilde{\lambda} + K_i \int_0^t \tilde{\lambda}(\tau) d\tau \tag{8.29}$$

where K_i is the integral gain, $K_i > 0$.

8.4.2 Derivation of Control Law

In this section, the sliding mode controller is derived to make the slip ratio λ to track the reference slip ratio λ^*. The sliding mode occurs when the state λ reaches the sliding surface defined by $s = 0$. The dynamics of sliding mode is governed by

$$\dot{s} = 0. \tag{8.30}$$

Differentiating Eq. (8.29) and substituting the result into Eq. (8.30) give

$$\left(\dot{\lambda} - \dot{\lambda^*}\right) + K_i \left(\lambda - \lambda^*\right) = 0. \tag{8.31}$$

The reference slip ratio λ^* is a constant, thus $\dot{\lambda^*} = 0$. Substituting Eq. (8.16) into Eq. (8.31) gives

$$f + bT_m + K_i(\lambda - \lambda^*) = 0 \tag{8.32}$$

and solving eq. (8.32) gives equivalent control input as

$$T_{meq} = \frac{1}{b}\left[-f - K_i \left(\lambda - \lambda^*\right)\right] \tag{8.33}$$

then the estimate of the equivalent control input can be obtained as

$$\hat{T}_{meq} = \frac{1}{b}\left[-\hat{f} - K_i \left(\lambda - \lambda^*\right)\right]. \tag{8.34}$$

For satisfying sliding condition (make state in the sliding mode) despite uncertainty on the dynamics f, the hitting control input is defined as

$$T_{mht} = \frac{1}{b}\left[-K \operatorname{sgn}(s)\right] \tag{8.35}$$

where

$$\operatorname{sgn}(s) = \begin{cases} -1 & s < 0 \\ 0 & s = 0 \\ 1 & s > 0 \end{cases} \tag{8.36}$$

and K is called sliding gain. Thus, the control law can be given by

$$\begin{aligned} T_m &= \hat{T}_{meq} + T_{mht} \\ &= \frac{1}{b}\Big[-\hat{f} - K_i(\lambda - \lambda^*) - K\,\mathrm{sgn}(s)\Big]. \end{aligned} \tag{8.37}$$

When no uncertainty in the system (i.e., no variation in c and M), T_{mht} is desired to be 0. Because Eq. (8.37) contains the estimate of the equivalent control $\hat{T}_{meq}$, T_m keeps the state on the sliding surface ($s = 0$ i.e., $\lambda = \lambda^*$). Because of the uncertainties in the system, the state λ could deviate from the sliding surface. The hitting control acts to return the state back to the sliding surface which implies the robustness of SMC.

Here, the sliding gain K is chosen as

$$K = F + \eta \tag{8.38}$$

with the value of F given by Eq. (8.27).

Then choose a Lyapunov function as

$$V = \frac{1}{2}s^2 \tag{8.39}$$

and differentiate Eq. (8.39) with respect to time, that gives

$$\dot{V} = \frac{1}{2}\frac{d}{dt}s^2 = s\dot{s}. \tag{8.40}$$

Substituting Eqs. (8.16), (8.30), (8.34), (8.35) and (8.37) into Eq. (8.40) yields

$$\begin{aligned} \dot{V} &= s\dot{s} \\ &= s\Big[f - \hat{f} - K\,\mathrm{sgn}(s)\Big] = s(f - \hat{f}) - K|s| \\ &\le F|s| - K|s| \\ &\le -\eta|s|. \end{aligned} \tag{8.41}$$

Thus, the control law introduced in Eq. (8.37) can guarantee the stability of the system in the Lyapunov sense under variations. Concretely, the stability of the system is guaranteed with an exponential convergence once the sliding surface is encountered, if the sliding condition is satisfied. So Eq. (8.41) guarantees the strategy can converge to the sliding surface in finite time if the error is not zero, that is to say, slip ratio can be controlled to the reference value in finite time whenever the uncertainties occur in the system.

8.4.3 Chattering Reduction

For sliding mode control design, the switched controller limits switching to a finite frequency, which produces chattering. To reduce the chattering, the hitting control T_{mht} can be rewritten by using the saturation function

$$T_{mht} = \frac{1}{b}\left[-K \,\mathrm{sat}\left(\frac{s}{\Phi}\right)\right] \tag{8.42}$$

where $\Phi > 0$ is a design parameter representing the width of the boundary layer around the sliding surface $s = 0$ and the saturation function is defined as

$$\mathrm{sat}\left(\frac{s}{\Phi}\right) = \begin{cases} -1 \; s < -\Phi \\ \frac{s}{\Phi} \; -\Phi \leq s \leq \Phi \\ 1 \; s > \Phi \end{cases} . \tag{8.43}$$

Thus, using Eqs. (8.37), (8.38) and (8.42), the control law of the system by the proposed SMC can be rewritten as

$$T_m = \frac{1}{b}\left[-\hat{f} - K_i\left(\lambda - \lambda^*\right) - (F + \eta)\mathrm{sat}\left(\frac{s}{\Phi}\right)\right]. \tag{8.44}$$

8.5 Numerical Examples

This section shows the numerical simulation results to demonstrate the effectiveness of the proposed method. In the all simulations, the width of the boundary layer Φ defied in Eq. (8.42) is set to 1. In Eq. (8.44), the proposed SMC law can be calculated with the values of design parameters K_i and η, which both impact on the steady state accuracy. Here, for confirm the energy conservation performance of the proposed method, the values of both parameters are set $K_i = 10$ and $\eta = 5$, which are determined by several tests.

By using Eqs. (8.28), (8.30), (8.35) and (8.38), the control law of the conventional SMC can be derived as

$$T_{mc} = \frac{1}{b}\left[-\hat{f} - (F + \eta)\mathrm{sat}\left(\frac{s}{\Phi}\right)\right]. \tag{8.45}$$

In the conventional SMC, the parameters $\eta = 1$ and $\Phi = 1$. The value of parameters used in the simulations are listed in Table 8.1.

As the input to the simulation of system, the torque is produced by the constant pressure on the accelerator pedal, which is decided on the vehicle speed desired by the driver. Here, the vehicle speed is desired to achieve 180$[km/h]$ in 15$[s]$ by a fixed acceleration after starting the car. The range of variation in mass of vehicle M and road condition coefficient c are imposed as $M_{max} =$ 1,400[kg], $M_{min}=$ 1,000[kg],

Table 8.1 Parameters used in the simulations

M Mass of vehicle	1,100 [kg]
J_w Inertia of wheel	21.1 [kg/m^2]
r Radius of wheel	0.26 [m]
λ^* Reference slip ratio	0.13
g Acceleration of gravity	9.81 [m/s^2]

$c_{max} = 0.9$ and $c_{min} = 0.1$ respectively. So the nominal values of mass and road condition coefficient can be obtained as $\hat{M} = 1{,}200$ [Kg] and $\hat{c}$ =0.5.

8.5.1 Robust Performance

8.5.1.1 Simulation 1

In order to verify the robustness of proposed method with variation both in the mass of vehicle and road condition, we compared it with the conventional SMC and no control used in the system. For making the variation to the mass of vehicle, the value of M was assigned to 1,000, 1,200 and 1,400 [kg] respectively. In this simulation, we consider two different road conditions, a high friction road (dry asphalt) for $t \in [0, 3)[s]$ and a low friction road (ice road) for t ∈[3,5] [s].

Figures 8.3, 8.4 and 8.5 show the responses of slip ratio for different masses. The responses with proposed SMC maintain to the reference slip ratio value 0.13 accurately in a very short time, regardless of both of the mass and road condition varying. Moreover, the slip ratio with the conventional SMC does not converge to the reference value. In addition, these figures show that the proposed method overcomes the advantage of the conventional SMC which may introduce the steady state error.

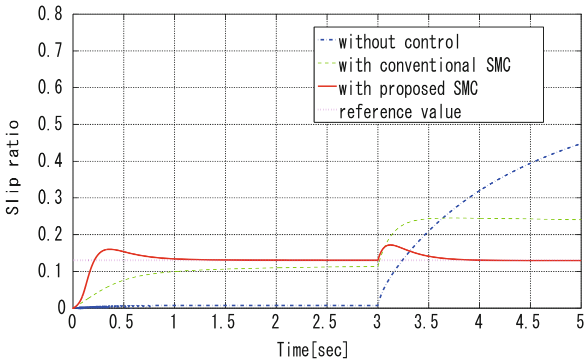


Fig. 8.3 Slip ratio with mass of vehicle equals 1,000 [kg] (Simulation 1)

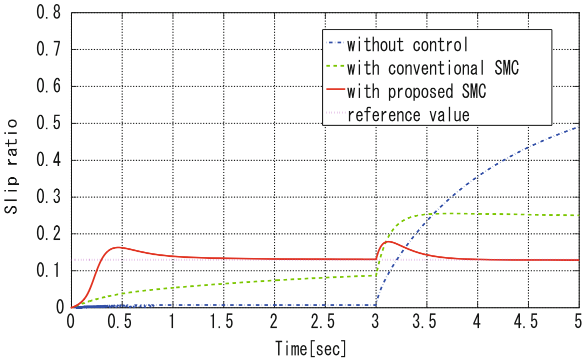


Fig. 8.4 Slip ratio with mass of vehicle equals 1,200 [kg] (Simulation 1)

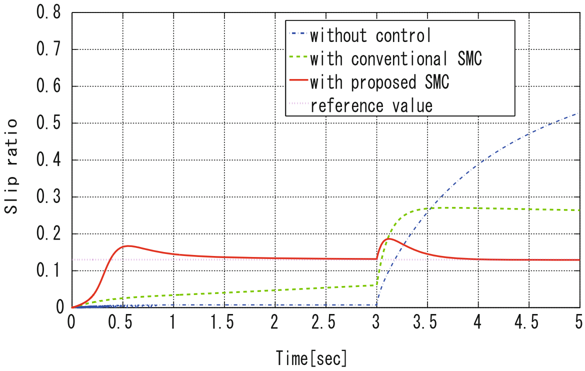


Fig. 8.5 Slip ratio with mass of vehicle equals 1,400 [kg] (Simulation 1)

These results indicate that the proposed SMC performs strong robustness to the variation in both of the mass of vehicle and road condition.

8.5.1.2 Simulation 2

In addition to simulation 1, to check the robustness performance of proposed method in other conditions, we perform simulation 2 as follows.

In simulations, we consider three different road conditions, a dry asphalt for $t \in [0, 2)[s]$, an icy road for $t \in [2, 8)[s]$ and a wet asphalt for $t \in [8, 10][s]$. The variation in the mass of vehicle is made by assigning the value of M to 1,000 [kg], 1,100 [kg], 1,200 [kg], 1,300 [kg] and 1,400 [kg] respectively.

We compared the proposed SMC with the conventional SMC and no control. Figures 8.6, 8.7 and 8.8 show the responses of slip ratio under three different road conditions for three different masses respectively.

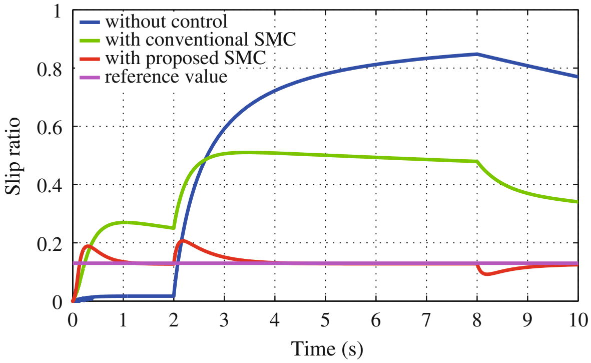


Fig. 8.6 Slip ratio with $M = 1{,}000$ [kg] (Simulation 2)

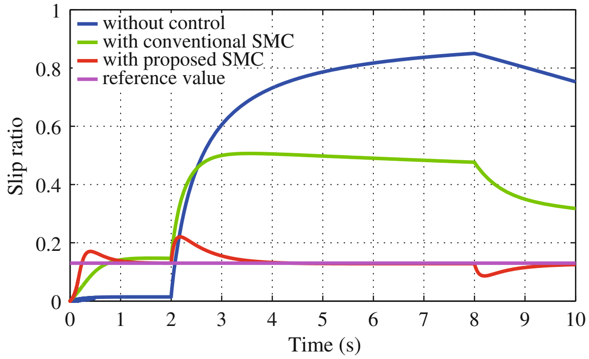


Fig. 8.7 Slip ratio with $M = 1{,}200$ [kg] (Simulation 2)

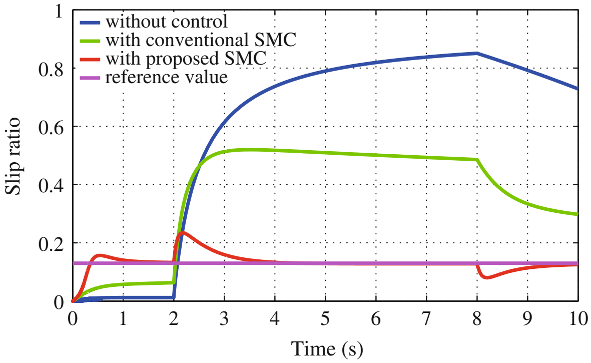


Fig. 8.8 Slip ratio with $M = 1{,}400$ [kg] (Simulation 2)

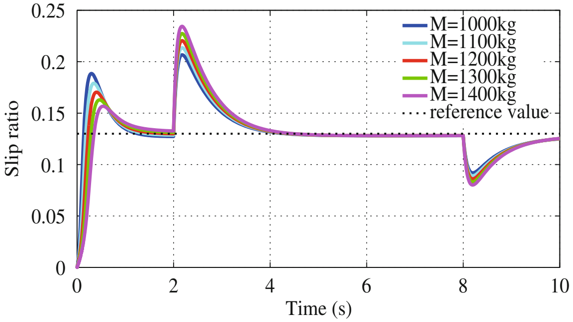


Fig. 8.9 Slip ratio by the proposed method against variation of M (Simulation 2)

Figure 8.9 shows the responses of slip ratio with different masses can converge to the reference value under the variation in the road condition. It is known that when the mass gets the nominal value 1,200 [kg], in the first 2 [s], the response is more accurately than the car with other mass. But after 2 [s], the performance drops down with the mass increases.

The responses with proposed SMC can suppress the slip ratio to the reference value 0.13 accurately in a very short time whenever both of the mass and road condition are changing. In addition, the slip ratio with the conventional SMC does not converge to the reference value because of the steady state error. When the car starts off at 0 [s] or runs into an icy road at 2 [s], the slip ratio response using control method grows with the increasing wheel speed as a result of too much torque generated. As the car travels from icy road to wet asphalt in 8 [s], the slip ratio decreases with the decreasing wheel speed, when the torque generated at that time cannot satisfy the one required on the wet asphalt. The car without control is to make the slip ratio to 0, so at the first stage the response is converged to 0. However, when the car runs into the ice road at 2 [s], the wheel spins out of control resulting that the wheel speed in-creasing suddenly, which leads to a large slip ratio value. Therefore, we can see that the proposed SMC has a good performance against the variation in both of the mass of vehicle and road condition.

8.5.2 Acceleration Performance

It is different from the simulation condition described in previous that the simulations are executed under unchanging road condition with mass every time.

Figure 8.10 shows the time required for 100 m by the car with different control method. The x-axis label indicates the cases of different road condition and mass, for example, $DA1000$ says the car with $M = 1{,}000$ [kg] is driving on the dry asphalt, $WA1200$ shows the case with $M = 1{,}200$ [kg] on the wet asphalt and $IR1400$ is the case with $M = 1{,}400$ [kg] on the icy road. As shown in Fig. 8.10, it takes minimum time by the proposed method in every case. So we can see that the car with proposed

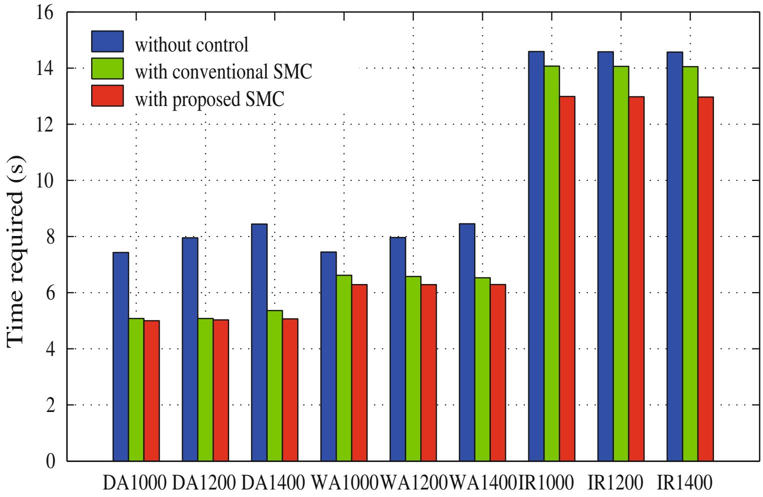


Fig. 8.10 Acceleration performance

SMC have gained the best acceleration. In other words, the results also indicate the vehicle with the proposed SMC can keep the loss of driving force at a minimum.

8.5.3 Energy Consumption

To confirm the effectiveness of the proposed SMC for energy saving, we compare it to the conventional SMC and no control method. Generally, It's difficult to evaluate the energy consumption accurately without measurement by experiments on the real vehicle. In this paper, therefore, we estimate the energy cost by calculating the rotational energy of motor. As a beginning, we give the following assumptions;
Assumption 1 The electric power is all used to drive the wheel.
Assumption 2 The power consumed by the vehicle is in proportional with the rotational energy due to the rotation of driven wheel.

The rotational energy E_r is defined by the rotational inertia of wheel J_w and the angular velocity w as

$$E_r = \frac{1}{2} J_w w^2. \tag{8.46}$$

Under these assumptions, we calculate the energy consumed in the simulations in 8.5.1.2.

Figure 8.11 shows the results of electric energy consumed by different mass case. From Fig. 8.11 we can see that the proposed SMC consumes minimum energy in every case. The car without control takes most energy because the spin of wheel on

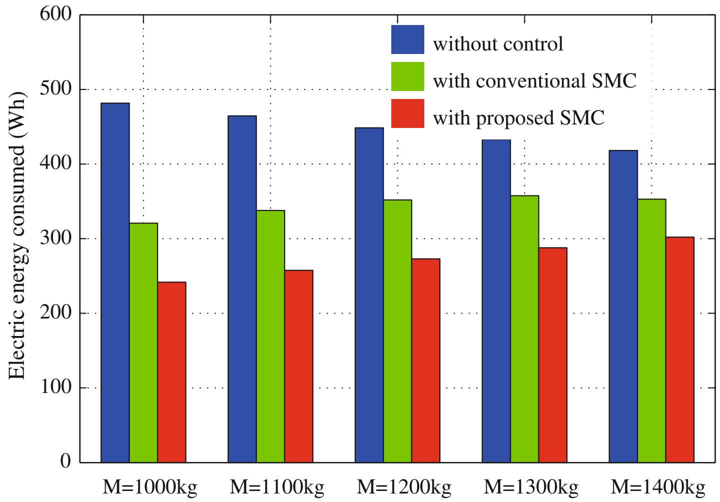


Fig. 8.11 Energy consumption

the icy road in $t \in [2,8)$ [s] leads to much energy loss. As the mass increases, the amount of energy cost decreases because the car suppresses the spin of wheel by increment of mass to get more driving force. Conversely, the energy consumption with the proposed SMC and conventional SMC increases due to the rising cost of control as the mass increases. From this perspective, it also shows that an EV should be made more light to save more energy.

8.6 Conclusions

This paper proposes new SMC method with the integral action for EV traction control. The method can improve the robust performance of EV traction by controlling the slip ratio with low energy consumption against the variation of mass of vehicle and road conditions. We can verified that the the proposed method shows good robust performance with low energy consumption by comparing to conventional method.

As future works, in this paper, the gain K_i of integral action added in the sliding function was determined by trial and error, so it is necessary to develop a systematic method to find the optimal value of K_i. Moreover, this paper was limited to showing the results with some example conditions using a simplified one wheel model, but to make the method practical, for a variety of road conditions and mass of vehicle must be verified for more detailed two-wheel and four-wheel models. In addition, the suitability of the proposed method must be studied not only for the slip suppression addressed by this paper but also for overall driving including during braking.

Even for this issue, however, the basic framework of the proposed method can be used as is and can also be extended relatively easily to form a foundation for making practical high performed robust traction control systems with low energy consumption for EVs by promoting further progress.

Acknowledgments This research was partially supported by Grant-in-Aid for Scientific Research (C) (Grant number: 24560538; Tohru Kawabe; 2012-2014) from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1. Brown, S., Pyke, D., Steenhof, P.: Electric vehicles: The role and importance of standards in an emerging market. Energy Policy **38**(7), 3797–3806 (2010)
2. Mousazadeh, H., Keyhani, A., Mobli, H., Bardi, U., Lombardi, G., Asmar, T.: Environmental assessment of RAMseS multipurpose electric vehicle compared to a conventional combustion engine vehicle. J. Cleaner Prod. **17**(9), 781–790 (2009)
3. Hirota, T., Ueda, M., Futami, T.: Activities of Electric Vehicls and Prospect for Future Mobility. Journal of The Society of Instrument and Control Enginnering **50**, 165–170 (2011). (in Japanese)
4. Kondo, K., 2011. Technological Overview of Electric Vehicle Traction, Journal of The Society of Instrument and Control Enginnering (in Japanese), Vol. 50, pp. 171–177.
5. Zanten, A.T., Erhardt, R. and Pfaff, G., 1995. VDC; The Vehicle Dynamics Control System of Bosch, Proc. Society of Automotive Engineers International Congress and Exposition 1995, Paper No. 950759.
6. Kin, K., Yano, O., Urabe, H.: Enhancements in Vehicle Stability and Steerability with VSA. Proc. JSME TRANSLOG **2001**, 407–410 (2001). (in Japanese)
7. Sawase, K., Ushiroda, Y. and Miura, T., 2006. Left-Right Torque Vectoring Technology as the Core of Super All Wheel Control (S-AWC), Mitsubishi Motors Technical Review, No.18, pp. 18–24 (in Japanese).
8. Kodama, K., Li, L. and Hori, H., 2004. Skid Prevention for EVs based on the Emulation of Torque Characteristics of Separately-wound DC Motor, Proc. The 8th IEEE International Workshop on Advanced Motion Control, VT-04-12, pp. 75–80.
9. Mubin, M., Ouchi, S., Anabuki, M., Hirata, H.: Drive Control of an Electric Vehicle by a Non-linear Controller. IEEJ Transactions on Industry Applications **126**(3), 300–308 (2006). (in Japanese)
10. Fujii, K. and Fujimoto, H., 2007. Slip ratio control based on wheel control without detection of vehicle speed for electric vehicle, IEEJ Technical Meeting Record, VT-07-05, pp. 27–32 (in Japanese).
11. Hori, Y., 2000. Simulation of MFC-Based Adhesion Control of 4WD Electric Vehicle, IEEJ Record of Industrial Measurement and Control, IIC-00-12 (in Japanese).
12. Kawabe, T., Kogure, Y., Nakamura, K., Morikawa, K., Arikawa, Y.: Traction Control of Electric Vehicle by Model Predictive PID Controller. Transaction of JSME Series C **77**(781), 3375–3385 (2011). (in Japanese)
13. J. J. E. Slotine, J.J.E. and Li, W., 1991. Applied Nonlinear Control, Prentice-Hall.
14. Eker, I.,AKinal, A., : Sliding Mode Control with Integral Augmented Sliding Surface: Design and Experimental Application to an Electromechanical system. Electrical Engineering **90**, 189–197 (2008)
15. Pacejka, H.B., Bakker, E.: The Magic Formula Tyre Model. Vehicle system dynamics **21**, 1–18 (1991)

Chapter 9
A Reactive Controller Based on Online Trajectory Generation for Object Manipulation

Wuwei He, Daniel Sidobre and Ran Zhao

Abstract In this paper, we present a new solution to build a reactive trajectory controller for object manipulation in Human Robot Interaction (HRI) context. Using an online trajectory generator, the controller build a time-optimal trajectory from the actual state to a target situation every control cycle. A human aware motion planner provides a trajectory for the robot to follow or a point to reach. The main functions of the controller are its capacity to track a target, to follow a trajectory with respect to a task frame, or to switch to a new trajectory each time the motion planner provides a new trajectory. The controller chooses a strategy from different control modes depending on the situation. Visual servoing by trajectory generation and control is presented as one application of the approach. To illustrate the potential of the approach, some manipulation results are presented.

Keywords Robotic · Manipulation · Trajectory generation · Human-robot interaction · Control

9.1 Introduction

Intuitive and natural object exchange is one of the basic necessary manipulation tasks in the context of Human Robot Interaction (HRI). This paper presents a controller

This work has been supported by the European Communitys Seventh Framework Program FP7/2007-2013 SAPHARI under grant agreement no. 287513 and by the French National Research Agency project ANR-07-ROBO-0011 ASSIST and ANR-10-CORD-0025 ICARO.

W. He (✉) · D. Sidobre · R. Zhao
CNRS, LAAS, 7 Avenue du colonel Roche, 31400 Toulouse, France
e-mail: wuwei.he@laas.fr

W. He · D. Sidobre · R. Zhao
UPS, LAAS, Univ. de Toulouse, 31400 Toulouse, France
e-mail: ran.zhao@laas.fr

D. Sidobre
e-mail: daniel.sidobre@laas.fr

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_9

which enables the robot to realize a complete task of object exchange while respecting human's safety and other HRI specifications, such as monitoring the accessibility of human. This elementary manipulation task demands integration of different elements like geometrical and human-aware reasoning, position and external force monitoring, 3D vision and human perception information. The control system presented in this paper proposes to define a plan as a series of control primitives, each associated with a trajectory segment and a control mode.

Structure of the Paper: We introduce firstly the architecture of the system and present the related work. In Sect. 9.2 we present briefly the trajectory generator and how cost values are associated to the trajectory based on cost maps. In Sect. 9.3, we discuss the controller, which is based on online trajectory generation. Some results and comparison could be found in Sect. 9.4, followed by the conclusion.

9.1.1 Software Architecture for Human Robot Interaction

The robots capable of doing HRI must realize several tasks in parallel to manage various information sources and complete tasks of different levels. Figure 9.1 shows the proposed architecture where each component is implemented as a GENOM module. GENOM [14] is a development environment for complex real time embedded software.

At the top level, a task planner and supervisor plans tasks and then supervises the execution. The module **Spa**tial **R**easoning and **K**nowledge (SPARK) maintains a 3D

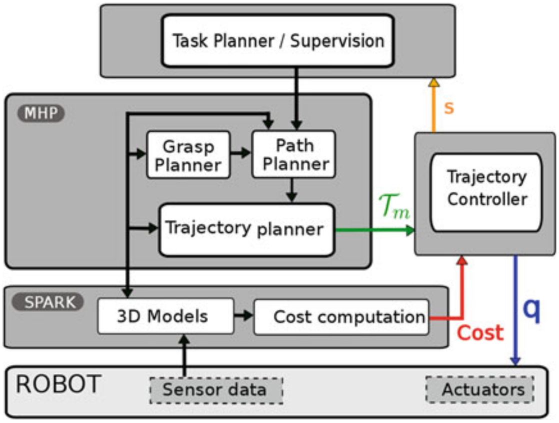


Fig. 9.1 Software Architecture of the robot for HRI manipulation. $\mathcal{T}_m$ is the main trajectory calculated initially by **M**otion in **H**uman **P**resence (MHP). The controller takes also cost maps from SPARK. The controller sends control signals in joint (q in the figure) to the servo system, and during the execution, the controller sends the state of the controller (s) to the supervisor

model of the whole environment, including objects, robots, posture and position of humans [29]. It manages also the related geometrical reasoning of the 3D models, such as collision risk between the robot parts and between robot and environment. An important element is that SPARK produces cost maps, which discribe a space distribution relatively to geometrical properties like human accesibility. The softwares for perception, from which SPARK updates the 3D model of the environment, are omitted here for simplicity. The module runs at a frenquency of 20Hz, limited mainly by the complexity of the 3D vision and of the perception of human.

Another important module, **M**otion in **H**uman **P**resence (MHP), integrates path and grasp planner. Rapidly exploring Random Tree (RRT) [24] and its variants are used by the path planner. The paths could be in Cartesian or joint spaces depending on the task type. From the path, an output trajectory is computed to take the time into account. MHP calculates a new trajectory each time the task planner defines a new task or when the supervisor decides that a new trajectory is needed for the changes of the environment.

The system should adapt its behavior to the dynamic environment, mainly the human activities. But as the planning algorithms are time consuming, we introduce a trajectory controller that runs at 100 Hz, an intermediate frequency between the one of the MHP planner and the one of the fast robot servo system. This trajectory controller allows the system to adapt faster the trajectory to the environment changes.

The main functionalities of the whole controller are:

- A decision process capable of integrating information form high-level software and building control primitives by segmenting the trajectory based on the HRI specifications.
- Algorithms for target or trajectory tracking, trajectory switching, and coordinates transformation.
- A low-level collision checker and a monitor of the external torques for the safety issues.

9.1.2 Related Works

Reactive controller for object manipulation is a research topic that is part of the fundamentals of robotic manipulation. Firstly, trajectory generation based approaches have been developed. In [7], results from visual system pass firstly through a low-pass filter. The object movement is modeled as a trajectory with constant acceleration, based on which, catching position and time is estimated. Then a quintic trajectory is calculated to catch the object, before being sent to a PID controller. The maximum values of acceleration and velocity are not checked when the trajectory is planned, so the robot gives up when the object moves too fast and the maximum velocity or acceleration exceeds the capacity of the servo controller. In [15], inverse kinematic functions are studied, catching a moving object is implemented as one application, a quintic trajectory is used for the robot manipulator to joint the closest point on the predicted object movement trajectory. The systems used in those works are all quite simple and no human is present in the workspace. A more recent work can be found in [19], in which a controller for visual servoing based on Online Trajectory Generation (OTG) is presented, and the results are promising.

Secondly, the research area of visual servoing provides also numerous results, a survey of which were presented by Chaumette and Hutchinson [9, 10] and a comparison of different methods can be found in [13]. Classical visual servoing methods produce rather robust results and stability and robustness can be studied rigorously, but they are difficult to integrate with a path planner, and could have difficulties when the initial and final positions are distant.

Another approach to achieve reactive movements is through Learning from Demonstration(LfD). In [8, 31] points in the demonstrated trajectory are clustered, then a Hidden Markov Model(HMM) is built. Classification and reproduction of the trajectories are then based on the HMM. A survey for this approach is proposed in [1]. Although LfD can produce the whole movement for objects manipulation, many problems may arise in a HRI context as LfD demands large set of data to learn, and the learned control policies may have problem to cope with a dynamic and unpredictable environment where a service robot works.

Our approach to build the controller capable of controlling a complete manipulation tasks is based on Online Trajectory Generation. More results on trajectory generation for robot control can be found in [16, 20, 22]. The controller is capable of dealing with various data in HRI context. Compared to methods mentioned above, approaches based on OTG have the following advantages:

- The integration with a path planner is easy and allows to comply with kinematic limits like the one given by human safety and comfort.
- The path to grasp a complex moving object is defined in the object frame, making sure that the grasping movement is collision free.
- The trajectory based method allows to create a simple standard interface for different visual and servo systems, easy plug-in modules can be created.

The controller integrates various information from high-level software. More information about human-aware motion planning in the system can be found in [23]. More about geometrical reasoning can be found in [24, 27, 28]. Physical Human Robot Interaction is a dynamic research area including various aspects, including safety, control architecture, planning, human intention recognition, and more. Readers may refer to [12, 26, 30] for more information of this promising research field.

9.2 Trajectory and Control Primitives

9.2.1 Online Trajectory Generation

Trajectories are time functions defined in geometrical spaces, mainly Cartesian space and joint space for robots. The books from Biagiotti [2] and the one from Kroger [17] summarize background materials. For detailed discussion about the trajectory generator used in the system, the reader can refer to [5, 6], here we describe some results without further discussion.

Given a system in which position is defined by a set of coordinate X, a trajectory $\mathcal{T}$ is a function of time defined as:

$$\begin{aligned} \mathcal{T} : [t_I, t_F] &\longrightarrow \mathbb{R}^N \\ t &\longmapsto \mathcal{T}(t) = X(t) \end{aligned} \tag{9.1}$$

The trajectory is defined from the time interval $[t_I, t_F]$ to $\mathbb{R}^N$ where N is the dimension of the motion space. The trajectory $\mathcal{T}(t)$ can be a direct function of time or the composition $\mathcal{C}(s(t))$ of a path $\mathcal{C}(s)$ and a function $s(t)$ describing the time evolution along this path. The time evolution could be used in the controller to slow down or to accelerate when necessary [4].

Our trajectory generator is capable of generating type V trajectories defined by Kroger [17] as satisfying:

$$\begin{array}{lll} X(t_I) \in \mathbb{R} & X(t_F) \in \mathbb{R} & |V(t)| \leqslant V_{max} \\ V(t_I) \in \mathbb{R} & V(t_F) \in \mathbb{R} & |A(t)| \leqslant A_{max} \\ A(t_I) \in \mathbb{R} & A(t_F) \in \mathbb{R} & |J(t)| \leqslant J_{max} \end{array} \tag{9.2}$$

Where X, V, A and J are respectively the position, the velocity, the acceleration and the jerk.

For a motion of dimension N, the algorithms find a trajectory $X(t)$, which satisfies:

1. The initial conditions (IC): $X(t_I) = X_I$, $V(t_I) = V_I$ and $A(t_I) = A_I$;
2. The final conditions (FC): $X(t_f) = X_F$, $V(t_f) = V_F$ and $A(t_f) = A_F$;
3. The continuity class of the trajectory is $\mathcal{C}^2$.
4. The kinematics limits V_{max}, A_{max} and J_{max}.

The problem is solved by a series of 3^{rd} degree polynomial trajectories. Such a trajectory is composed of a vector of one-dimensional trajectories, which can be written as $\mathcal{T}(t) = ({}_1Q(t), {}_2Q(t), \ldots, {}_NQ(t))^T$ for joint motions or $\mathcal{T}(t) = ({}_1X(t), {}_2X(t), \ldots, {}_NX(t))^T$ in Cartesian space.

For the discussion of the next sections, we define Motion Condition as the position, velocity and acceleration at time t of the trajectory: $M(t) = (X(t), V(t), A(t))$. Once the trajectory is calculated, the function $M(t) = getMotion(t, \mathcal{T})$ returns the Motion Condition on trajectory $\mathcal{T}$ at time t.

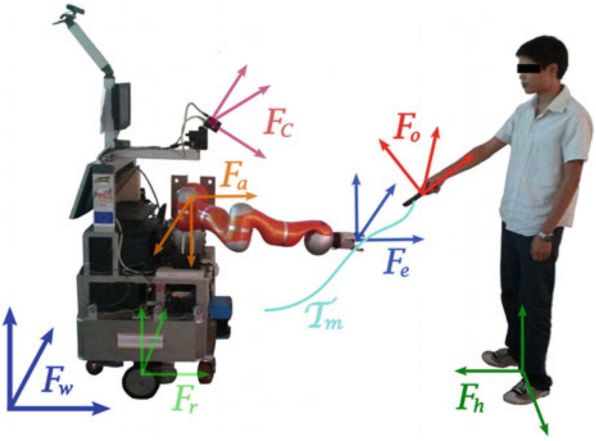


Fig. 9.2 Frames for object exchange manipulation: F_w world frame; F_r robot frame; F_c camera frame; F_e end effector frame; F_o object frame; F_h human frame. The trajectory realizing a manipulation should be controlled in different task frames

9.2.2 Control Primitives

In HRI, the robot does various tasks like picking up an object, giving an object to human, taking an object from the human. For each of the task, a path is planned to realize it, and then the path is transformed into a trajectory. The controller designed here takes directly the trajectory as input and segments it based on the cost maps.

Figure 9.2 shows the basic frames needed to define a task. The trajectory $\mathcal{T}_m$ defines the move that allows the robot to do the task of grasping an object handed by the human.

Based on the cost values associated to each point of the trajectory, the trajectory is divided into segments associated to a control strategy. The 3D cost maps used are of different types: collision risk map calculated based on the minimum distance between trajectory and the obstacles; visibility and reachability map of a human [27] and safety and comfort 3D map of a human, Fig. 9.3 shows two examples of cost maps. For example, when the risk of collision with the robot base is high, the trajectory can be controlled in the robot frame. Similarly, in the case where the human is handing an object to the robot, the grasping must be controlled in the object frame. [26] details other aspects of the use of cost maps to plan manipulation tasks.

To simplify the presentation, in the reminder of the paper we focus on the manipulation tasks where a human hands over an object to the robot. During the manipulations, the human moves and the different frames defining the task move accordingly. Based on the change of cost values, we divide the trajectory $\mathcal{T}_m$ in Fig. 9.2 into three segments, as illustrated in the configuration space in the left part of Fig. 9.7. In the figure, the points connecting the trajectory segments are depicted

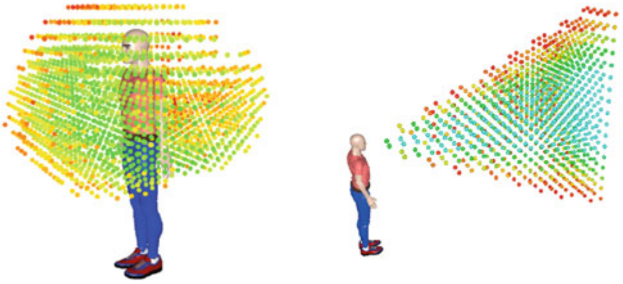


Fig. 9.3 *Left* 3D reachability map for a human. *Green points* have low cost, meaning that it is easier for the human to reach, while the *red* ones, having high cost, are difficult to reach. One application is that when the robot plans to give an object to a human, an exchange point must be planned in the *green* zone. *Right* 3D visibility map. Based on visibility cost, the controller can suspend the execution if the human is not looking at the robot

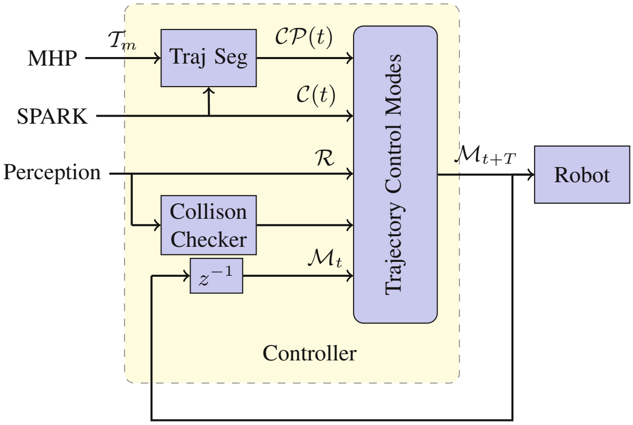


Fig. 9.4 Input and output of the controller. $\mathcal{T}_m$ is the trajectory computed by MHP, it is then segmented into control primitives ($\mathcal{CP}(t)$). Traj Seg represents *trajectory segmentation*. $\mathcal{C}(t)$ are the cost values. $\mathcal{R}$ represents the tranformation matrices, giving the position of the target and of the robot. $\mathcal{M}_t$ is the current state of the robot, $\mathcal{M}_{t+T}$ is desired motion condition for the next control cycle. z^{-1} represents delay of a control cycle

by red dots. The first segment $\mathcal{T}_1$, which is defined in the robot frame, has a high risk of auto-collision. When human or object moves, the cost value of collision risk stays the same. Segment $\mathcal{T}_2$ has a lower collision cost value, so modifying the trajectory inside this zone does not introduce high collision risk. The end part, segment for grasping movement $\mathcal{T}_g$, has a high collision cost value. To ensure the grasping succeeds without collision this segment of trajectory should be controlled in the moving object frame.

We name *task frame* the frame in which the trajectory must be controlled. We define a *control primitive* $\mathcal{CP}$ by the combination of five elements: a segment of trajectory, a cost function, a task frame, a control mode, and a stop condition.

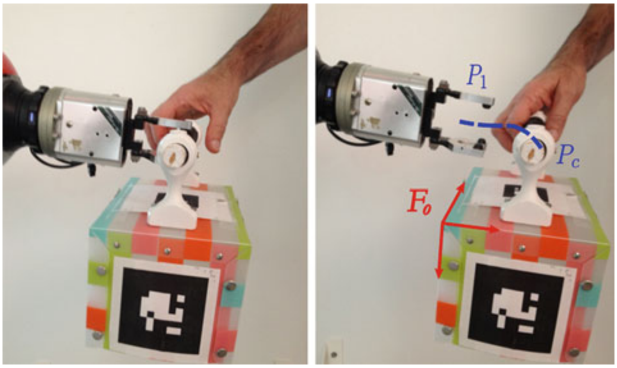


Fig. 9.5 A simple case of grasp. *Left* a planned grasp defines contact points between the end effector and the object. *Right* To finish the grasping, the manipulator must follow the blue trajectory P_1 - P_c, and then close the gripper. This movement must be controlled in the object frame F_o

$$\mathcal{CP}(t) = (\mathcal{T}_{seg}(t),\ \mathcal{C}(t),\ \mathcal{F},\ \mathcal{O},\ \mathcal{S})^T \tag{9.3}$$

In which, $\mathcal{T}_{seg}(t)$ is the trajectory segment, $\mathcal{C}(t)$ is the cost value associated to the trajectory which is monitored during the execution of a control primitive, $\mathcal{F}$ is the task frame, $\mathcal{O}$ is the control mode which we will define in next section, and $\mathcal{S}$ is the stop condition of the control primitive. For example, the grasping movement includes five elements: the trajectory segment $\mathcal{T}_g$, the high collision risk cost value $\mathcal{C}(t)$, the task frame $\mathcal{F}_o$, the control mode as trajectery tracking, and the stop condition $\mathcal{S}$ as a predefined threshold for the distance between the robot end effector and the end point of $\mathcal{T}_g$. In the literature, Manipulation Primitives or Skill Primitives are often the concept for the intermediate level between planning and control and have been discussed in numerous works, as in [18].

Using the definition of control primitives ($\mathcal{CP}(t)$) and Motion Condition: $M(t) = (X(t), V(t), A(t))$, the different components of the trajectory controller and the input and output are presented in Fig. 9.4. The initial trajectory $\mathcal{T}_m$ is segmented into a series of $\mathcal{CP}(t)$. The cost values $\mathcal{C}(t)$ are used during the segmentation, they are also monitored by the controller during execution of a control primitive. The collision checker integrates data from vision, human perception and encoder of the robot. It prevents collision risk by slowing down or suspending the task execution. With all the data and the current Motion Condition $\mathcal{M}_t$ of the robot, different control modes can compute Motion Condition for the next control cycle, which are the input for the robot sorvo system.

Figure 9.5 shows the last control primitive of *grasping an object*. It is similar to the end part, $\mathcal{T}_g$, of the trajectory in Fig. 9.2. The grasp position, the contact points and the final trajectory are planned by the grasp planner. More details on the grasp planner are given in [3, 25]. When the object moves, the object frame F_o and the path of the trajectory moves also. So to avoid collision, the trajectory of these control primitives must be controlled in the object frame F_o.

9.3 Reactive Trajectory Control

At the control level, a task is defined by a series of control primitives, each defined by a quintuplet. The first level of the proposed trajectory controller is a state machine, which controls the succession of the control modes, the collision managing and other special situations. Target tracking and trajectory tracking are parts of the control modes presented after the state machine.

9.3.1 Execution Monitoring

A state machine controls the switching between the different control modes associated to each control primitive and monitors the execution. Due to human presence, the robot environment is moving and the control task must be adapted accordingly. The state machine can also suspend or stop the control of a control primitive like depicted in Fig. 9.6.

Suspend Events: When the visual system fails or the target becomes unavailable, or because of some specific human activities based on the monitoring of cost value $\mathcal{C}(t)$, the trajectory controller should suspend the task.

Stop Events: Whatever the control mode chosen, unpredictable collisions can occur and they must stop the robot. Our controller uses two modules to detect these situations. The first one based on [11] monitors the external torques. The second is a geometric collision checker based on results from [21], it updates a 3D model of the workspace of the robot, and runs at the same frequency as the trajectory controller.

Slow Down on Trajectory: Based on the input cost function, the controller can slow down on the main trajectory by changing the time function $s(t)$. Imagine that a fast movement could cause some people anxiety when the robot is close to them, for example. Details about this specific situation can be found in [4]. In this situation, the controller is still executing the task but only at a slower speed.

Each elementary controller based on online trajectory controller is implemented with a simple state machine inside.

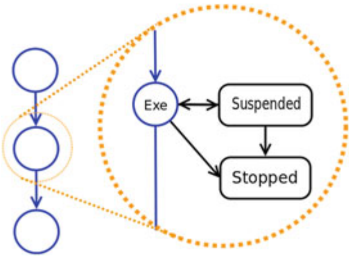


Fig. 9.6 In the *left*, each *circle* represents the controller of a control primitive. The system can suspend or stop the execution of a control primitive

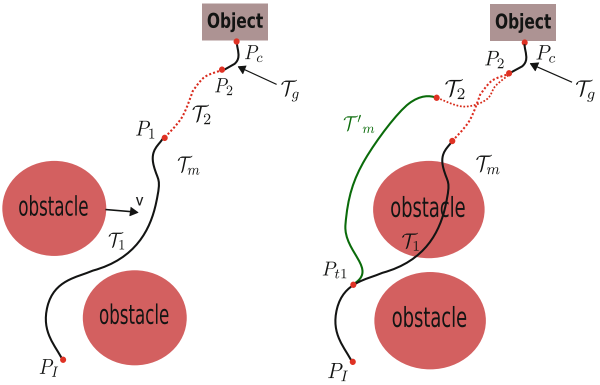


Fig. 9.7 *Left* trajectories of the control primitives. *Right* trajectory switching for the controller due to the movement of an obstacle

9.3.2 Trajectory Control Modes

Depending on the context defined by the control primitives, different control strategies must be developed. Online trajectory generator gives a flexible solution to build these controllers, which can easily react to unforeseen sensor events and adapt the kinematic parameters, mainly velocity, to the environment context. Switching to a new trajectory or a new frame in which the trajectory is controlled is also possible.

The main idea of the controller is to compute and follow a trajectory while joining up the target trajectory or a target point from the current state. Several control modes are defined to solve the reactive HRI manipulation problem.

Control Mode 1: Target Tracking. If we suppose the robot is in an area without risk of collision, the system can track the end point of the trajectory. In this case, the controller generates iteratively a trajectory to reach the end point and send the first step of this trajectory to a low-level controller. In the special case where the controller does target tracking with visual system, it does visual servoing.

Figure 9.8 shows the details of the trajectory control mode for *Target Tracking*. The object is at position O at current time, and moves following the curve $\mathcal{T}_{obj}$. This curve is obtained by a simple Kalman filter, building a movement model from the results of 3D vision system. $\mathcal{F}_r$ is the robot base frame, $\mathcal{F}_c$ and $\mathcal{F}_o$ are camera frame and object frame, respectively. Also, R_r^c is the 4×4 transformation matrix from $\mathcal{F}_r$ to $\mathcal{F}_c$ and R_c^o the transformation matrix from $\mathcal{F}_c$ to $\mathcal{F}_o$. They are all in dashed line and they change with time when the humans or objects move. Initially, the robot is at point P_e, since there is no risk of collision, the controller can simply track point P_2, which is the end point of the segment. It is also possible for the robot to join up the trajectory at another point P_{joint} defined in the object frame which is the task frame. The details of the algorithm is given in Algorithm 1 where:

T: duration of one control cycle.
$\mathbf{M_r}$: current motion condition of the robot, so $M_r = (X_r, V_r, A_r)$.

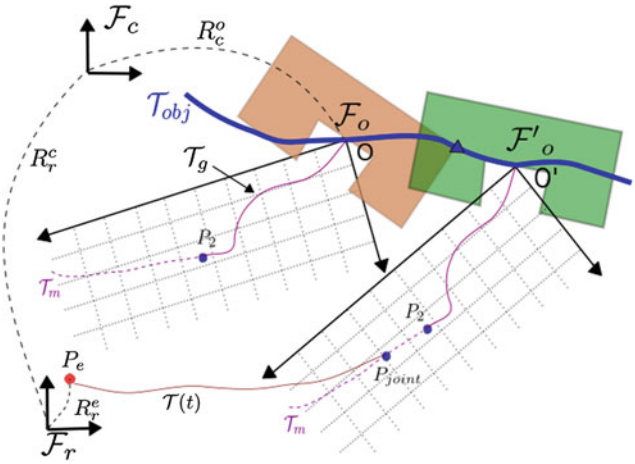


Fig. 9.8 *Control Mode 1*. The robot tracks a point. The object moves to the *right*, it is drawn at two times: firstly in *brown* for time t_1 and then in *green* at time t_2. In both cases, the entry point P_2 of the trajectory $\mathcal{T}_g$ is drawn relatively to the object frame $\mathcal{F}_o$

δ: distance threshold to stop the tracking process.
$\mathbf{M_g(t)}$: motion conditions at time t on trajectory $\mathcal{T}_g$.
$\mathbf{M_{P_2}}$: motion conditions of the target P_2 on the main trajectory, which is calculated by the planner.
MaxLoop: the maximum times the loop repeats for the controller to track the target or the trajectory. Once the time exceeds the value, the trajectory controller is suspended and a signal is sent to the supervisor, requiring the replanning of new task or a new path to realize the task.
X, **Q**: input signal in Cartesian space and in joint space for low-level controller.

Control Mode 2: Trajectory tracking in task frame. Once robot reaches point P_2, it starts the grasping movement, which corresponds to $\mathcal{T}_g$ in Fig. 9.7. The object is still moving, but as the robot is in the high cost zone, it should track the main trajectory in the task frame. The details of the control mode is given in Algorithm 2.

Figure 9.9 shows the details of the control mode, all the frames and object movements are the same as in Fig. 9.8, but the robot is at point P'_e. The robot tracks $\mathcal{T}_g$ in the frame $\mathcal{F}_o$, and will end up executing $\mathcal{T}'(t)$ in the robot frame.

Control Mode 3: Path re-planning and trajectory switch: during the execution, a path can be re-planned, for example when an obstacle moves (see Fig. 9.7). A new trajectory is computed and given to the controller that switches to the new trajectory. While the controller is following the trajectory $\mathcal{T}_m$, an obstacle moves and invalidates the initial trajectory. Then the system provides the controller with a new trajectory $\mathcal{T}'_m$ beginning at time t_1 in the future. The controller anticipates the switch, and when the robot reaches P_{t_1} at time t_1, the robot switches to the new trajectory $\mathcal{T}'_m$. Because the new trajectory $\mathcal{T}'_m$ is calculated using the state of the robot at time t_1 as

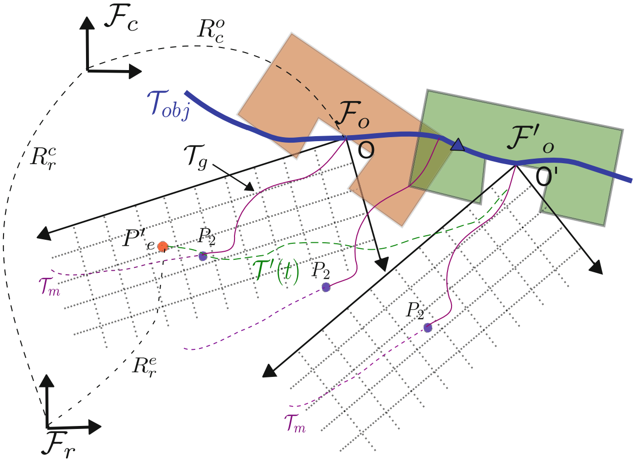


Fig. 9.9 *Control Mode 2.* control mode. Object at time t_1 is colored in *light brown*, and *green* at time t_2. It follows a movement model given as the *blue* trajectory $\mathcal{T}_{obj}$. The *purple* trajectory for grasping $\mathcal{T}_g$ stays unchanged in the object frame. The robot tracks the trajectory $\mathcal{T}_g$ as it does th7e grasping movement

Algorithm 1: Control for target tracking (*Control Mode 1*).

```
input : Target point $P_2$;
while $(distance(P_2, M_r) > \delta) \wedge (Loop < MaxLoop)$ do
    system time $t = t + T$, $Loop = Loop + 1$;
    Update perception data;
    if Collision Detected then Emergency stop;
    if Suspend Events Occur then Suspend task;
    Coordinates transformations;
    Generate Type V control trajectory $\mathcal{T}(t)$, for which: $IC = M_r$ , $FC = M_{P_2}$;
    $X = getMotion(t + T, \mathcal{T}(t))$;
    Inverse kinematics: $X \rightarrow Q$;
    $Q$ to the position servo system;
end
```

its initial condition, the trajectory is switched without problem. The controller keeps the possibility of path re-planning. In some cases, a new path is needed to accomplish the task.

In this paper, we essentially solved the problem of the task of grasping a moving object held by the human counterpart. For other tasks, like picking an object, giving an object to human or putting an object on the table, the same functionalities can also be used. For example, putting an object on a moving platform would require the end segment of the main trajectory to be controlled in the frame of the platform, which moves in the robot frame. Likewise giving an object to a moving human hand will require the manipulator to track the exchange point, normally planned by a human-aware motion planner till the detection that human grasps the object successfully.

Algorithm 2: Control for trajectory tracking in a moving work frame (*Control Mode 2*).

```
input : Trajectory segment $\mathcal{T}_g$;
while $(distance(P_c, M_r) > \delta) \wedge (Loop < MaxLoop)$ do
    system time $t = t + T$, $Loop = Loop + 1$;
    Update perception data and object movement model;
    if Collision Detected then Emergency stop;
    if Suspend Events Occur then Suspend task;
    Coordinates transformations;
    $M_{\mathcal{T}_g} = getMotion(t + T, \mathcal{T}_g)$;
    $M_{object} = getMotion((t + T, \mathcal{T}_{obj})$;
    $X = M_{\mathcal{T}_g} + M_{object}$;
    Inverse kinematics: $X \rightarrow Q$;
    $Q$ to the position servo system;
end
```

Although a general algorithm to decompose the tasks into control primitives is still to develop, the basic HRI tasks can all be controlled by the control modes discussed above.

9.4 Manipulation Results

We focus on some results of how the controller is integrated in a HRI manipulator. For the performance of the trajectory generator, readers may refer to [4, 6].

The controller has been implemented on the robot Jido at *LAAS-CNRS*. Jido is a mobile manipulator built up with a *Neobotix* mobile platform *MP-L655* and a *Kuka LWR-IV* arm. Jido is equipped with one pair of stereo cameras and an *ASUS Xtion* depth sensor. The software architecture that we used for Jido is presented in 9.1.1. Note that the pan-tilt stereo head may move during manipulations, then the transformation matrix from robot frame to camera frame is updated at the same frequency as the controller. The stereovision system uses marks glued on manipulated objects for localization. Unfortunately, the localization of these marks is highly sensible to lighting conditions and the estimated position of the object O in the robot frame $\mathcal{F}_r$ is very noisy and unstable. Figure 9.10 shows the results returned by the 3D-vision system in poor lighting conditions and the result given by a Kalman filter. Even when raw data oscillates, this filter is capable to reduce the offset at the price of an acceptable delay. This reduces the oscillations of the robot arm too.

Figure 9.11 shows the results of the target tracking by the trajectory controller, as in *Case 2*, over 25 seconds. For simplicity, only axis X is shown. The black dashed line is the position of the target, generated by the 3D vision system. The red line is the position of the robot. The two bottom diagrams show the velocity and acceleration of the robot in the same period. Firstly, we can see that the controller produces robust

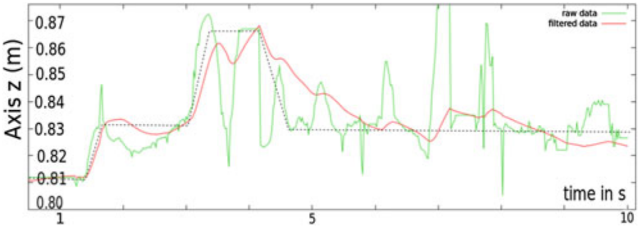


Fig. 9.10 Instability of the 3D vision in poor lighting condition. The *green curve* shows axis z of the localization result of an object in the robot frame over 10 s and the *red* one shows the filtered result. The object was held by a human that intended to move the object accordingly to the *black dashed curve*

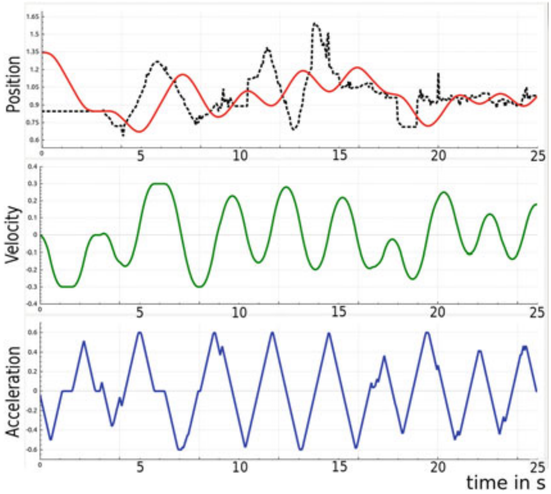


Fig. 9.11 Results of robot tracking a target: position (in m), velocity (in m/s) and acceleration (in m/s^2) during the tracking for 25 s. The *black dashed line* is the target position, with noise of the 3D vision, and the *red line* is the position of the robot, which tracks the target with a delay. The positions of the robot are calculated from measured joint values and the kinematic model, while velocity and acceleration are estimated. The velocity, acceleration and jerk are always limited, maintaining a smooth tracking process

behavior to the noise in the visual system. Secondly, the velocity and acceleration of the robot are saturated as type V trajectories and calculated.

Finally, we show the behavior of the controller for a complete manipulation task. Figure 9.12 shows the scenario of the manipulation and Fig. 9.13 shows the real position of the robot end effector in the robot frame (see Fig. 9.2 for the axes assignment of the robot base). The high-level task planner plans a task to receive the object. When the robot sees the object held by the human, the grasp planner calculates a valid grasp and the path planner with the trajectory generator plans the main trajectory for the robot to take the object.

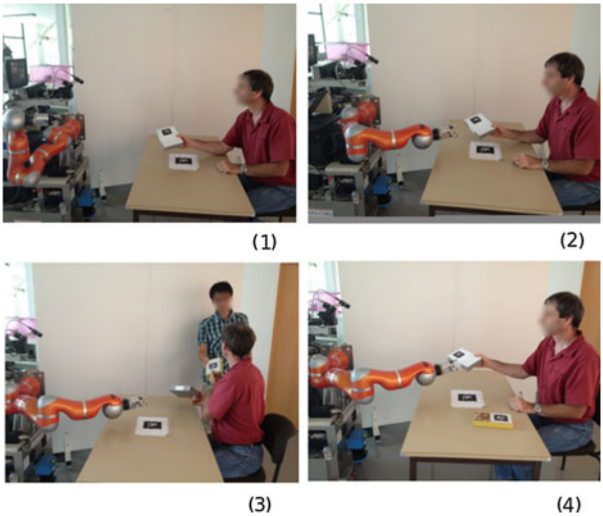


Fig. 9.12 (*1*) The controller is given the task of receiving the object from human. It tracks the first segment in the robot frame. (*2*) The object is moving and the robot tracks a target. (*3*) Human is distracted by another human and the task is suspended. (*4*) Human returns to the task, and the robot resumes the task and grasps the object

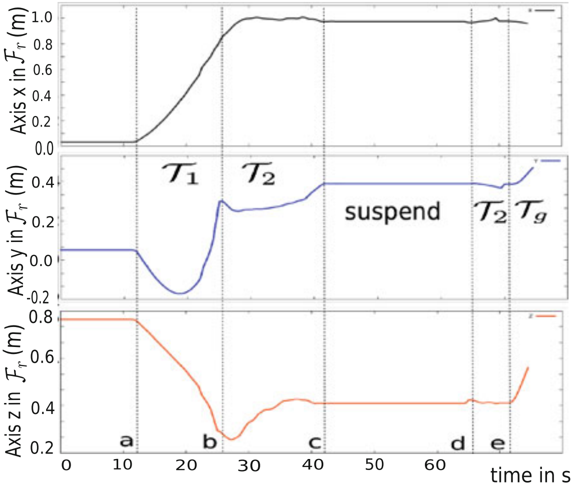


Fig. 9.13 Real position of the robot arm end effector in the robot frame. The motion starts at time a; Between a and b: the controller tracks the first trajectory segment $\mathcal{T}_1$ in $\mathcal{F}_r$; From b to c and d to e: target tracking; From c to d: the task is suspended; From e to end: the grasping movement controlled in $\mathcal{F}_o$

The trajectory is divided into three segments by the controller, and different control modes are chosen. As we as seen above, each control primitive is associated to a trajectory segment. In this case, we obtain three segments, the first one is controlled in the robot frame, the second is defined as the tracking of the entry point of the third segment and the third segment is a trajectory defined in the object frame.

During the target tracking, human is distracted because a second human arrives and gives an object to him. High-level software detects this event by monitoring the visibility cost map of the human. Because of the event, the controller suspends the task. It resumes the tracking when the human look again at the robot and the object to exchange come back in the reachable zone. Then, the grasping movement is finished. Note the performance of the target tracking process in the time intervals: between b to c, and between d to e. The controller finished the task reactively without the need of task or path replanning. The results shows that a reactive controller can be built based on Online Trajectory Generation, and as it is more responsive for the human, the robot is easier to interact with. Before the implementation of the reactive controller, the human needs to hold the object and stay still until the robot grasps it successfully.

9.5 Conclusion

A controller based on online trajectory generation has been presented with some results of robot grasping an object held by a human. The first results presented in the paper illustrate the versatility of the controller. In the example shown here, the controller switches between frames and suspends the control task when the human is distracted.

The trajectory controller employs different control modes for different situations. The control modes are all based on a trajectory generator. It is easy to use and to implement and gives an efficient solution to follow trajectories and track moving objects in the HRI context. More precisely, it can adapt kinematic limits to the changing state of the scene and switch between trajectories and control modes.

The future work is to extend the trajectory controller to manage forces and to handle force events. Furthermore, the possibility to apply this type of trajectory control and the concept of control primitives to dual arm manipulations opens also interesting perspectives.

References

1. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robot. Auton. Syst. **57**(5), 469–483 (2009)
2. Biagiotti, L., Melchiorri, C.: Trajectory Planning for Automatic Machines and Robots. Springer, Berlin (2008)
3. Bounab, B., Sidobre, D., Zaatri, A.: Central axis approach for computing n-finger force-closure grasps. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1169–1174 (2008)

4. Broquère, X.: Planification de trajectoire pour la manipulation d'objets et l'interaction Homme-robot. PhD thesis, LAAS-CNRS and Université de Toulouse, Paul Sabatier (2011)
5. Broquère, X., Sidobre, D.: From motion planning to trajectory control with bounded jerk for service manipulator robots. In: IEEE Intenational Conference of Robotics And Automation (2010)
6. Broquère, X., Sidobre, D., Herrera-Aguilar, I.: Soft motion trajectory planner for service manipulator robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 2808–2813 (2008)
7. Buttazzo, G. Allotta,B., Fanizza, F.: Mousebuster: A robot for real-time catching. IEEE Control Syst. Mag., **14**(1):49–56 (1994).
8. Calinon, S., Billard, A.: Stochastic gesture production and recognition model for a humanoid robot. In: Intelligent Robots and Systems, 2004. (IROS 2004), vol. 3, pp. 2769–2774 (2004)
9. Chaumette, F., Hutchinson S.: Visual servo control. part I: Basic approaches. IEEE Robot. Autom. Mag., 4(13):82–90 (2006)
10. Chaumette, F., Hutchinson S.: Visual servo control. part II: Advanced approaches. IEEE Robot. Autom. Mag., **1**(14):109–118 (2007)
11. De Luca, A., Ferrajoli, L.: Exploiting robot redundancy in collision detection and reaction. In: IROS 2008, pp. 3299–3305, sept. 2008.
12. De Santis, A., Siciliano, B., De Luca, A., Bicchi, A.: An atlas of physical human-robot interaction. Mech. Mach. Theor. **43**(3), 253–270 (2008)
13. Farrokh, J.-S., Lingfeng,D., William J.: Comparison of basic visual servoing methods. IEEE/ASME Trans. Mech, **16**(5):967–983 (2011)
14. Fleury, S. Herrb, M., Chatila, R.: Genom: A tool for the specification and the implementation of operating modules in a distributed robot architecture. In: IEEE/RSJ International Conference on Intelligent Robotics and Systems (1997)
15. Gosselin, G., Cote, J., Laurendeau, D.: Inverse kinematic functions for approach and catching operations. IEEE Trans. Syst. Man Cybern., **23**(3):783–791 (1993)
16. Haschke, R., Weitnauer, E., Ritter, H.: On-Line Planning of Time-Optimal, Jerk-Limited Trajectories. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008, pp. 3248–3253 (2008)
17. Kröger, T.: On-Line Trajectory Generation in Robotic Systems, volume 58 of Springer Tracts in Advanced Robotics. Springer, Berlin, Heidelberg, Germany, 1st edn. (2010)
18. Kröger, T., Finkemeyer, B., Wahl, F.: Manipulation Primitives—A Universal Interface between Sensor-based Motion Control and Robot Programming, volume 67 of Springer Tracts in Advanced Robotics. Springer, Berlin Heidelberg (2011)
19. Kröger, T., Padial, J.: Simple and Robust Visual Servo Control of Robot Arms Using an On-Line Trajectory Generator. In: IEEE International Conference on Robotics and Automation (2012)
20. Kröger, T., Tomiczek, A., Wahl, F.: Towards on-line trajectory computation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Citeseer, Beijing (2006)
21. Larsen, E., Gottschalk, S., Lin, M., Manocha, D.: Fast proximity queries with swept sphere volumes (1999)
22. Liu, S.: An on-line reference-trajectory generator for smooth motion of impulse-controlled industrial manipulators. In: 7th International Workshop on Advanced Motion Control, pp. 365–370 (2002)
23. Mainprice, J., Sisbot, E., Jaillet, L., Cortés, J., Siméon, T., Alami, R.: Planning Human-aware motions using a sampling-based costmap planner. In IEEE International Conference Robotics and Automation (2011)
24. Mainprice, J., Sisbot, E., Siméon, T., Alami, R.: Planning Safe and Legible Hand-over Motions for Human-Robot Interaction (2010)
25. Saut, J.-P.: Efficient models for grasp planning with a multi-fingered hand. Robot. Auton. Syst. **60**, 347–357 (2012)

26. Sidobre, D., Broquère, X., Mainprice, J., Burattini, E., Finzi, A., Rossi, S., Staffa, M.: Human-robot interaction. Advanced Bimanual Manipulation, pp. 123–172 (2012)
27. Sisbot, E., Ros, R., Alami, R.: Situation assessment for human-robot interactive object manipulation. In: 20th IEEE International Symposium on Robot and Human Interactive Communication (2011)
28. Sisbot, E.A., Marin-Urias, L.F., Alami, R., Siméon, T.: Spatial reasoning for human-robot interaction. In: IEEE/RSJ International Conference on Intelligent Robotics and Systems, San Diego, CA, USA (2007)
29. Sisbot, E. A., Urias, L. F. M., Alami, R., Siméon, T.: Spatial reasoning for human-robot interaction. In IEEE/RSJ International Conference on Intelligent Robotics and Systems, IROS, San Diego, CA, USA, (2007)
30. Strabala, K.W., Lee, M.K., Dragan, A.D., Forlizzi, J.L., Srinivasa, S., Cakmak, M., Micelli, V.: Towards seamless human-robot handovers. J. Hum. Robot Interaction **2**(1), 112–132 (2013)
31. Vakanski, A., Mantegh, I., Irish, A., Janabi-Sharifi, F.: Trajectory learning for robot programming by demonstration using hidden markov model and dynamic time warping. Systems, man, and cybernetics, part B. IEEE Trans. Cybern. **42**(4), 1039–1052 (2012)

Chapter 10
Output-Feedback Dynamic Control over Packet-Switching Networks

Stefano Falasca, Massimiliano Gamba and Antonio Bicchi

Abstract When trying to stabilise a dynamical system under the assumption that every communication among the sensors, the actuators and the controller is carried out via a shared communication channel, network-induced constraints come into play. Among such constraints, we address: variable transfer intervals, time varying, large communication delays; non-simultaneous access to the network. In this paper, we devise a method for using an output-feedback dynamic controller whose design is carried out without taking into account the presence of the network. The stability of the resulting nonlinear networked control system is assessed. In order to corroborate the validity of the presented approach, the results of three experiments are presented. Each experiment is carried out using an Ethernet network as the communication medium. One of the experiments involves a real plant, while the remaining have been carried out with simulated plants.

Keywords Control over network · Nonlinear control systems · Output feedback

10.1 Introduction

The past decade has witnessed a dramatic growth in interest for control over distributed networked architectures, which have the strong potential to increase flexibility and scalability of a plant, while inducing a remarkable reduction of costs for both installation and maintenance. However, because of the use of the network and because of the system being distributed, some problems arise: e.g. bandwidth limitations, time-delays and packet losses, which cannot be ignored in the control design. The state-of-the-art is reported and discussed in [1].

An essential aspect of many Networked Control Systems (NCS), such as those using Ethernet as a communication layer, is that they organise data transmission

S. Falasca · M. Gamba · A. Bicchi (✉)
Research Center "E. Piaggio", Largo Lazzarino 1, 56126 Pisa, Italy
e-mail: bicchi@centropiaggio.unipi.it

A. Bicchi
Istituto Italiano di Tecnologia, Genova, Italy

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_10

in packets. Such networks carry larger amount of data with less predictable rates with respect to circuit-switching communication channels. Packetised transmissions substantially alters the bandwidth/performance trade-off of traditional design. The potentially large size of packet payload can be exploited to reduce data transmissions without degrading the overall NCS performance. Bemporad [2] pioneered the idea of sending feed-forward control sequences computed in advance on the basis of a model-based predictive (MBP) scheme to the aim of compensating for large delays in communication channels. The technique has been generalised to address time-varying delays and transfer intervals in [3].

In [4] a control strategy (namely *Packet-Based Control*) for stabilising an uncertain nonlinear NCS affected by varying transmission intervals, varying large delays and constrained access to the network is presented. A model of the plant is used to build a prediction of the control law. Feedback is provided by measuring the plant state. The state is measured by distributed sensors. A network protocol is in charge of deciding which sensor node can communicate at each instant. The control sequence sent by the remote controller is stored in an embedded memory on the plant side, a control command in the sequence is chosen based on the time stamp contained in the packet.

One major limitation of [4] is that it only applies to static-feedback controllers. The stabilisation of NCSs by means of dynamic controllers has been considered in [5]—where it is addressed under the assumption of small delays—and in [3]—where the authors solve the problem in the assumption that all the plant state is sent simultaneously. In this paper we aim at extending the *Packet-Based Control* to dynamic controllers, hence allowing for large delays and non-simultaneous transmissions. Indeed, a direct application of the aforementioned framework to the use a dynamic controller would require updating the internal state of both the system model and the controller by means of a protocol ensuring some nice error-decreasing properties.

We depart from the basic idea of updating the internal state of the dynamic controller in a way consistent with the behavior it would have had if it were directly connected to the plant. We then devise a method which produces the same effects as having a controller on the plant side which sends its internal state through the network towards the remote controller, the same way the state of the plant is sent. We consider the effects of our algorithm as virtual sendings. The only information we need to consistently run the controller is the history of the inputs to the controller. The virtual transmission of the controller state is hence realised by sending the history of the outputs of the plant and by using these outputs to feed the remote controller. The output history can be sent—again—by exploiting the large payload of packets. If output sensors are distributed, outputs are partitioned and the history of each sensor is sent according to a static protocol similar to Round Robin. The drawback of the virtual sendings—especially in the case of outputs partitioned over many nodes—is that a potentially large delay on the arrival of the virtual packets is introduced. Such a delay has to be directly taken into account in the conditions ensuring the stability of the overall system. We prove the exponential stability of the NCS over a prescribed basin of attraction, provided that some explicit bounds on the Maximum Allowable Delay (MAD [1]) and on the Maximum Allowable Transfer Interval (MATI [6]) are satisfied. We finally apply our technique to the control of a magnetic levitator and of a Furuta pendulum involving an output-feedback dynamic controller. It will be shown that if the proposed technique is not used, the network strongly affects the behavior of the NCS. On the other hand, the presented algorithm closely mimic the ideal closed-loop behavior; in accordance with the paradigm adopted, i.e. the presence of the network must be as transparent as possible to the designer of the stabilising controller.

10.2 System Description

In this section we provide the reader with all the characteristics we assume the controller, the network and the plant to have. Figure 10.1 shows the control architecture. The plant and the controller communicate via a shared-bus communication network. An output-feedback dynamic control law for the system is assumed to be available. The plant is equipped with network-enabled devices for actuation and sensing. The sensors measure the internal state of the system and its output. A protocol grants access to the network to one node at a time. Output-measuring sensors send the whole history of readings in tranches. State-measuring sensors send the current reading each time they are granted access to the network.

On the controller side, upon reception of output data, the exact knowledge of the control law is exploited in order to infer a suitable value for the internal state of the

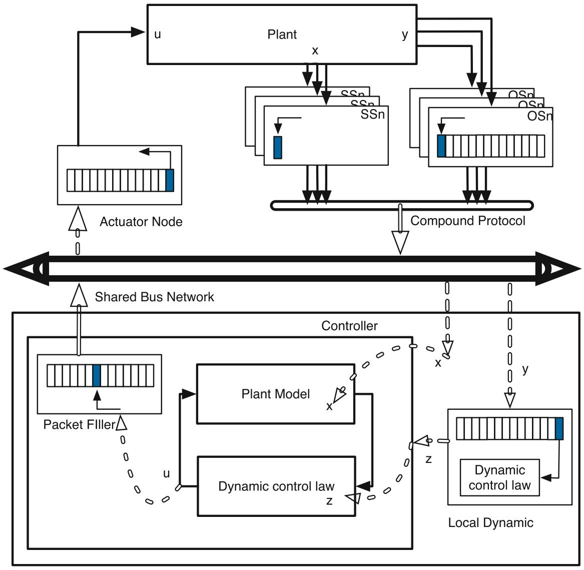


Fig. 10.1 The proposed control architecture

controller at a given time. The so computed internal state of the controller, together with the received information about the internal state of the plant, is used to initialise a model for the *ideal closed loop* composed by a model for the plant dynamics and the control law. By means of simulating the *ideal closed loop* behavior, a sequence of control actions is computed, which is intended to be used in the future. The control actions are then sent to the plant; they will be received on the actuator side and used appropriately.

10.2.1 The Plant and the Controller

We address the stabilisation of a nonlinear continuous-time system of the form

$$\dot{x}_p = f_p(x_p, u) \tag{10.1}$$

$$y = g_p(x_p), \tag{10.2}$$

where $x_p : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_p}$ is the plant state, $y : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_y}$ is the output, $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_u}$ represents the control input, and $f_p : \mathbb{R}^{n_p} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_p}$ and $g_p : \mathbb{R}^{n_p} \to \mathbb{R}^{n_y}$ denote locally Lipschitz functions. For this system, we assume that a nominal dynamic feedback controller of the form

$$\dot{x}_c = f_c(x_c, y) \tag{10.3}$$

$$u = g_c(x_c, x_p, y) \tag{10.4}$$

is available. Here $x_c : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n_c}$ is the controller state, and $f_c : \mathbb{R}^{n_c} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_c}$ and $g_c : \mathbb{R}^{n_c} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_u}$ denote locally Lipschitz functions. Letting $x(t) \triangleq \left[x_p^T(t), x_c^T(t)\right]^T \in \mathbb{R}^{n_p+n_x} = \mathbb{R}^n$ and

$$f(x,u) \triangleq \begin{bmatrix} f_p(x_p, u) \\ f_c(x_c, g_p(x_p)) \end{bmatrix}$$

$$g(x) \triangleq g_c(x_c, x_p, g_p(x_p)),$$

the closed-loop system (10.1)–(10.4) in the absence of network effects simply reads

$$\dot{x} = f(x, u) \tag{10.5}$$

$$u = g(x). \tag{10.6}$$

We assume that the nominal controller (10.3)–(10.4) globally exponentially stabilises the plant (10.1)–(10.2) in the absence of network effects.

Assumption 1 (*Nominal GES*) The origin of the system (10.1)–(10.2) in closed-loop with (10.3)–(10.4) is globally exponentially stable (GES) and there exists a differentiable function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ and constants $\underline{\alpha}, \overline{\alpha}, \alpha, d > 0$ such that the following conditions hold for all $x \in \mathbb{R}^n$

$$\underline{\alpha} \|x\|^2 \leq V(x) \leq \overline{\alpha} \|x\|^2$$

$$\frac{\partial V}{\partial x}(x) f(x, g(x)) \leq -\alpha \|x\|^2$$

$$\left\| \frac{\partial V}{\partial x}(x) \right\| \leq d \|x\| .$$

Local Lipschitz constants are assumed to be available to the designer.

Assumption 2 (*Local Lipschitz*) Given some constants $R_x, R_u > 0$, there exist some constants $\lambda_f, \lambda_\kappa > 0$ and all $u_1, u_2 \in B_{R_u}$ such that the following inequalities hold

$$\|f(x_1, u_1) - f(x_2, u_2)\| \leq \lambda_f (\|x_1 - x_2\| + \|u_1 - u_2\|) \qquad (10.7)$$

$$\|g(x_1) - g(x_2)\| \qquad \leq \lambda_\kappa \|x_1 - x_2\| . \qquad (10.8)$$

The control strategy analysed in this paper aims at compensating the network-induced effects by relying on a prediction of the plant behavior. To that aim, we assume that a model for (10.1)–(10.2) is known:

$$\dot{\hat{x}}_p = \hat{f}_p(\hat{x}_p, \hat{u}) \qquad (10.9)$$

$$\hat{y} = \hat{g}_p(\hat{x}_p). \qquad (10.10)$$

This model in closed-loop with the nominal controller (10.3)–(10.4) reads

$$\dot{\hat{x}} = \hat{f}(\hat{x}, \hat{u}) \qquad (10.11)$$

$$\hat{u} = \hat{g}(\hat{x}), \qquad (10.12)$$

where $\hat{x} \triangleq (\hat{x}_p^T, \hat{x}_c^T)^T : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ and

$$\hat{f}(\hat{x}, \hat{u}) \triangleq \begin{bmatrix} \hat{f}_p(\hat{x}_p, \hat{u}) \\ f_c(\hat{x}_c, \hat{g}_p(\hat{x}_p)) \end{bmatrix}$$

$$\hat{g}(\hat{x}) \triangleq g_c(\hat{x}_c, \hat{x}_p, \hat{g}_p(\hat{x}_p)).$$

The plant-model inaccuracy is assumed to be sector-bounded.

Assumption 3 (*Sector-Bounded Model Inaccuracy*) Given $R_x, R_u > 0$, there exists a constant $\lambda_{f\hat{f}} \geq 0$ such that, for all $x \in B_{R_x}$ and all $u \in B_{R_u}$,

$$\left\| f(x, u) - \hat{f}(x, u) \right\| \leq \lambda_{f\hat{f}} (\|x\| + \|u\|) . \qquad (10.13)$$

10.2.2 The Network

Measurements are taken by distributed sensors and sent to the controller into packets. Sensors are assumed to be synchronised with each other. We assume that the measurement part of the network is partitioned into ℓ nodes and only a *unique* node at a time can send its information (i.e. only partial knowledge of the plant state is available at each time instant). The controller is required to be a unique node. The overall state of the system $x(t) \in \mathbb{R}^n$ is thus decomposed into $\ell + 1$ nodes as $x(t) = [x_{p,1}^T(t), \ldots, x_{p,\ell}^T(t), x_c^T(t)]^T$ with $x_{p,i}(t) \in \mathbb{R}^{p_i}$ and $\sum_{i=1}^{\ell} p_i = n_p$.

Control sequences are sent as packets. An embedded control device receives, decodes, synchronises these packets and applies control commands to the plant. We consider that measurements are taken and sent at instants $\{\tau_i^m\}$, and are received by the remote controller at instants $\{\tau_i^m + T_i^m\}$. In other words, $\{T_i^m\}$ denotes the sequence of (possibly time-varying) measurement data delays. Delays cover both processing time and transmission delays on the measurement chain. Similarly, control commands are sent over the network at time instants $\{\tau_i^c\}$. They reach the plant at instants $\{\tau_i^c + T_i^c\}$, where $\{T_i^c\}$ denotes the sequence of delays accounting for both the computation time and the transmission delay from the remote controller to the plant.

Assumption 4 (*Network*) The communication network satisfies the following properties:

(i) **(MATI)** There exist two constants $\tau^m, \tau^c \in \mathbb{R}_{\geq 0}$ such that $\tau_{i+1}^m - \tau_i^m \leq \tau^m$ and $\tau_{i+1}^c - \tau_i^c \leq \tau^c, \forall i \in \mathbb{N}$;

(ii) **(mTI)** There exist constants $\varepsilon^m, \varepsilon^c \in \mathbb{R}_{\geq 0}$ such that $\varepsilon^m \leq \tau_{i+1}^m - \tau_i^m$ and $\varepsilon^c \leq \tau_{i+1}^c - \tau_i^c, \forall i \in \mathbb{N}$.

(iii) **(MAD)** There exist two constants $T^m, T^c \in \mathbb{R}_{\geq 0}$ such that $T_i^m \leq T^m$ and $T_i^c \leq T^c, \forall i \in \mathbb{N}$;

10.2.3 The Network Protocol

The use of a *dynamic* controller imposes a careful update of the controller internal model in order to generate meaningful control sequences. We propose a strategy that consists in transmitting the measurement history of each output nodes over a prescribed time horizon, as well as the instantaneous value of the plant's state when access is granted to the network. The system thus involves two different kinds of sensor nodes: ℓ_y output-sending nodes (OSn) and ℓ state-sending nodes (SSn).

The access to the network is ruled by a protocol choosing, at each instant τ_i^m, which node communicates its data. In order to limit the cumulated delays induced by this approach, we assume that the nodes are granted access to the network according to the following rule: after each SSn access, all OSn are required to send their data according to a prescribed ordering (Round Robin). Then access is again granted to a

Table 10.1 The sequences $\{s_i\}$ and $\{o_i\}$ for $\ell_y = 3$

s_0	o_0	o_1	o_2	s_1	o_3	o_4	o_5	s_2	…
0	1	2	3	4	5	6	7	8	…

SSn, and so on. This rule can be formally stated by extracting from the sequence of access times $\left\{\tau_i^m\right\}$ two subsequences $\left\{\tau_{o_i}^m\right\}$ and $\left\{\tau_{s_i}^m\right\}$. More precisely, we define two sequences $\{s_i\}, \{o_i\}$ having values in $\mathbb{N}$. Such sequences have the following meaning: at time τ_s^m, $s \in \{s_i\}$ a SSn is granted access to the network; at time τ_o^m, $o \in \{o_i\}$ an OSn has the ability to send. The policy is such that the two sequences exhibit the following properties

(a) $\{s_i\} \cup \{o_i\} = \mathbb{N}, \ \{s_i\} \cap \{o_i\} = \emptyset$;
(b) $s_i = (\ell_y + 1)i$;
(c) o_i is strictly increasing.

Consider, as an example, $\ell_y = 3$; the sequences $\{s_i\}$ and $\{o_i\}$ are shown in Table 10.1.

We keep track of which OSn is granted access to the network at a given time by means of the definition of the sequence $\{\nu_i\}$ having values in $\left[1, \ell_y\right] \subset \mathbb{N}$ defined as

$$\nu_i = i \mod \ell_y + 1 \tag{10.14}$$

The ν_i-th OSn is thus granted the access to the network at time $\tau_{o_i}^m$.

The SSn are granted access to the network according to a protocol ruled by the map involving the error $e_p(t) \in \mathbb{R}^{n_p}$ defined as $e_p(t) \triangleq \hat{x}_p(t) - x_p(t)$:

$$e_p(\tau_{s_i}^{m+}) = h_p\left(i, e_p(\tau_{s_i}^m)\right), \quad \forall i \in \mathbb{N} \tag{10.15}$$

where $h_p : \mathbb{N} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_p}$. This protocol is assumed to induce an exponential decrease of the error e_p when the inter-sample dynamics are neglected; i.e. we are interested in UGES protocols. We recall here a slightly modified version of the definition in [7] as given in [4].

Definition 1 A function $h : \mathbb{N} \times \mathbb{R}^n \to \mathbb{R}^n$ is said to be an UGES protocol having parameters $\underline{a}, \overline{a}, \rho, c$ if there exists a function $W : \mathbb{N} \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ locally Lipschitz in its second argument and there exist constants $\underline{a}, \overline{a} > 0$; $c > \underline{a}$ and $\rho \in [0, 1)$ such that the following conditions hold for the auxiliary discrete time system $\xi(i+1) = h(i, \xi(i))$:

$$\begin{gathered} \underline{a}\, \|\xi\| \leq W(i, \xi) \leq \overline{a}\, \|\xi\| \\ W(i+1, h(i, \xi)) \leq \rho W(i, \xi) \end{gathered} \tag{10.16}$$

for all $\xi \in \mathbb{R}^n$ and all $i \in \mathbb{N}$, and

$$\left\| \frac{\partial W}{\partial \xi}(i, \xi) \right\| \leq c \tag{10.17}$$

for almost all $\xi \in \mathbb{R}^n$ and all $i \in \mathbb{N}$.

Assumption 5 The protocol (10.15) is UGES with parameters $\underline{a}_p, \overline{a}_p, \rho_p, c_p$.

Remark 1 The UGES protocol class is often used in the network control literature. Although it might seem to be very conservative, it has to be stressed out that it is not. Indeed, no limits are being posed on the rate of convergence of the error. The network protocol is used as a control-oriented description of the effects of the sending order (which is not necessarily pre-defined). UGES network protocols which are often used in practice are the Round Robin and the so called *Maximum Error First - Try Once Discard*.

10.3 Algorithm Description

The algorithm we propose can be decomposed in different modules. At the plant side of the network, three kinds of *devices* are needed, namely the actuator node, the state-sending node and the output-sending node. The controller is divided into two modules: the first one—namely *Local Dynamics*—is in charge of converting the output data it receives from the plant into information about the internal state the controller is supposed to have; the second one has state-related information as an input (either received from the network or produced by the *Local Dynamics*); based on its input it computes the controls to be sent to the actuator node.

In this section we describe the behavior of each module and provide a model for the overall closed-loop system.

10.3.1 The Plant

10.3.1.1 Actuator Node

Such node is in charge of receiving, decoding and re-synchronise packets sent by the controller. Each received packet contains a timestamp and a certain number of control values which are stored in a local buffer. Upon choosing which one of the control values is the most appropriate, it actuates the plant. More precisely, the actuator node compares the timestamp of the last packet it received with its internal clock and moves within the control sequence it received up to the corresponding starting point.

10.3.1.2 State-sensor Node

When such a node is granted access to the network (see Sect. 10.2.3 for a description of the policy used to take this decision), it encodes the sensed values into a network packet, timestamps it and sends it to the controller.

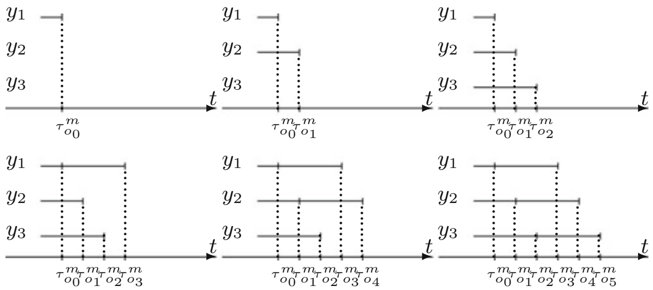

y_1
y_2
y_3
t
$\tau^m_{o_0}$
$\tau^m_{o_0}\tau^m_{o_1}$
$\tau^m_{o_0}\tau^m_{o_1}\tau^m_{o_2}$
$\tau^m_{o_0}\tau^m_{o_1}\tau^m_{o_2}\tau^m_{o_3}$
$\tau^m_{o_0}\tau^m_{o_1}\tau^m_{o_2}\tau^m_{o_3}\tau^m_{o_4}$
$\tau^m_{o_0}\tau^m_{o_1}\tau^m_{o_2}\tau^m_{o_3}\tau^m_{o_4}\tau^m_{o_5}$

Fig. 10.2 Example of OSn sendings for $\ell_y = 3$

10.3.1.3 Output-sensor Node

An output-sensor node continuously monitors the sensed outputs, storing the readings into a buffer. The buffer content is timestamped and encoded into packets sent to the controller when the network is available. Upon sending, data is discarded. The sending of tranches of data is consistent with how control data is sent in [4]. Figure 10.2 shows an example of the sendings carried out by the output-sensor nodes.

Remark 2 It is tacitly assumed that the effects of sending discrete-time values instead of continuous function to the controller and to the actuator node can be neglected.

10.3.2 The Controller

10.3.2.1 Local Dynamics

By exploiting output readings it is possible to mathematically consider the controller as being executed on the plant side and sending its internal state over the network. We will define the sequence of time-instants at which such virtual-sendings happen and the delay each of these packets incurs into. Finally, we will use such virtual packets, together with the state packets, in order to design an overall UGES protocol acting on the complete system as expressed in (10.5).

First of all, we need to give some definitions.

Definition 2 Given $T \in \mathbb{R}_{\geq 0}$, we will say that $y_{[0,\tau]\,l}$ is known at time $\tau + T$ if there exist $j \in \{o_i\}$, $k \in \mathbb{N}$ such that:

(a) $\tau^m_j \geq \tau,\ \tau^m_j + T^m_j \leq \tau + T$;
(b) $o_k = j,\ \nu_k = l$.

Moreover, we will say that $y_{[0,\tau]}$ is known at time $\tau + T$ if $\forall l \in [1, \ell_y] \subset \mathbb{N}$ we have that $y_{[0,\tau]\,l}$ is known at time $\tau + T$.

This definition reflects the natural notion of the controller having already received a packet related to y_l which was sent no later than time τ.

Definition 3 Given $T \in \mathbb{R}_{\geq 0}$, we will say that $x_c(\tau)$ is known at time $\tau + T$ if either $\tau = 0$ or $y_{[0,\tau]}$ is known to the controller at time $\tau + T$.

This definition formalises the fact that, given σ such that $x_c(\sigma)$ is known at time $\tau + T$, it is possible to use the solution for the differential equation $\phi_{f_c}^{[\sigma,\tau]} : [\sigma, \tau] \times [\sigma, \tau] \times \mathbb{R}^{n_c} \times \mathbb{R}^{n_y}_{[\sigma,\tau]} \to \mathbb{R}^{n_c}$ in order to compute $x(\tau)$, since

$$x_c(\tau) = \phi_{f_c}^{[\sigma,\tau]} \left(\tau, \sigma, x_c(\sigma), y_{[\sigma,\tau]}\right). \tag{10.18}$$

With the previous definitions in mind we can state the following.

Proposition 1 (Virtual Packets) *Given $i \geq \ell_y - 1, i \in \mathbb{N}$ the value $x_c\left(\tau^m_{o_{(i-(\ell_y-1))}}\right)$ is known at time $\tau^m_{o_{(i-(\ell_y-1))}} + \ell_y \tau^m + T^m$.*

This means that we can consider that at time $\tau^m_{o_{(i-(\ell_y-1))}}$ a packet containing $x_c\left(\tau^m_{o_{(i-(\ell_y-1))}}\right)$ is sent by the plant; such a virtual packet incurs a delay which is no longer than $\ell_y \tau^m + T^m$.

We will consider those virtual packets in conjunction with the packets containing $x_s(\tau^m_{s_i})$, which arrive at the controller at time $\tau^m_{s_i} + T^m_{s_i}$. The sequence of time instants at which such packets are sent is $\left\{\tau^m_i\right\}$. As for the delays, we define a new sequence $\left\{T^f_i\right\}$.

Definition 4 (*Sequence of state-sending delays*) The sequence of state-sending delays $\left\{T^f_i\right\}$, $T^f_i \in \mathbb{R}_{\geq 0}$ such that:

$$T^f_i = \begin{cases} T^m_i & \text{if } \exists\, k : i = s_k \\ \min\left\{\mathcal{K}_{\tau^m_i}\right\} & \text{otherwise} \end{cases} \tag{10.19}$$

where

$$\mathcal{K}_\tau \triangleq \{T : x_c(\tau) \text{ is known at time } \tau + T\}. \tag{10.20}$$

By virtue of Proposition 1, the following inequality holds:

$$T^f_i \leq \ell_y \tau^m + T^m = T^f. \tag{10.21}$$

From now on we will consider the packets containing the state of the system and the virtual packets containing the state of the controller. The information they gather will be used in order to design a protocol which acts on the error $e(t) = (e^T_p(t), e^T_c(t))^T \triangleq \hat{x}(t) - x(t)$, where $e_c(t) \in \mathbb{R}^{n_c} : e_c(t) \triangleq \hat{x}_c(t) - x_c(t)$. We will show that, provided that an UGES protocol acting on e_p is in use (see Assumption 5), the designed protocol is UGES.

Proposition 2 (Compound Protocol) *The function* $h : \mathbb{N} \times \mathbb{R}^n \to \mathbb{R}^n$

$$\left(i, \begin{bmatrix} \xi_p \\ \xi_c \end{bmatrix}\right) \mapsto \begin{bmatrix} \begin{cases} h_p\left(k, \xi_p\right) & \text{if } \exists\, k : s_k = i \\ \xi_p & \textit{otherwise} \end{cases} \\ \begin{cases} \xi_c & \text{if } \exists\, k : s_k = i \\ 0 \in \mathbb{R}^{n_c} & \textit{otherwise} \end{cases} \end{bmatrix} \tag{10.22}$$

defines an UGES protocol. Here $\xi_p \in \mathbb{R}^{n_p}$, $\xi_c \in \mathbb{R}^{n_c}$.

10.3.2.2 Computing the Control Law

When a new measurement is received, the remote controller uses the new data in order to update an estimate of the internal state of the ideal closed loop system. The controller then computes a prediction of the control signal over a fixed time horizon

$$T_0^p \geq T^c + T^f + \tau^m + \tau^c \tag{10.23}$$

by numerically running the model (10.11)–(10.12). Such computation generates values for the function $\hat{u}(t)$ (cf. Eq. (10.12)) which are then coded, marked with the appropriate timestamp, and put into a single packet which is sent at the next network access.

10.3.3 The Overall Model

The loop composed of the system (10.1)–(10.2) and the controller node which executes the algorithms described in Sect. 10.3.2 can be modelled by means of the following equations (see also [4]).

The NCS model has a state $x(t)$ which models the internal state of the plant as well as the state of the controller as it would act if it were connected to the outputs of the plant. Moreover, N vectors of additional state variables are used for modelling the estimations of the vector $\hat{x}$. N represents the number of packets, either real or virtual, that can be received by the controller during the time T_0^p. It is defined as

$$N \triangleq \left\lceil \frac{T_0^p - \tau^m}{\varepsilon^m} \right\rceil + 1. \tag{10.24}$$

By defining $\bar{x}(t)$, $\tilde{x}(t)$, $e(t) \in \mathbb{R}^{Nn}$ as $\bar{x}(t) \triangleq \left[x^T(t), \ldots, x^T(t)\right]^T$ and $e(t) \triangleq \left[e_1^T(t), \ldots, e_N^T(t)\right]^T$, $e_i(t) \in \mathbb{R}^n$, the closed-loop dynamics of the NCS can be compactly written as

$$\dot{x} = F(t, \bar{x}, e) \tag{10.25a}$$

$$\dot{e} = G(t, \bar{x}, e) \tag{10.25b}$$

$$e(\tau_i^{m+}) = H(i, e(\tau_i^m)), \tag{10.25c}$$

where

$$F(t, \bar{x}, e) = f(x, \upsilon(t, e + \bar{x})) \tag{10.26a}$$

$$G(t, \bar{x}, e) = \begin{bmatrix} \hat{f}(e_1 + x, \hat{g}(e_1 + x)) - f(x, \upsilon(t, e + \bar{x})) \\ \vdots \\ \hat{f}(e_N + x, \hat{g}(e_N + x)) - f(x, \upsilon(t, e + \bar{x})) \end{bmatrix} \tag{10.26b}$$

$$H(i, e) = \begin{bmatrix} e_1 + (h(i, e_N) - e_1)\,\eta(i, 1) \\ e_2 + (h(i, e_1) - e_2)\,\eta(i, 2) \\ \vdots \\ e_N + (h(i, e_{N-1}) - e_N)\,\eta(i, N) \end{bmatrix}, \tag{10.26c}$$

where $\eta : \mathbb{N} \times \{1, \ldots, N\} \to \{0, 1\}$ identifies the index of the relevant state estimate

$$\eta(i, r) \triangleq \begin{cases} 1 \text{ if } \mu(i) = r \\ 0 \text{ otherwise} \end{cases} \tag{10.27}$$

and $\mu : \mathbb{N} \to \{1, \ldots, N\}$ is defined as

$$\mu(i) \triangleq ((i - 1) \bmod N) + 1\ . \tag{10.28}$$

The control signal υ in (10.26a) and (10.26b) is defined as $\upsilon : \mathbb{R}_{\geq 0} \times \mathbb{R}^{Nn} \to \mathbb{R}^{n_u}$,

$$(t, \left[\xi_1^T, \ldots, \xi_N^T\right]^T) \mapsto \sum_{k=1}^{N} \hat{g}(\xi_k)\zeta(t, k) \tag{10.29}$$

where $\xi_i \in \mathbb{R}^n$ and $\zeta : \mathbb{R}_{\geq 0} \times \{1, \ldots, N\} \to \{0, 1\}$ is the map

$$(t, k) \mapsto \begin{cases} 1 & \text{if } \exists j \in \mathbb{N} \text{ s.t. } \mu(\gamma(j)) = k \text{ and} \\ & t \in (\tau_j^c + T_j^c, \tau_{j+1}^c + T_{j+1}^c] \\ 0 & \text{otherwise} \end{cases} \tag{10.30}$$

and $\gamma : \mathbb{N} \to \mathbb{N}$

$$j \mapsto \max\left\{i \in \mathbb{N} \mid \tau_i^m + T_i^f < \tau_j^c\right\} \tag{10.31}$$

denotes the index of the latest measurement received before τ_j^c.

10.4 Main Result

Theorem 1 *Assume that Assumptions 1, 4, 5 hold. Given some $R > 0$, fix $R_x = R$ and $R_u = \lambda_k R$ and suppose that assumptions 2 and 3 hold with these constants. Let $\underline{a}_p, \overline{a}_p, \rho_p, c_p$ $\underline{\alpha}, \overline{\alpha}, \alpha, d, \lambda_{f\hat{f}}, \lambda_f$ and λ_k be generated by these assumptions. Pick*

$$\underline{a} = \underline{a}_p \rho_p,\ \overline{a} = \overline{a}_p,\ \rho = \rho_p^{\frac{1}{\ell_y+1}},\ c = c_p \tag{10.32}$$

and define $a_H \triangleq \overline{a}$, $a_L \triangleq \frac{\underline{a}}{N} \min\left\{1, \left(\frac{\underline{a}}{\overline{a}}\right)^2 \frac{1}{\rho}\right\}$. Assume that the following conditions on $\tau^m, T^f, \tau^c, T^c, \epsilon^m$ hold:

$$\begin{aligned} &\tau^m \in [\varepsilon^m, \tau^{m*}),\ \tau^{m*} \triangleq \tfrac{1}{L} \log\left(\tfrac{M\gamma_2 + a_L L}{M\gamma_2 + a_L \rho L}\right) \\ &N = \left\lceil \tfrac{T^c + T^f + \tau^c}{\varepsilon^m} \right\rceil + 1 \end{aligned} \tag{10.33}$$

where

$$\begin{aligned} L &\triangleq \tfrac{c}{a_L}\left((1+\lambda_k)\sqrt{N}\lambda_{f\hat{f}} + \sqrt{N}\lambda_f + \left(\sqrt{N-1} + N - 1\right)\lambda_f \lambda_k\right) \\ M &\triangleq (1+\lambda_k)\, cN\lambda_{f\hat{f}} \\ \gamma_2 &\triangleq \tfrac{d}{\alpha}\sqrt{\tfrac{\overline{a}}{\underline{a}}}\lambda_f \lambda_k \end{aligned} \quad . \tag{10.34}$$

Then the origin of the NCS (10.25) is exponentially stable with radius of attraction

$$\tilde{R} \triangleq \frac{R}{K} \tag{10.35}$$

where $K \triangleq \frac{\sqrt{2}}{1-\gamma_1\gamma_2} \max\{(1+\gamma_1)k_2, (1+\gamma_2)k_1\}$, $\gamma_1 \triangleq \frac{M \exp(L\tau^m) - 1}{a_l L(1 - \rho \exp(L\tau^m))}$, $k_1 \triangleq \frac{a_H}{\rho a_L}$ and $k_2 \triangleq \sqrt{\frac{\overline{\alpha}}{\underline{\alpha}}}$.

Conditions expressed in (10.33) establish a relation between the relevant parameters, namely ε^m, T^c, T^f, τ^c and τ^m. Notice that (10.21) can be used to express such a relation in terms of T^m and ℓ_y instead of T^f. Note that since Theorem 1 guarantees only local properties, Assumption 1 could be relaxed to *local* exponential stability of the nominal plant, over a sufficiently large domain.

Remark 3 The presented formulation of the MATI and the expression for the radius of convergence are based on [4] where examples showing that the MATI constitutes an improvement over the previously existing state-of-the-art can be found. It is easily seen that due to the high number of variables involved in the presented expressions, it is impractical to actually compute the MATI and the radius of convergence for a real plant. In fact, the same can be said for most of the similar results which can be found in literature. In this case the presented theorem can be seen as an *existence* result: it assesses that it is possible to stabilise the system by means of the presented architecture.

10.5 Experiments

In this section we show results of experiments with a real Ethernet based network. Both network-in-the-loop experiments with a simulated plant and with a real plant are presented.

The experiments have been carried out by means of a software for networked control systems based on the one presented in [8] which allows for network-in-the-loop tests to be carried. For this purpose a module implementing the *local dynamics* algorithm has been designed. Furthermore, this software can be used to communicate with embedded software which implements the various geographically distributed nodes–actuators and sensors.

Both experiments point out the robustness of the proposed approach to the model uncertainties.

10.5.1 Network-in-the-Loop Experiment–Magnetic Levitator

We provide results of the experiments carried out for the networked control of a magnetic levitator. The setup uses two computers, one for the controller and the other for simulating the plant with sensors and actuators. The computers are connected through a real Ethernet link. The experimental network setup is such that

$$\begin{array}{l} 1 \times 10^{-3}\,[\mathrm{s}] \le \tau^m_{i+1} - \tau^m_i \le 5 \times 10^{-3}\,[\mathrm{s}] \\ 1 \times 10^{-3}\,[\mathrm{s}] \le \tau^c_{i+1} - \tau^c_i \le 5 \times 10^{-3}\,[\mathrm{s}] \end{array}. \tag{10.36}$$

Figure 10.3 shows the measured round trip time. Based on the measurements, we can consider the maximum delays[1] to be $T^m, T^c = \frac{\max \mathrm{RTT}}{2} \approx 26 \times 10^{-3}\,[\mathrm{s}]$.

The plant parameters (equations are shown in Fig. 10.4) are $\alpha = \frac{\pi}{6}$, $g = 9.8 \left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]$, $m = 0.05 \left[\mathrm{Kg}\right]$, $c = 0.5\,[\mathrm{Hm}]$, $L = 1\,[\mathrm{H}]$, $R = 10\,[\Omega]$.

According to Sect. 10.2.1, we assume that a stabilising controller is given for the nominal plant. In our case, the nominal controller is the result of the straightforward application of numerical self-tuning routines in Matlab, and has the transfer function:

[1] The measurements include both the network-induced delays and some additional delays which have been added via software in order to simulate the effects of additional traffic. The program `tc` has been used on both `Linux` hosts to provide additional sending delays, which have a normal probability distribution of $\mathcal{N}(15, 5)$. Hence, the mean value of the added round trip time is 30ms. Additional delays account for the larger portion of the overall measured delay.

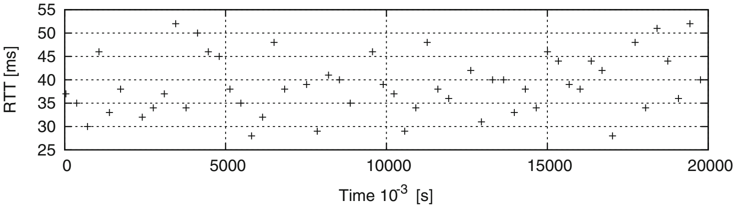
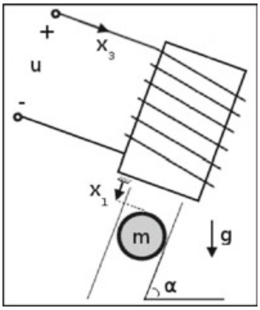


Fig. 10.3 Round trip time



$$\dot{x} = \begin{bmatrix} x_2 \\ \sin(\alpha) g - \frac{c}{m}\left(\frac{x_3}{x_1}\right)^2 \\ -\frac{R}{L}x_3 + \frac{2c}{L}\frac{x_2 x_3}{x_1^2} + \frac{1}{L}u \end{bmatrix}$$

$$y = x_1$$

Fig. 10.4 Magnetic levitator and its model

$$C(s) = -\frac{a_2 s^2 + a_1 s + a_0}{s^2(s+b)} \tag{10.37}$$

where $a_2 = 1{,}418$, $a_1 = 767$, $a_0 = 377$ and $b = 29$.

The plant model used in the controlling computer is subject to parametric uncertainties. The parameters for the model it uses are very different from the real ones, i.e. $\alpha = \frac{\pi}{2}$, $g = 9.8\left[\frac{\text{m}}{\text{s}^2}\right]$, $m = 1\,[\text{Kg}]$, $c = 1\,[\text{Hm}]$, $L = 5\,[\text{H}]$, $R = 0.1\,[\Omega]$.

Figure 10.5 shows the results of the experiments for a reference signal $x_d = 0.05\,[\text{m}]$. One of the trajectories shows the behavior of the ideal closed-loop; the second one shows the networked system. Finally, the behavior of the networked system when the algorithm taking into account local dynamics (cf. Sect. 10.3.2.1) is not used is shown, i.e. the protocol $h : \mathbb{N} \times \mathbb{R}^n \to \mathbb{R}^n$

$$\left(i, \begin{bmatrix} \xi_p \\ \xi_c \end{bmatrix}\right) \mapsto \begin{bmatrix} \begin{cases} h_p(k, \xi_p) & \text{if } \exists\, k : s_k = i \\ \xi_p & \text{otherwise} \end{cases} \\ \xi_c \end{bmatrix}$$

is used. Experiments show that the proposed algorithm manages to produce results resembling the ideal closed loop behavior. If the proposed algorithm is not used, the behavior is altered; for instance—for the given example—the steady state error is not zero.

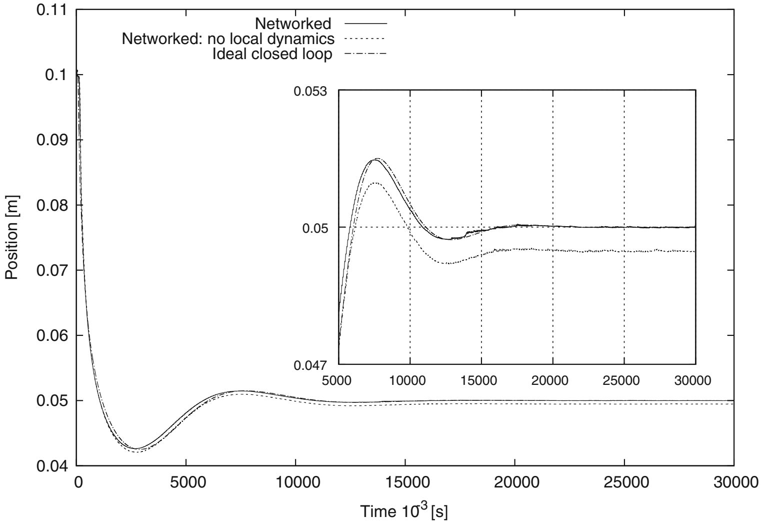


Fig. 10.5 Trajectory $x_{1p}(t)$

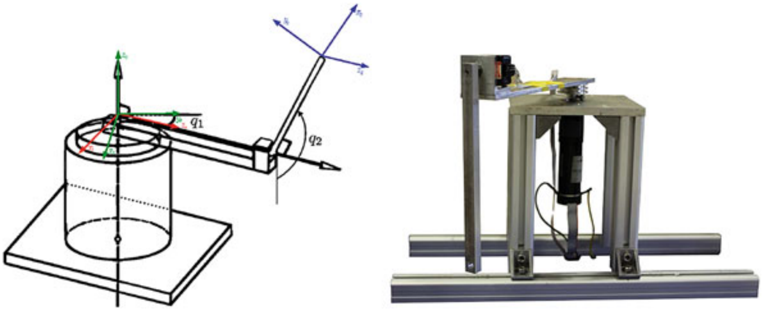


Fig. 10.6 Furuta pendulum

10.5.2 Real System—Furuta Pendulum

The PBC approach has been experimented with on a real plant. These experiments assess the real-world applicability of the approach. A comparison between netwoked control and local control is shown.

The test-bed is the Furuta pendulum is represented in Fig. 10.6 and its parameters are contained in Table 10.2.

The vector $q = [q_1, q_2]^T$ describes the vector of the state variables: q_1 is the angular position of the arm and q_2 is the angular position of the pendulum. The system is under-actuated, meaning that only the arm joint is actuated by means of the torque τ. The dynamics of the nonlinear model for the plant is given by:

Table 10.2 Parameters of Furuta pendulum

Physical quantity	Symbol	Value	Units
Arm mass	m_1	200×10^{-3}	kg
Pendulum mass	m_2	72×10^{-3}	kg
Arm length	L_1	224×10^{-3}	m
Arm COM	l_1	144×10^{-3}	m
Pendulum COM	l_2	106×10^{-3}	m
Arm z_0 inertia	J_{z_0}	0.9×10^{-3}	kg m^2
Pendulum x_2 inertia	J_{x_2}	1.65×10^{-6}	kg m^2
Pendulum y_2 inertia	J_{y_2}	2.7×10^{-4}	kg m^2
Pendulum z_2 inertia	J_{z_2}	2.71×10^{-4}	kg m^2
Arm friction	c_1	0.9×10^{-2}	N m s
Pendulum friction	c_2	2.71×10^{-7}	N m s
Motor torque constant	K	2.2274	N m A^{-1}
Motor inductance	L_a	0.044	H
Motor resistance	R_a	1.9	Ω

$$\begin{bmatrix} \pi_1 + \pi_2 \sin^2 q_2 + \pi_3 \cos^2 q_2 & \pi_4 \cos q_2 \\ \pi_4 \cos q_2 & \pi_7 \end{bmatrix} \ddot{q} +$$
$$\begin{bmatrix} \pi_6 + \pi_5 \dot{q}_2 \sin 2q_2 - \pi_4 \dot{q}_2 \sin q_2 + \pi_5 \dot{q}_1 \sin 2q_2 \\ -\pi_5 \sin(2q_2)\dot{q}_1 & \pi_8 \end{bmatrix} \dot{q} + \begin{bmatrix} 0 \\ \pi_9 \sin q_2 \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}, \tag{10.38}$$

where the quantities π_i represent the dynamic parameters of the system, which are defined, according to the mechanical parameters in Table 10.2, as follows:

$$\begin{array}{lll} \pi_1 = J_{z_0} + m_1 l_1^2 + m_2 L_1^2 & \pi_2 = J_{y_2} + m_2 l_2^2 & \pi_3 = J_{x_2} \\ \pi_5 = \frac{1}{2}\left(J_{y_2} - J_{x_2} + m_2 l_2^2\right) & \pi_4 = m_2 L_1 l_2 & \pi_6 = c_1 \\ \pi_7 = J_{z_2} + m_2 l_2^2 & \pi_8 = c_2 & \pi_9 = m_2 l_2 g \end{array}$$

where g is the gravity.

The dynamics of the torque τ are described by the following first order linear dynamics, representing the model of a DC motor:

$$L_a \dot{\tau} = KV - R_a \tau - K^2 \dot{q}_1 \tag{10.39}$$

where V is the voltage applied to the motor and K, L_a, R_a are the motor parameters described in Table 10.2.

The control law has been taken from [9]. The stabilisation around the unstable equilibrium point is achieved by means of a state observer plus a state feedback;

those have been designed based on the linearised system. A strategy for bringing the pendulum to the upright position is also implemented in [9]—it is based on the work in [10].

The experimental setup uses a computer for the controller, whilst the sensor and the actuator nodes are implemented in dedicated embedded hardware. The shared communication network consists in Ethernet links with a star topology. The experimental network setup is such that $1 \times 10^{-3}\,\mathrm{s} \leq \tau^m_{i+1} - \tau^m_i \leq 10 \times 10^{-3}\,\mathrm{s}$ and $1 \times 10^{-3}\,\mathrm{s} \leq \tau^c_{i+1} - \tau^c_i \leq 10 \times 10^{-3}\,\mathrm{s}$. Based on the measurements, we can consider the maximum delays to be $T^m, T^c = \frac{\max \mathrm{RTT}}{2} \approx 8 \times 10^{-3}\,\mathrm{s}$, where RTT is the packet round trip-time. The delays are induced by the Ethernet network and by computation overhead.

The goal of the control is to keep the pendulum rod in the upright position ($q_2 = 2k\pi$ rad, $k \in \mathbb{Z}$ in Fig. 10.7). The initial condition for the plant is near the equilibrium point, i.e. 10° from the downright position and all the other state variables set to zero.

Figure 10.7 shows the results of the experiments, the two trajectories represent the behavior of the controlled pendulum.

The results contained in Fig. 10.7 show the behavior of the pendulum controlled both in local and PBC fashion.

Experiments show that the proposed algorithm is able to resemble the local control.

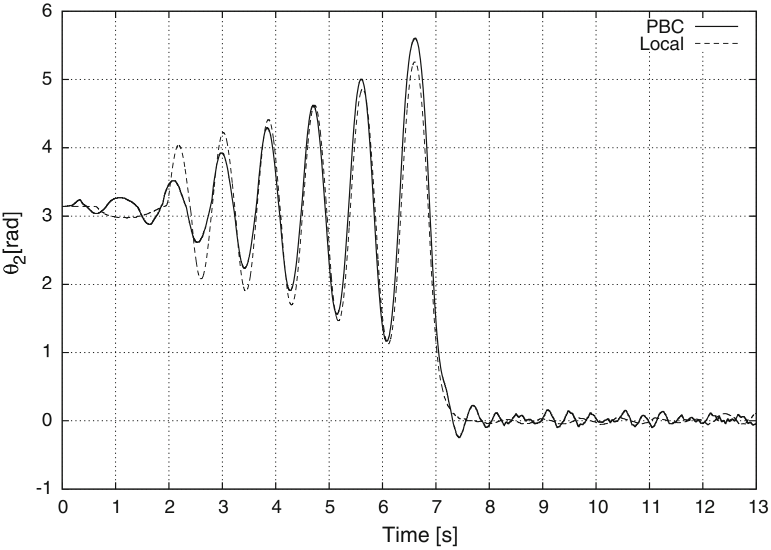


Fig. 10.7 Trajectories of the pendulum rod

10.6 A Special Case

In this section, some simplifying assumptions are made. Under such assumptions, some interesting conclusions will be drawn—namely, we will see that measuring the state is not needed. Of course, the whole content of this section is not valid when the additional assumptions are not met.

Suppose that $\hat{f}_p\left(x_p, u\right) \equiv 0$ and $\hat{g}_p\left(x_p\right) \equiv g_p\left(x_p\right)$. Under such hypothesis, Eqs. (10.9) and (10.10) read respectively:

$$\dot{\hat{x}}_p = 0 \tag{10.40}$$

$$\hat{y} = g_p(\hat{x}_p) \tag{10.41}$$

and of course $\dot{\hat{y}} = 0$. Suppose also that the dynamic control law (10.3)–(10.4) is an output feedback, namely:

$$\dot{x}_c = f_c(x_c, g_p(x_p)) \tag{10.42}$$

$$\hat{u} = g_c(x_c, g_p(x_p)). \tag{10.43}$$

Hence, the closed loop between the model and the controller—(10.11) and (10.12)—becomes:

$$\dot{\hat{x}} = \begin{bmatrix} 0 \\ f_c(\hat{x}_c, g_p(\hat{x}_p)) \end{bmatrix} \tag{10.44}$$

$$\hat{u} = g_c(\hat{x}_c, g_p(\hat{x}_p)). \tag{10.45}$$

As described in Sect. 10.3.2.2, the controller node runs (10.44)-(10.45) in order to compute a prediction over the fixed horizon T_0^p. Since $\hat{y}$ remains constant during the prediction, an update of the model using $y_p\left(\tau\right)$, which is sent over the network, is indistinguishable from using $x_p\left(\tau\right)$ along with the perfect knowledge of g_p. Then, from a mathematical point of view, we can consider that $\hat{g}_p\left(x_p\right) \equiv g_p\left(x_p\right)$ and that, in addition, x_p is sensed and sent over the network. When implementing the control scheme—on the other hand—sending y_p only will be preferred.

As said $\dot{\hat{y}} = 0$; nonetheless, it is clear from (10.44)-(10.45) that $\dot{u} \neq 0$. When computing the control law the evolution of the closed loop system is taken into account as *prescribed* by PBC.

Under the current assumptions, the following additional definition can be given:

Definition 5 Given $T \in \mathbb{R}_{\geq 0}$, we will say that $x_p(\tau)$ is known at time $\tau + T$ if $y_{[0,\tau]}$ is known to the controller at time $\tau + T$.

This definition—which scope is limited to this section—formalises the fact that if $y_{[0,\tau]}$ is known to the controller at time $\tau + T$, the dynamic control law over the fixed time prediction can be computed as if $x_p\left(\tau\right)$ had been measured and sent to the controller.

Because of definition 5, proposition 1 holds when $x_c(\cdot)$ is changed into $x(\cdot)$ and the protocol becomes then:

$$e(\tau_{o_i}^{m+}) = h\left(i, e(\tau_{o_i}^{m})\right) = 0,\ \forall i \in \mathbb{N},\ \text{with } \{o_i\} = \mathbb{N}, \tag{10.46}$$

which is clearly UGES.

As an illustrative example, we consider the linear system

$$\dot{x} = \begin{pmatrix} 0.8145 & 0.5056 & 0.444 \\ 0.87891 & 0.6357 & 0.06 \\ 0.8523 & 0.95 & 0.8667 \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u \tag{10.47}$$

$$y = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} x \tag{10.48}$$

which is an unstable system. The controller

$$\dot{x}_c = \begin{pmatrix} 0.81450 & 0.50560 & -1585.5090140 \\ 0.78910 & 0.63570 & 1123.345947373776 \\ -41.14770 & -51.04910 & 963.5497999990 \end{pmatrix} x_c + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} y \tag{10.49}$$

$$u = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} x_c \tag{10.50}$$

(exponentially) stabilises the system.

As for the hypotheses of this section, the model of the system is $\dot{\hat{x}}_p = \hat{f}_p(\hat{x}_p, u) = 0$ and the designed controller is an output-feedback dynamic controller. Figure 10.8

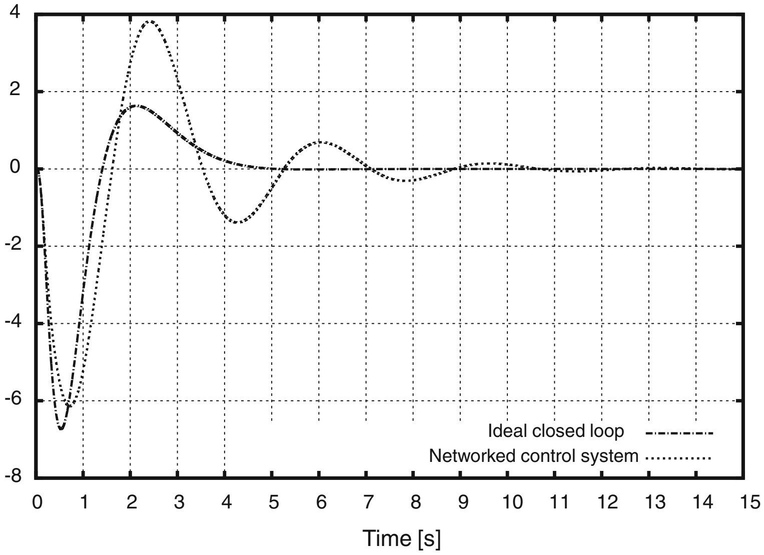


Fig. 10.8 System output: $y(t)$—with different control architectures

shows the behavior of the output of the ideal closed-loop system along with the behavior of the network controlled system, for which no state measurement is performed.

10.7 Proofs

We start by giving a proof for Proposition 1.

Proof (Virtual packets) Pick $i \in \mathbb{N}, i \geq \ell_y - 1$ and assume that $x_c(\tau^m_{o_{i-(\ell_y-1)}})$ is not known at time $T = \tau^m_{o_{i-(\ell_y-1)}} + \ell_y \tau^m + T^m$. Then there exists $l \in [1, \ell_y]$: $y_{\left[0,\tau^m_{o_{i-(\ell_y-1)}}\right]l}$ is not known at time T. It follows that $\nu_{i-(\ell_y-1)} \neq l$ and that there exists $j < \ell_y$: $\nu_{i-(\ell_y-1)+j} = l$. Hence $y_{\left[0,\tau^m_{o_{i-(\ell_y-1)+j}}\right]l}$ is known at time $\tau^m_{o_{i-(\ell_y-1)+j}} + T^m_{o_{i-(\ell_y-1)+j}}$. But since $\tau^m_{o_{i-(\ell_y-1)+j}} + T^m_{o_{i-(\ell_y-1)+j}} \leq \tau^m_{o_{i-(\ell_y-1)+j}} + T^m \leq \tau^m_{o_{i-(\ell_y-1)}} + j\tau^m + T^m \leq \tau^m_{o_{i-(\ell_y-1)}} + \ell_y \tau^m + T^m = T$ we can conclude that $y_{\left[0,\tau^m_{o_{i-(\ell_y-1)+j}}\right]l}$ is known at time T; which is an absurd.

In order to prove Proposition 2 we need to state some preliminary results.

Lemma 1 (Sum of Protocols) *Given two UGES protocols* $h_s : \mathbb{N} \times \mathbb{R}^{n_s} \to \mathbb{R}^{n_s}$ *and* $h_p : \mathbb{N} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_p}$ *with relative constants and functions as contained in Definition 1, distinguished by the subscript s and p, let define* $e \triangleq \left[s^T, p^T\right]^T$, *with* $s \in \mathbb{R}^{n_s}$, $p \in \mathbb{R}^{n_p}$. *The protocol:*

$$h(i, e) \triangleq \begin{bmatrix} h_s(i, s) \\ h_p(i, p) \end{bmatrix} \tag{10.51}$$

is UGES with parameters $\underline{a} = \min\left\{\underline{a}_p, \underline{a}_s\right\}$, $\overline{a} = \max\left\{\overline{a}_s, \overline{a}_p\right\}$, $\rho = \max\left\{\rho_s, \rho_p\right\}$ *and* $c = \max\left\{c_s, c_p\right\}$.

Proof With the parameters defined in the lemma and the Definition 1 in mind, we define the function $W : \mathbb{N} \times \mathbb{R}^{(n_s+n_p)} \to \mathbb{R}_{\geq 0}$ as

$$W(i, e) = \sqrt{W_s(i, s)^2 + W_p(i, p)^2}. \tag{10.52}$$

Hence

$$\begin{aligned} &\underline{a}^2 \|e\|^2 = \underline{a}^2 \left(\|s\|^2 + \|p\|^2\right) \leq W(i, e)^2 \leq \overline{a}^2 \left(\|s\|^2 + \|p\|^2\right) \\ &= \overline{a}^2 \|e\|^2 \, W\left(i + 1, h(i, e)\right)^2 \leq \rho^2 \left(W_s(i, s)^2 + W_p(i, p)^2\right). \end{aligned}$$

Furthermore

$$\begin{aligned}\left\|\frac{\partial W}{\partial e}\right\| &= \frac{1}{W}\left\|\left[W_s\frac{\partial W_s}{\partial s},\ W_p\frac{\partial W_p}{\partial p}\right]\right\| \\ &= \frac{1}{W}\sqrt{\left\|W_s\frac{\partial W_s}{\partial s}\right\|^2 + \left\|W_p\frac{\partial W_p}{\partial p}\right\|^2} \\ &\leq \frac{1}{W}\sqrt{W_s^2c_s^2 + W_p^2c_p^2} \leq \frac{c}{W}\sqrt{W_s^2 + W_p^2} = c,\end{aligned}$$

by means of which the lemma is proven.

Lemma 2 (ℕ-dilation of a Protocol) *Let us consider a UGES protocol* $h : \mathbb{N}\times\mathbb{R}^n \to \mathbb{R}^n$ *and a sequence* $\{\omega_k\}_{k\in\mathbb{N}}$ *having values in* ℕ *such that* $0 < \omega_{k+1} - \omega_k \leq \bar{\imath}$*, the protocol defined as:*

$$h_D\,(i, e) = \begin{cases} h\,(i, e) & \textit{if } \exists k : \omega_k = i \\ e & \textit{otherwise} \end{cases} \tag{10.53}$$

is UGES. In particular, if h is UGES with parameters $\underline{a}$*,* $\overline{a}$*,* ρ*,* c*,* h_D *is UGES with parameters* $\rho\underline{\alpha}$*,* $\overline{\alpha}$*,* $\rho^{\frac{1}{\bar{\imath}}}$ *and* c*.*

Proof Given the function W associated with the UGES protocol h, we define a function W_D

$$\begin{aligned} &W_D(0, e) = W(0, e) \\ &W_D(i + 1, e) = \begin{cases} W(k + 1, e) & \text{if } \exists k : \omega_k = i \\ \rho^{\frac{1}{\bar{\imath}}}W_D(i, e) & \text{otherwise.} \end{cases}\end{aligned} \tag{10.54}$$

The following conditions hold:

$$\rho\underline{a}\|e\| \leq W_D(i, e) \leq \overline{a}\|e\|.$$

Furthermore, when $\nexists k : \omega_k = i$

$$W_D\,(i + 1, h_D\,(i, e)) = W_D(i + 1, e) \leq \rho^{\frac{1}{\bar{\imath}}}W_D(i, e).$$

Consider now the case $\exists k : \omega_k = i$. We have:

$$W_D\,(\omega_{k-1} + 1, e) = W\,(k, e) \tag{10.55}$$

and

$$W_D\,(\omega_k, e) = \rho^{\frac{\omega_k - \omega_{k-1} - 1}{\bar{\imath}}}W\,(k, e) \geq \rho^{\frac{\bar{\imath}-1}{\bar{\imath}}}W\,(k, e)\,.$$

Therefore:

$$\rho W\left(k, e\right) \leq \rho^{\frac{1}{\bar{\imath}}} W_D\left(\omega_k, e\right). \tag{10.56}$$

We can then write

$$\begin{aligned} W_D\left(\omega_k + 1, h_D\left(\omega_k, e\right)\right) &= W\left(k+1, h\left(k, e\right)\right) \\ &\leq \rho W\left(k, e\right) \leq \rho^{\frac{1}{\bar{\imath}}} W_D\left(\omega_k, e\right). \end{aligned}$$

Since $\rho < 1$, it is apparent from the very definition of W_D that

$$\left\| \frac{\partial W(i, e)}{\partial e} \right\| \leq c \Rightarrow \left\| \frac{\partial W_D(i, e)}{\partial e} \right\| \leq c,$$

which concludes the proof.

We can now give a proof for Proposition 2.

Proof Define the $\mathbb{N}$-dilated protocol from h_p:

$$h_{Dp}(i, e_p) = \begin{cases} h_p(k, e_p) \text{ if } \exists k : s_k = i \\ e_p \qquad\qquad \text{otherwise} \end{cases} \tag{10.57}$$

From the definition of s_i, it follows that $\bar{\imath} = \ell_y + 1$. Hence $h_{Dp}(i, e_p)$ is UGES with parameters $\underline{a}_{Dp} = \underline{a}_p \rho_p$, $\overline{a}_{Dp} = \overline{a}_p$, $\rho_{Dp} = \rho_p^{\frac{1}{\ell_y+1}}$ and $c_{Dp} = c_p$.

Define now $h_c(i, e_c) = 0$, which is an UGES protocol having parameters $\underline{a}_c = \overline{a}_c = c_c = \underline{a}_p$ and $\rho_c = \rho_p$.[2] Define the $\mathbb{N}$-dilated protocol

$$h_{Dc}(i, e_c) = \begin{cases} 0 \quad \text{if } \exists k : o_k = i \\ e_c \qquad \text{otherwise.} \end{cases} \tag{10.58}$$

From the definition of o_i, it follows that $\bar{\imath} = 2$. Hence $h_{Dc}(i, e_c)$ is UGES with parameters $\underline{a}_{Dc} = \underline{a}_p \rho_p$, $\overline{a}_{Dc} = \underline{a}_p$, $\rho_{Dc} = \rho_p^{\frac{1}{2}}$ and $c_{Dc} = \underline{a}_p$.

Now, the protocol in Proposition 2 can be obtained by applying Lemma 1 to the protocols h_{Dp} and h_{Dc}, hence it is UGES with parameters $\underline{a} = \underline{a}_p \rho_p$, $\overline{a} = \overline{a}_p$, $\rho = \rho_p^{\frac{1}{\ell_y+1}}$ and $c = c_p$.

In order to prove Theorem 1, we are now going to invoke theorem 1 from [4]. The assumption on the network made there are satisfied by means of (10.21) which follows from our Assumption 4 in conjunction with the definition of the sequence of state-sending times and delays. The assumption regarding the protocol being UGES is here satisfied by means of Proposition 2. The ideal closed-loop system has the

[2] The function $W_c(i, e_c) = \underline{a}_c \left\| e_c \right\|$ can be used to show this.

nominal GES property, as stated in Assumption 1. Moreover, Assumptions 2 and 3 are the same required in [4].

Theorem 1 is a rewriting of Theorem 1 in [4] in terms of the quantities involved in the writing of system (10.25).

10.8 Conclusions

The networked stabilisation of a nonlinear plant via an output-feedback dynamic controller has been considered. An algorithm is proposed which exploits the packet-based nature of the considered network. Sufficient condition for the local exponential stability of the resulting system are given. The stabilisation of a magnetic levitator and of a Furuta pendulum are presented as an example. Network-in-the loop experiment results show that the resulting network controlled system closely mimics the behavior of the ideal closed-loop system. If–on the contrary–the proposed algorithm is not used, the network is shown to strongly affect the behavior of the controlled system.

Acknowledgments The research leading to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement n257462 HYCON2 Network of excellence.

References

1. Heemels, W.P.M.H., Teel, A.R., van de Wouw, N., Nešić, D.: Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance. IEEE Trans. on Automat. Contr. **55**(8), 1781–1796 (2010)
2. Bemporad, A.: Predictive control of teleoperated constrained systems with unbounded communication delays. In: Proceedings of IEEE International Conference on Decision and Control, vol. 2, pp. 2133–2138. Tampa (1998) .
3. Polushin, I.G., Liu, P.X., Lung, C.H.: On the model-based approach to nonlinear networked control systems. Automatica **44**(9), 2409–2414 (2008)
4. Greco, L., Chaillet, A., Bicchi, A.: Exploiting packet size in uncertain nonlinear networked control systems. Automatica **48**, 2801–2811 (2012)
5. Nesic, D., Liberzon, D.: A unified framework for design and analysis of networked and quantized control systems. IEEE Trans. Autom. Control **54**(4), 732–747 (2009).
6. Walsh, G. C., Ye, H., Bushnell, L.: Stability analysis of networked control systems. In: Proceedings of American Control Conference (1999).
7. Nesic, D., Teel, A.: Input-output stability properties of networked control systems. IEEE Trans. Autom. Control **49**(10), 1650–1667 (2004).
8. Falasca, S., Belsito, C., Quagli, A., Bicchi, A.: A Modular and Layered Cosimulator for Networked Control Systems. In: Proceedings of IEEE Mediterranean Conference on Control (2010).
9. Fabbri, T., Fenucci, D., Falasca, S., Gamba, M., Bicchi, A.: Packet-based dynamic control of a furuta pendulum over ethernet. In: Proceedings of IEEE Mediterranean Conference on Control (2013).
10. Astrom, K. J., Furuta, K.: Swinging up a pendulum by energy control. In: IFAC 13th World Congress, San Francisco, California, 1996. (1996).

Chapter 11
Real-Time Bug-Like Dynamic Path Planning for an Articulated Vehicle

Thaker Nayl, George Nikolakopoulos and Thomas Gustafsson

Abstract This article proposes a novel real time bug like algorithm for performing a dynamic smooth path planning scheme for an articulated vehicle under limited and sensory reconstructed surrounding static environment. In the general case, collision avoidance techniques can be performed by altering the articulated steering angle to drive the front and rear parts of the articulated vehicle away from the obstacles. In the presented approach factors such as the real dynamics of the articulated vehicle, the initial and the goal configuration (displacement and orientation), minimum and total travel distance between the current and the goal points, and the geometry of the operational space are taken under consideration to calculate the update on the future way points for the articulated vehicle. In the sequel the produced path planning is iteratively smoothed online by the utilization of Bezier lines before producing the necessary rate of change for the vehicle's articulated angle. The efficiency of the proposed scheme is being evaluated by multiple simulation studies that simulate the movement of the articulated vehicle in open and constrained spaces with the existence of multiple obstacles.

Keywords Articulated vehicle · Path planning · Obstacle avoidance

11.1 Introduction

Recently, there have been significant advances in designing automated articulated vehicles mainly for their utilization in the mining industry, where the aim has been the overall increase of the production, while making the working conditions for the human operators safer [1]. In most of the cases, these vehicles are remotely operated, while there is a continuous trend for increasing the autonomy levels, especially in the area of path planning and obstacle avoidance as the vehicles need: (a) to perceive the

T. Nayl (✉) · G. Nikolakopoulos · T. Gustafsson
Automatic Control Group, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 971 87 Luleå, Sweden
e-mail: thanay@ltu.se
http://www.ltu.com

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_11

changing environment, based on the onboard sensory systems and (b) autonomously plan their route towards the final objective [2].

For the classical task of path planning, with an obstacle detection and avoidance capability, the simplest technique to solve the problem is the altering of the vehicle's orientation, while predicting a non collision path, based on the vehicle's kinematic model, the sensing range and the safety range. In this approach a finite optimal sequence of control inputs, according to the initial vehicle position and the desired goal point is being generated, which is able to take under consideration positioning and measuring uncertainties, such that the collision with any obstacle at a given future time never occurs.

From another point of view, path planning can be divided in two main categories according to the assumptions of: (a) global approaches where it is being assumed that the map is a priori available, and (b) a partially known and reconstructed surrounding environment based on reactive approaches, which utilizes sensors like infrared, ultrasonic and local cameras. Characteristic examples of the first case are the Road–Map algorithm [3], the Cell Decomposition [4], the Voronoi diagrams [5], the Occupancy Grinds [6] and the new Potential Fields techniques [7], while in most of the cases, a final step of smoothing the produced path curvatures, by the utilization of Bezier curves is being utilized [8].

For the second case of a partially known and online reconstructed environment, the Bug family algorithms are well known mobile vehicle navigation methods for local path planning based on a minimum set of sensors and with a decreased complexity for online implementation [9]. One of the most commonly utilized path planning algorithm in this category is the Bug1 and Bug2 [10]. Bug1 algorithm exhibits two behaviors; motion to goal with boundary following and a corresponding hit point and leave point, while Bug2 algorithm presents similar behaviors like the Bug1 algorithm, except from the fact that it tries to follow the fixed line from a start point to the goal, during obstacle avoidance. Other Bug algorithms that also incorporate range sensors are TangentBug [11], DistBug [12] and VisBug [13]. Tangent Bug algorithm is an improvement of the Bug2 algorithm since it is able to determine the shorter path to the goal using a range sensor with a 360° infinite orientation resolution. DistBug has a guaranteed convergence and will find a path if one exists, while it requires the perception of its own position, the goal position and the range sensory data [14]. The VisBug algorithm, needs global information to update the value of the minimum distance to the goal point, during the boundary following and for determining the completion of a loop during the convergence to the goal. In all the presented path planning algorithms, the vehicle is being modeled as a point within the world space, without any constraint in the movements, while the actual kinematics of the vehicle, which is important especially in the case of non–holonomic vehicles are being neglected.

The novelty of this article stems from the proposal of a new bug like path planning algorithm based on the model of an articulated vehicle, which is able to consider: (a) the physical constraints of the vehicle, (b) proper obstacle detection and avoidance, and (c) smooth path generation based on an online Bezier lines processing of the produced way points. In the presented approach the solution to the path planning problem is generated online, based on partial and online sensory information of the vehicle's surrounding environment, while the path is being calculated by solving the inverse kinematic problem of the articulated vehicle or by calculating the optimal articulation angle. Moreover, as in the case of all the exploration and final goal seeking algorithms, it is assumed that the vehicle is constantly aware of the final goal coordinates. During the convergence to this goal and based on the limited range sensing of the surrounding environment, the vehicle is able to detect and avoid obstacles, while continuously converging to the optimum goal. This approach provides an online and sub optimal solution, when compared with the global path planning techniques, and it can be directly applied to the case of articulated vehicles. As it has been applied in the previous path planning algorithms for the case of a priori known space configuration, in the proposed scheme, the Bezier curves are being also utilized for filtering the produced way points and thus guarantee for an online smooth path planning due to the Bezier's line property of continuous higher-order derivatives.

The rest of the article is organized as it follows. In Sect. 11.2 the model of the articulated vehicle and the corresponding state space equations will be presented. In Sect. 11.3 the proposed novel scheme for smooth path planning and obstacle avoidance based on the articulated vehicle's dynamics will be introduced, while in Sect. 11.4 multiple simulation results will be depicted that prove the efficacy of the path planning scheme in different test cases. Finally, the concluding remarks are provided in Sect. 11.5.

11.2 Articulated Vehicle Model

An articulated vehicle is constructed by two parts, a front and a rear, linked with a rigid free joint, while each body has a single axle and the wheels are all non–steerable, with the steering action to be performed on the joint, by changing the corresponding articulated angle γ between the front and the rear of the vehicle [15] as it being also presented in Fig. 11.1.

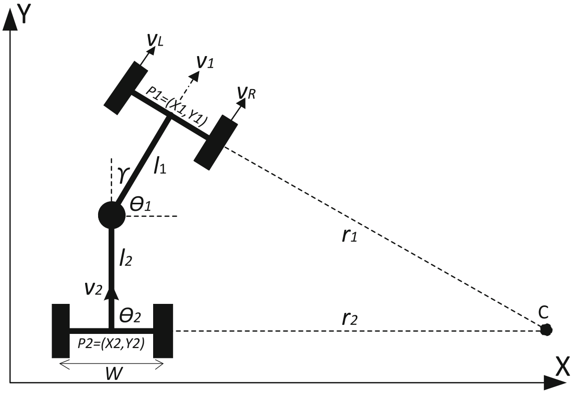


Fig. 11.1 Articulated vehicle's geometry

The main assumptions to derive the kinematic model of the articulated vehicle are: (a) the steering angle γ remains constant under small displacement, (b) dynamical effects due to low speed, like tire characteristic, friction, load and breaking force are being neglected, (c) it's assumed that the vehicle moves on a plane without slipping effects, during low-level control, the vehicle's velocities are bounded within the maximum allowed velocities, which prevents the vehicle from slipping, and (c) each axle is composed of two wheels and when replaced by a unique wheel, can get:

$$\dot{X}_1 = V_1 \cos \theta_1 \tag{11.1}$$

$$\dot{Y}_1 = V_1 \sin \theta_1 \tag{11.2}$$

The steering angle γ is being defined as the difference between the orientation angles of the front θ_1 and the rear parts θ_2 of the vehicle.

The velocity V_1 at the front and V_2 at the rear parts have the same changing with respect to the velocity at the rigid free joint of the vehicle, and it can be defined by the relative velocity vector equations as it follows:

$$V_1 = V_2 \cos \gamma + \dot{\theta}_2 l_2 \sin \gamma \tag{11.3}$$

$$V_2 \sin \gamma = \dot{\theta}_1 l_1 + \dot{\theta}_2 l_2 \cos \gamma \tag{11.4}$$

where $\dot{\theta}_1$, $\dot{\theta}_2$ and l_1, l_2 are the angular velocities and the lengths of the front and rear parts of the vehicle respectively. By combining these equations it yields:

$$\dot{\theta}_1 = \frac{V_1 \sin \gamma + l_2 \dot{\gamma}}{l_1 \cos \gamma + l_2} \tag{11.5}$$

while the angles γ and θ_1 can be measured with a great accuracy. For the case that there is a steering limitation for driving the rear part, according to the coordinates of the point $P_2 = (X_2, Y_2)$, the geometrical relationship between P_1 and P_2 is provided by:

$$X_2 = X_1 - l_1 \cos \theta_1 - l_2 \cos \theta_2 \tag{11.6}$$

$$Y_2 = Y_1 - l_1 \sin \theta_1 - l_2 \sin \theta_2 \tag{11.7}$$

The realistic dynamic motion behavior of the articulated vehicle with initial parameters $[X_r \ Y_r \ \theta_r \ \gamma_r]$, is depicted in Fig. 11.2, where the vehicle is requested to reach the goal destination with a specific orientation. As it can be observed, when the dynamics of the vehicle are being incorporated the motion and the overall behavior of the vehicle significantly deviates from the case where the vehicle is being considered of having the dynamics of an unconstraint point and this is one of the major contributions of this article. Moreover another test scenario will be considered, in this case following the 8-pattern in open loop. In Fig. 11.3 the tracking of the desired path can be observed with respect to the control signals $\dot{\gamma} = 3.5°/\text{s}$ depicted in Fig. 11.4.

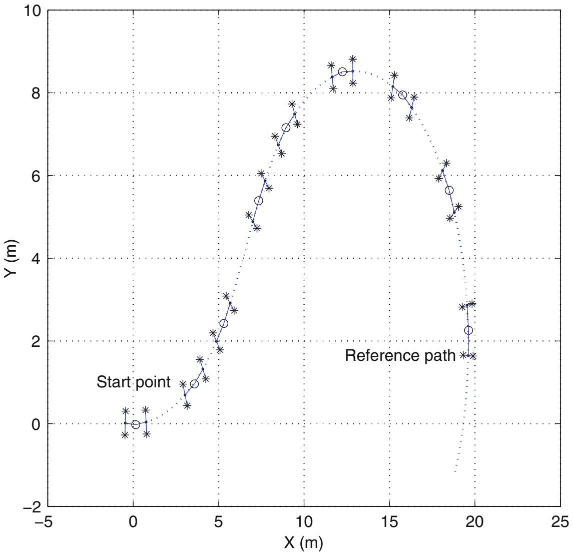


Fig. 11.2 Realistic dynamic motion behavior of the articulated vehicle starting at $[0\ 0\ 0\ \ -10°]$ *with* V=1 m/s and $\dot{\gamma} = 0°$/s for the first 5 s of movement, while $\dot{\gamma} = 3.5°$/s for the next 6 s to reach the goal point at [20 2]. The vehicle dimensions are $l_1 = l_2 = 0.6$ m and $W = 0.58$ m

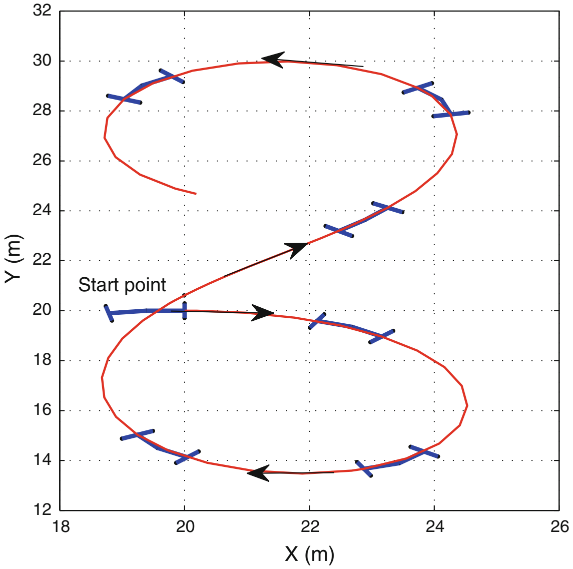
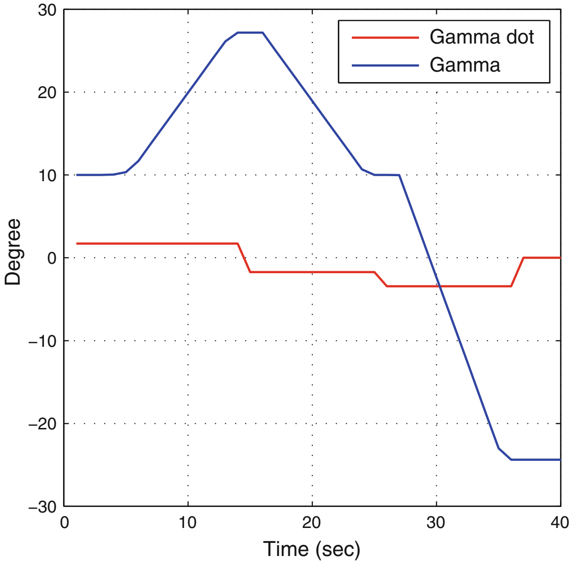


Fig. 11.3 Motion behavior of the articulated vehicle (8-pattern) simulating the non-linear model by applying the control signals $\dot{\gamma}$ and γ

The state parameters of the articulated vehicle are; $\mathbf{X} = [X\ Y\ \theta\ \gamma]^T$ and the manipulated variables are $\mathbf{u} = [V\ \dot{\gamma}]^T$, while the kinematic model of the articulated vehicle, in a state space formulation can be written as it follows:

Fig. 11.4 Control signals $\dot{\gamma}$ and γ applied to the steering angle in order to get the 8-pattern



$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ \frac{\sin\gamma}{l_1\cos\gamma + l_2} & \frac{l_2}{l_1\cos\gamma + l_2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ \dot{\theta}_{\gamma}, \end{bmatrix} \tag{11.8}$$

where $\dot{\theta}_{\gamma}$ is the rate of change for the articulated angle.

11.3 On Line Smooth Path Planning for an Articulated Vehicle

The introduced path planning algorithm can be applied for the objective of moving a vehicle from a starting point to the goal point, while detecting and avoiding identified obstacles based on the real vehicles dynamic equations of motion. As a common property of the Bug like algorithms, the proposed scheme initially faces the vehicle towards the goal point, which it is being assumed to be a priori known. In the proposed path planning module it is also being assumed that the vehicle is able to online sense the surrounding environment based on the available sensory systems. The proposed scheme is able to replan the produced path, by generating new way points, after the identification of an obstacle and produce proper path deformations that need to be done for avoiding it. In all these cases the produced set of new way points are utilized as control points for a Bezier curve algorithm for online smoothing of the suggested path. The overall proposed concept of the novel path planning algorithm is being presented in the following Fig. 11.5.

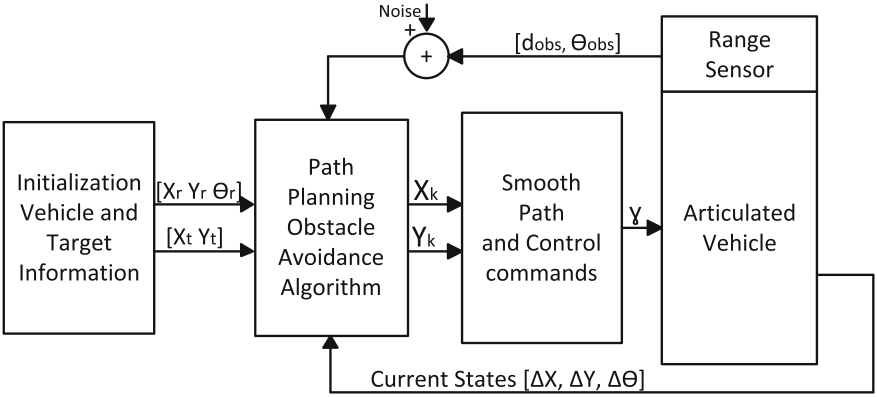

Noise
+
+
[dobs, Θobs]
Range Sensor
Initialization Vehicle and Target Information
[Xr Yr Θr]
[Xt Yt]
Path Planning Obstacle Avoidance Algorithm
Xk
Yk
Smooth Path and Control commands
γ
Articulated Vehicle
Current States [ΔX, ΔY, ΔΘ]

Fig. 11.5 Block diagram of the proposed path planning algorithm based on the nonlinear articulated kinematic model

As it can be observed from this diagram the algorithm starts by defining the current position and orientation of the vehicle, denoted by $[X_r,\ Y_r,\ \theta_r]$ and the final goal position denoted by $[X_g,\ Y_g,\ \theta_g]$. Based on the onboard sensory system, the vehicle identifies the surrounding environment and obstacles and generates the way points for reaching the goal destination. In the sequel the way points are been smoothed by the utilization of Bezier filtering, while as a last step and based on the vehicles dynamics, an open loop control signal (articulated angle) is being generated to guide the vehicle. In the presented approach it is also assumed that the system is fully observable and good and timely available measurements can be provided for the displacement and orientation of the vehicle.

The assumed sensory system is able to detect the obstacles and the surrounding environment, measure the distance of the articulated robot from the obstacle $d_{obs} \in \Re$, while a sensing radius $\theta_{obs} \in \Re$ is being considered, reflecting real life sensing limitations. In the presented approach all the obstacles and the surrounding environment are being considered as point clouds in a 2-dimensional space, while the overlapping obstacles are being clustered and represented by a single and unified obstacle. The notations utilized and the overall concept of the proposed path planning algorithm are depicted in Fig. 11.6. The overall flowchart diagram for path planning and obstacle avoidance for the case of an articulated vehicle is being depicted in Fig. 11.7, while is can be summarized as it follows:

[Step 1: Initialization]. Define initial $[X_r\ Y_r\ \theta_r\ \dot{\gamma}_r]$ and goal $[X_g\ Y_g]$, define the articulated vehicle's specific parameters $V,\ d_{\min},\ \theta_{\min},\ d_{obs},\ \theta_{obs}$, the path update rate defined as T and the vehicle's mechanical and physical constraints that needs to be taken under consideration. Set $[X_k\ Y_k\ \theta_k\ ] = [X_r\ Y_r\ \theta_r]$, with $k \in Z^+$ the sample index.

[Step 2: Path Update]. Utilize Eqs. (11.1–11.8) to update the coordinates of the next way point as:

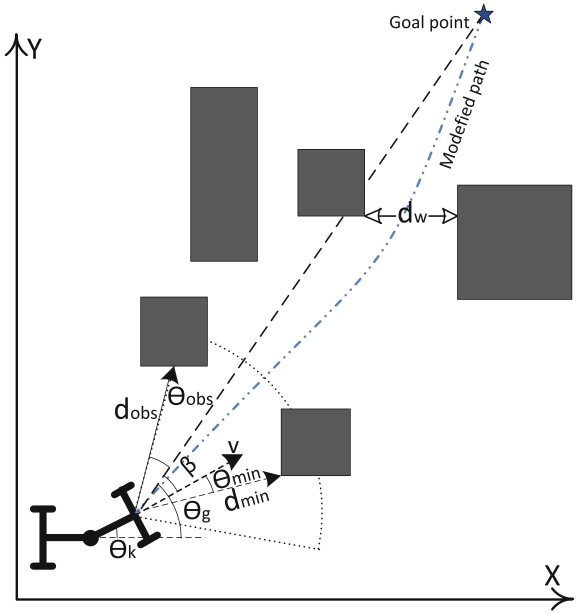


Fig. 11.6 Notations and overall concept of the proposed path planning algorithm

$$[X_{k+1}\ Y_{k+1}\ \theta_{k+1}] = [X_k\ Y_k\ \theta_k] + T \cdot [\Delta X\ \Delta Y\ \Delta\theta]$$

calculate:

$$\theta_g = \arctan \frac{Y_{k+1} - Y_g}{X_{k+1} - X_g}$$
$$\beta = \theta_g - \theta_k$$

to produce γ with θ_g the angle between the line that connects the center of gravity of the vehicle's front part to the goal point and the X axis, while β is the difference angle among the vehicle's orientation angle, with the X axis and θ_g. During the application of this step, constraints can be imposed on the articulated vehicle just by bounding the allowable articulation within $\gamma^+ \le \dot{\gamma} \le \gamma^-$, with $^+$ and $^-$ representing the maximum and the minimum bounds on the rate of articulated angle.

[Step 3: Obstacle Avoidance] The obstacle avoidance strategy becomes active when the safety conditions $d_{\min}$ and $\theta_{\min}$ are satisfied. This can be evaluated by calculating the update of the distance from the obstacle and the obstacle's angle by:

$$Dis_{obs} = \sqrt{(X_{k+1} - X_{obs})^2 + (Y_{k+1} - Y_{obs})^2}$$
$$\theta_{obs} = \arctan \frac{Y_{k+1} - Y_{obs}}{X_{k+1} - X_{obs}}$$
$$\theta_{obs,g} = \theta_{obs} - \theta_g$$

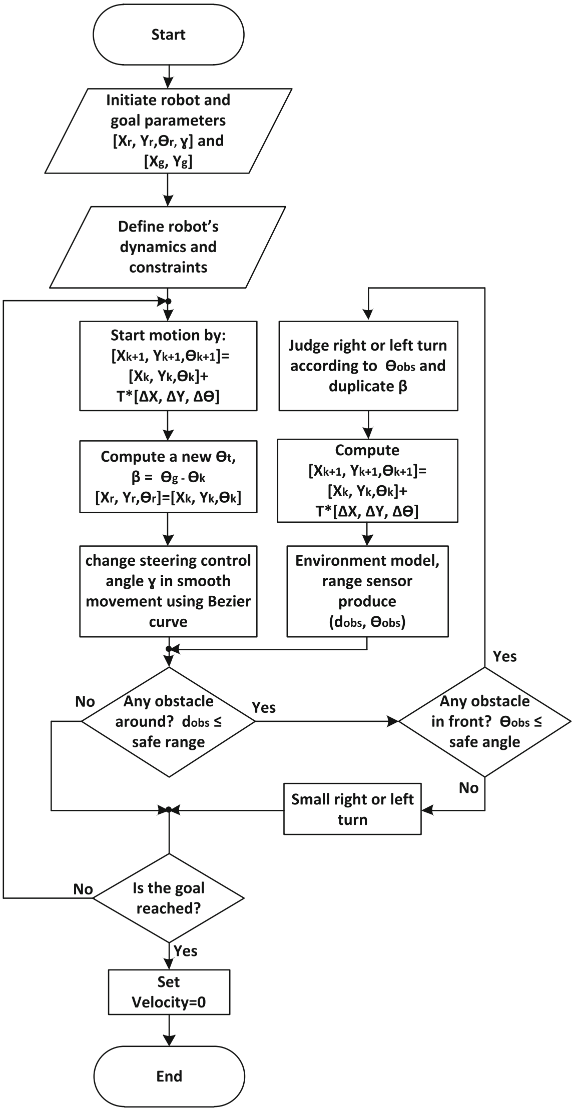
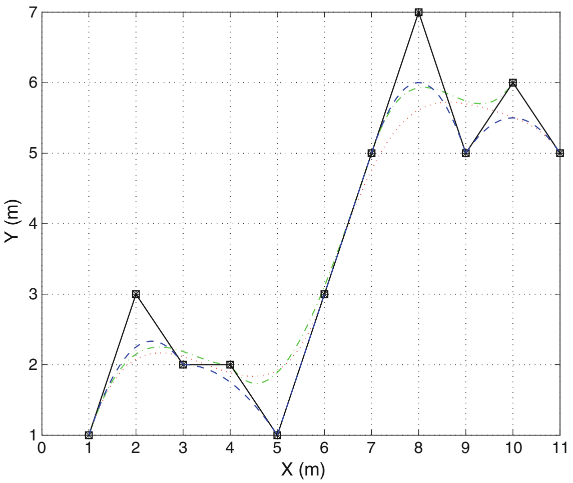

Start
Initiate robot and goal parameters $[X_r, Y_r, \Theta_r, \gamma]$ and $[X_g, Y_g]$
Define robot's dynamics and constraints
Start motion by: $[X_{k+1}, Y_{k+1}, \Theta_{k+1}] = [X_k, Y_k, \Theta_k] + T*[\Delta X, \Delta Y, \Delta\Theta]$
Compute a new Θ_t, $\beta = \Theta_g - \Theta_k$ $[X_r, Y_r, \Theta_r] = [X_k, Y_k, \Theta_k]$
change steering control angle ɣ in smooth movement using Bezier curve
Judge right or left turn according to Θ_{obs} and duplicate β
Compute $[X_{k+1}, Y_{k+1}, \Theta_{k+1}] = [X_k, Y_k, \Theta_k] + T*[\Delta X, \Delta Y, \Delta\Theta]$
Environment model, range sensor produce (d_{obs}, Θ_{obs})
Yes
No
Any obstacle around? $d_{obs} \leq$ safe range
Yes
Any obstacle in front? $\Theta_{obs} \leq$ safe angle
No
Small right or left turn
No
Is the goal reached?
Yes
Set Velocity=0
End

Fig. 11.7 Main flowchart of path planning motion

while in the case that the following conditions are true:

$$
\begin{gathered}
(Dis_{obs} < d_{\min})\ AND\ (\theta_{obs,g} < \theta_{\min}) \\
OR \\
(Dis_{obs} < d_{\min})\ AND\ (\theta_{obs,g} < -\theta_{\min})
\end{gathered}
$$

Fig. 11.8 Bezier *curves* based on different number of control points resulting in different paths, (*Black line-solid* path produced from the generated way points, *red line-dots* for 6 points, *green line-dash dot* for 4 points and the *blue line-dash* for 3 points)



the changing in the steering angle is being duplicated and Step 3 is repeated again till the condition in (11.9) is false and the algorithm continues from Step 2 or the bounds on the articulated angle cannot meet and the algorithm jumps to Step 4. **[Step 4: Reaching Final Goal]** If $[X_{k+1}\ Y_{k+1}] = [X_g\ Y_g] \pm Dis_{tolr}$, set $velocity = 0$ and the path planning algorithm has been terminated. Otherwise, the algorithm jumps to Step 2 and the whole process is being repeated in order to avoid collisions with the obstacles until vehicle reaches the goal tolerance distance.

During the execution of the proposed path planning algorithm, and especially Step 2, the proposed path planning algorithm always smoothes the produced way points by the utilization of Bezier curve filtering. The mathematical formulation of the applied Bezier smothering is denoted as:

$$\mathbf{B}(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i}\, t^i\, P_i\ , \quad t \in [0, 1] \tag{11.9}$$

The number n of the considered control points for the Bezier curve generation plays a significant role in the final shape of the produced smooth path as it can be observed from Fig. 11.8, where multiple Bezier lines are being displayed with respect to different number of control points. A n degree Bezier line always passes through the first and last control points and it can be proved that it always lies within the convex hull of the control points, while being tangent to the lines connecting the way points [16].

11.4 Simulation Results

For simulating the efficacy of the proposed path planner, the following articulated vehicle's characteristics have been considered: $l_1 = l_2 = 0.6$ m, $W = 0.58$ m, while the vehicle's speed is constant and equal to 1 m/s. Moreover, the constraints imposed on the articulated angle γ have been defined as ± 0.523 rad and random measurement Gaussian noise with a fixed variance was added to all measurements of the range sensor to simulate the real life measurement distortion.

The effectiveness of the proposed algorithm will be evaluated in arenas of different types and dimensions. More analytically, the algorithm is simulated on six types of environments with different obstacle configurations, where the vehicle and obstacle geometry is described in a 2D workspace. The obtained outputs of the path planning solutions from the indicated starting points to the goal points is being depicted in Figs. 11.9, 11.10, 11.11, 11.12, 11.13 and 11.14, displaying cases with the same sensing radius $d_{obs} = 3$ m and various $d_{\min}$.

As it can be observed in all the examined cases the vehicle is able to avoid all the obstacles, including the bounding surrounding (e.g. walls), which can also be considered as obstacles without loss of generality. In the presented simulations, the articulated vehicle reaches the reference final goal, independently of the initial vehicle's orientation, while in all the simulations the safety radius has been also displayed with the red circle notation. The basic assumption in all these simulations is that the articulated vehicle, in every time instant, is aware of the coordinates of the final goal and thus the path is being tuned in every step based on the identified obstacles, while the vehicle explores the surrounding environment towards the final goal. In an obstacle free environment, the optimal solution to this problem would have been a straight line connecting the initial with the goal point, a case that can be easily identified in the presented simulation in Fig. 11.9. The online identification of

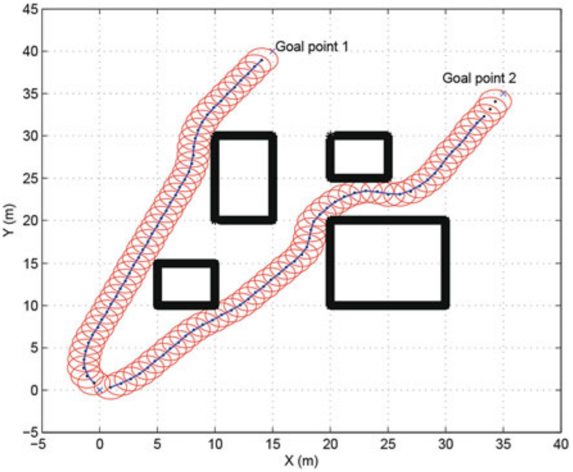


Fig. 11.9 Different shape obstacles placed in the workspace. During these simulations the vehicle starts from different initial angles at $[0,\ 0,\ 120°,\ 7.5°]$ and $[0,\ 0,\ 20°,\ 7.5°]$ to reach the goal points located at $[15,\ 40]$ and $[35,\ 35]$ respectively, with safe distance $d_{\min} = 3$ m, $d_{obs} = 3$ m

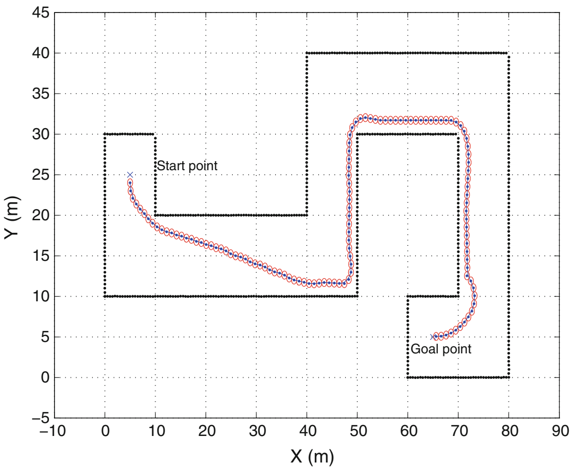


Fig. 11.10 Path planning in an arena having boundaries on both sides of the road a fact that restricts the articulated vehicle motions. During this simulation scenario the vehicle is starting from the initial posture $[5,\ 25,\ -90^\circ,\ 7.5^\circ]$ and the goal is located at $[65,\ 5]$ and $d_{\min} = 0.5$ m, $d_{obs} = 3$ m

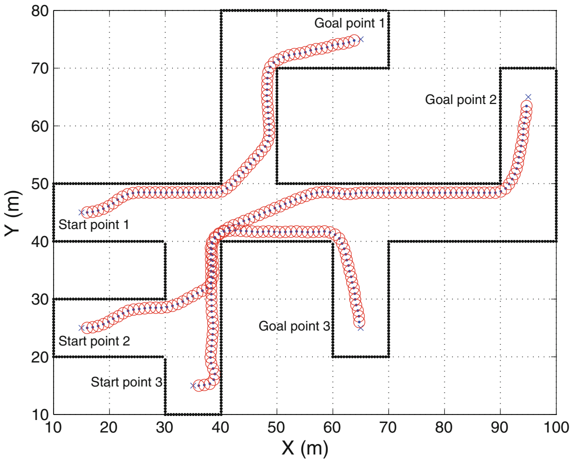


Fig. 11.11 Path planning in an arena having more complicated boundaries on both sides of the road, with noise measurement. The scenario is starting from different initial postures $[15,\ 45]$, $[15,\ 25]$ and $[35,\ 15]$ with initial $[\theta,\ \gamma] = [10^\circ,\ 7.5^\circ]$ and the goals are located at $[65,\ 75]$, $[95,\ 65]$ and $[65,\ 25]$ respectively with $d_{\min} = 1.0$ m, $d_{obs} = 3$ m

obstacles produces distortions from following the straight line, connecting the robot with the final goal point, while the sensing and safety radius are having a major effect on the path calculation. As it can observed in Fig. 11.12, the safety radius plays a very significant role in shape of the path. In this Figure, different paths with different safe distances $d_{\min}$ and with the same sensing radius d_{obs} are being presented. In the case that the vehicle is moving in a bounded space, the selection of a relatively big safety radius introduces oscillations in the translation of the robot due to sequential safety violations that produce corresponding change in the direction of the vehicle for avoiding the obstacle. In case that small safety radius are being selected, this effect vanishes and smooth and shorter, non oscillatory, paths can be produced.

It should be stated that the arenas in Figs. 11.10 and 11.11 are typical realistic examples of areas where articulated vehicles operate, as the mine tunnels and the civil

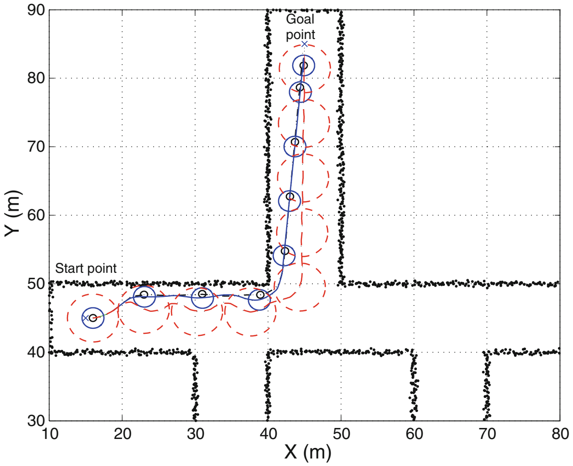


Fig. 11.12 During these simulations the vehicle is starting from the initial posture $[15,\ 45,\ 0,\ 5°]$ and the goal point is located at $[45,\ 85]$, while during movement different safe distances $d_{\min}$ have been utilized as 0.5, 1.5 and 3.5m, with the same $d_{obs} = 3$ m

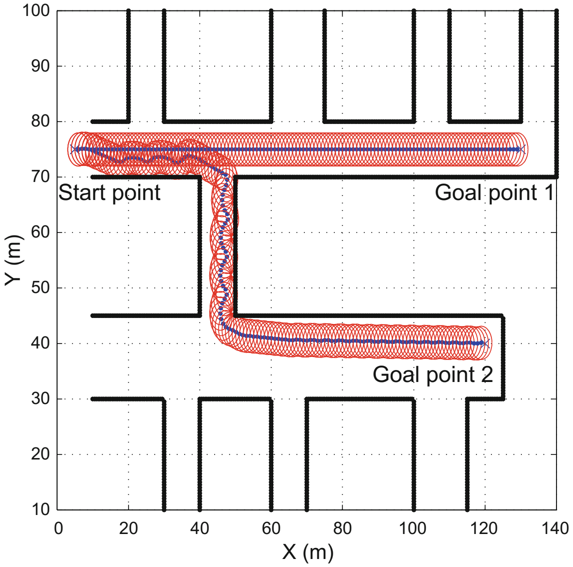


Fig. 11.13 Different boundaries on both sides of the road, the scenario is starting from the same initial postures $[5,\ 75]$ with initial $[\theta,\ \gamma] = [0°,\ 7.5°]$ and the goals are located at $[130,\ 75]$ and $[120,\ 40]$ respectively with $d_{\min} = 3.0$ m, $d_{obs} = 2$ m

roads are. In the presented simulations the consideration of the articulated vehicle's dynamic motion is obvious especially in the time instances where the vehicle is turning towards the goal and while performing at the same time obstacle avoidance. This effect is of paramount importance for the case of articulated vehicles as classical point dynamic approaches in path planning will obviously results in non-realistically achievable paths that would directly lead to collisions.

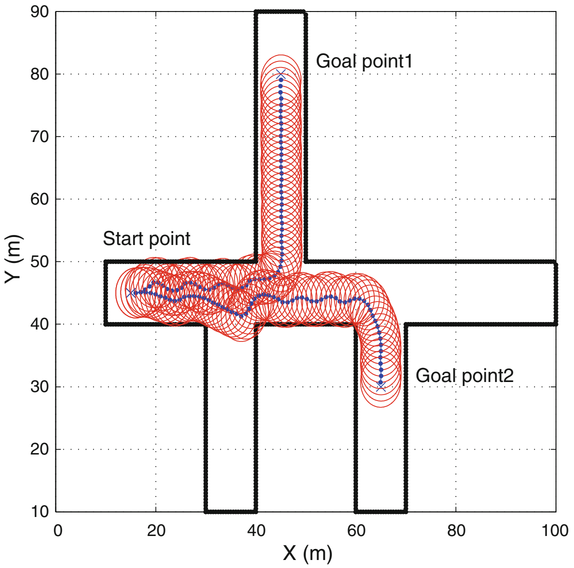


Fig. 11.14 Path planning in different arena having different boundaries on both sides of the road. The vehicle starting from the same initial postures $[5,\ 45]$ with initial $[\theta,\ \gamma] = [10°,\ 7.5°]$ and the goals are located at $[45,\ 80]$ and $[65,\ 30]$ respectively with $d_{\min} = 2.0$ m, $d_{obs} = 3$ m

11.5 Conclusions

In this article a novel online dynamic smooth path planning scheme based on a bug like modified path planning algorithm for an articulated vehicle under limited and sensory reconstructed surrounding static environment has been proposed. In the presented approach factors such as the real dynamics of the articulated vehicle, the initial and the goal configuration, the minimum and total travel distance between the current and the goal points, the geometry of the operational space, and the path smothering approach based on Bezier lines have been taken under consideration to produce a proper path for an articulated vehicle, which can be followed by.

References

1. Scheding, S., Dissanayake, G., Nebot, E., Durrant-Whyte, H.: An experiment in autonomous navigation of an underground mining vehicle. IEEE Trans. Robot. Autom. **15**(1), 85–95 (1999)
2. Roberts, J., Duff, E., Corke, P., Sikka, P., Winstanley, G., Cunningham, J.: Autonomous control of underground mining vehicles using reactive navigation. In: Robot. Autom., 2000. Proceedings ICRA'00. IEEE international conference on vol. 4, IEEE, pp. 3790–3795 (2000)
3. Nilsson, N.: A mobile automaton: an application of artificial intelligence techniques. Technical report, DTIC Document (1969)
4. Li, G., Yamashita, A., Asama, H., Tamura, Y.: An efficient improved artificial potential field based regression search method for robot path planning. In: Mechatronics and automation (ICMA), 2012 international conference on, IEEE, pp. 1227–1232 (2012)

5. Guechi, E., Lauber, J., Dambrine, M.: On-line moving-obstacle avoidance using piecewise bezier curves with unknown obstacle trajectory. In: Control and automation, 2008 16th Mediterranean conference on, IEEE, pp. 505–510 (2008)
6. Usher, K.: Obstacle avoidance for a non-holonomic vehicle using occupancy grids. In: 2006 Australasian conference on robotics and automation (2006)
7. Ge, S., Cui, Y.: New potential functions for mobile robot path planning. IEEE Trans. Robot. Autom. **16**(5), 615–620 (2000)
8. Škrjanc, I., Klančar, G.: Optimal cooperative collision avoidance between multiple robots based on bernstein-bézier curves. Robot. Autonom. syst. **58**(1), 1–9 (2010)
9. Ng, J., Bräunl, T.: Performance comparison of bug navigation algorithms. J. Intell. Robot. Syst. **50**(1), 73–84 (2007)
10. Lumelsky, V., Stepanov, A.: Dynamic path planning for a mobile automaton with limited information on the environment. IEEE Trans. Autom. Control **31**(11), 1058–1063 (1986)
11. Kamon, I., Rimon, E., Rivlin, E.: Tangentbug: a range-sensor-based navigation algorithm. Int. J. Robot. Res. **17**(9), 934–953 (1998)
12. Kamon, I., Rivlin, E.: Sensory-based motion planning with global proofs. IEEE Trans. Robot. Autom. **13**(6), 814–822 (1997)
13. Lumelsky, V., Skewis, T.: Incorporating range sensing in the robot navigation function. IEEE Trans. Syst. Man Cybern. **20**(5), 1058–1069 (1990)
14. Buniyamin, N., Wan Ngah, W., Sariff, N., Mohamad, Z.: A simple local path planning algorithm for autonomous mobile robots. Int. j. syst. appl. Eng. dev. **5**(2), 151–159 (2011)
15. Nayl, T., Nikolakopoulos, G., Guastafsson, T.: Kinematic modeling and simulation studies of a lhd vehicle under slip angles. In: Computational Intelligence and Bioinformatics/755: Modelling, Identification, and Simulation, ACTA Press (2011)
16. Chaudhry, T., Gulrez, T., Zia, A., Zaheer, S.: Bézier curve based dynamic obstacle avoidance and trajectory learning for autonomous mobile robots. In: Intelligent systems design and applications (ISDA), 2010 10th international conference on, IEEE, pp. 1059–1065 (2010)

Chapter 12
Hybrid Metric-topological Mapping for Large Scale Monocular SLAM

Eduardo Fernández-Moral, Vicente Arévalo and Javier González-Jiménez

Abstract Simultaneous Localization and Mapping (SLAM) is a central problem for autonomous mobile robotics. Monocular SLAM is one of the ways to tackle the problem, where the only input information are the images from a moving camera. Current approaches for this problem have achieved a good balance between accuracy and density of the map, however, they are not suited for large scale. In this paper, we present a dynamic mapping strategy where the metric map is divided into regions with highly connected observations, resulting in a topological structure which permits the efficient augmentation and optimization of the map. For that, a graph representation where the nodes represent keyframes, and their connections are a measure of their overlapping, is continuously rearranged. The experiments show that this hybrid metric-topological approach outperforms the efficiency and scalability of previous approaches.

Keywords Monocular SLAM · Metric-topological mapping · Map partitioning

12.1 Introduction

Monocular SLAM is an appealing way of solving the localization and mapping problem in mobile robotics because cameras are inexpensive, compact, easy to calibrate and consume low power. During the last years monocular SLAM has advanced notably with the use of parallel processing and efficient algorithms for data association and map optimization. It has made possible that current state-of-the-art approaches can operate accurately in some large scale scenarios, facilitating its appli-

E. Fernández-Moral (✉) · V. Arévalo · J. González-Jiménez
MAPIR Group, Universidad de Málaga, E.T.S. Ingeniería de Informática-Telecomunicación,
Campus de Teatinos, 29071 Málaga, Spain
e-mail: eduardofernandez@uma.es
http://mapir.isa.uma.es/

V. Arévalo
e-mail: varevalo@uma.es

J. González-Jiménez
e-mail: javiergonzalez@uma.es

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_12

cation in a wide range of areas such as augmented reality, scene reconstruction and, particularly, mobile robotics.

The increasingly larger maps that are now possible with monocular SLAM are fundamental to cope with a wider range of real autonomous robotics applications. Such ability to operate in large scale brings the need of appropriate strategies for managing the map. Applying abstraction (as humans do) is an effective way of dealing with the huge amount of detail present in large metric maps. The result of such abstraction process is the so-called metric-topological map, consisting of a two-layer representation, one containing pure geometrical information and a second one containing higher level symbolic information [25].

The benefit of a metric-topological arrangement is twofold: on the one hand, it offers a natural integration with symbolic planning that permits a robot to reason about the world and to execute high level tasks [10]. On the other hand, the efficiency and scalability of the SLAM process itself are improved by limiting the scope of localization and mapping to the region of the environment where the robot is operating. Also, loop closure and relocalisation can be more efficiently solved using topological information [1, 9, 20].

In this work, we present an online submapping technique which creates a topological representation of the world from the metric map being built by a monocular SLAM technique[1]. The key idea of our proposal is to cluster in the same submap those keyframes with higher observation overlap. This presents some important advantages over other approaches (as it will be explained latter on). The generated map consists of a topological structure composed of nodes representing local metric maps and arcs representing relative geometric transformations among the so-called submaps. In this paper, we will focus on the benefits of such a hybrid map for improving the efficiency and scalability of conventional (metric) monocular SLAM, concretely PTAM [12].

Next, we discuss some relevant related work and explain in detail the advantages of our approach. We then describe our partitioning procedure and show how it is combined with the SLAM process (PTAM). The experiments and their results are presented next, and finally, we expose the conclusions of our work.

12.2 Related Works

12.2.1 Construction of the Metric Map

Many solutions have been presented to build metric maps with monocular SLAM since Davison [5] presented the first real-time solution for the problem in 2003. Two main strategies have been applied since then: Bayesian filtering (following the work

[1] A preliminary version of this paper was presented in the "10th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Reykjavík (Iceland), 2013" [8]

of Davison) and Bundle Adjustment (BA) on keyframes, as introduced in [12]. The latter represents the base for the current state of the art since it allows handling denser maps and generally offers a better ratio accuracy/cost [24].

BA, traditionally used as an offline method for Structure from Motion (SfM), is now widely used in visual SLAM thanks to the introduction of parallel processing and efficient algorithms which exploit the sparse structure of the problem. Its application to visual SLAM was inspired by real time visual odometry and tracking [18], where the most recent camera poses where optimized to achieve accurate localization. In such line, PTAM selects keyframes and applies BA in a fixed size window, around the last keyframe incorporated, to obtain good metric maps and accurate localization. Then, once the local optimization is performed, a low priority global BA is run to improve the map consistency. This approach is extended in [11] by combining it with relative bundle adjustment—RBA—[22], allowing fixed-time, consistent exploration. An improvement of the latter to exploit the problem' sparse structure was recently presented by [4].

The work of [23] is also related to RBA, they propose a double window optimization: a first window as in PTAM and a second one including the periphery of the first to improve consistency by optimizing a pose-graph. Despite the impressive results obtained, such unique map solution has intrinsic limitations for managing maps of real large environments. To avoid such a limitation, we propose a topological arrangement in local metric maps.

12.2.2 *Dividing the Map*

Map division has been addressed in a number of works. Some relevant examples are: the Atlas framework [15], where a new local map is started whenever localization performs poorly in the current local map, or the hierarchical SLAM presented in [7], where sensed features are integrated into the current local map until a given number of them is reached. However, none of these provides a mathematically grounded solution based on the particular perception of the scene.

In [6], the map is divided in nodes where the landmarks are represented in a local coordinate frame and, these landmarks are updated using an information filter. This method uses the common features between adjacent nodes to calculate their relative pose. A different approach called Tectonic-SAM [17] uses a "divide and conquer" approach with locally optimized submaps in a Smoothing and Mapping framework (SAM). This approach is improved in [16] to build a hierarchy of multiple-level submaps using nested dissection.

Other works employ "graph cut" to divide the map according to a measurable property of the map observations. On that mathematical sound basis, [26] addresses the problem of automatic construction of a hierarchical map from images; [2] generates metric-topological maps using a range scanner, and generalizes the approach for other sensors; and [19] splits the map within a Bayesian monocular SLAM framework to reduce the problem complexity.

Our method, which also relies on graph cut, differs from the above works in the way the graph is constructed, which is specifically tailored for BA-based monocular SLAM. Our approach resembles also the stereo-SLAM framework of [14] who divide the map keyframes into groups (called segments) according to their geodesic distances in the graph. On the contrary, our map partitioning is independent of the keyframe positions, and is only based on observations acquired from the scene. Concretely, the map is split where there are less shared observations, minimizing the loss of information and therefore, enforcing the coherency and consistency of the submaps.

12.3 Map Partitioning

Splitting a map into locally metric consistent and globally coherent regions provides some relevant advantages for SLAM. Next, we explain the benefits of such map structure (Sect. 12.3.1), and describe our proposal to obtain this metric-topological arrangement of the map (Sect. 12.3.2).

12.3.1 SLAM Improvements Through Hybrid Mapping

The advantages of applying a coherent map partition in monocular SLAM are diverse: (a) all the metric data in each submap can be referred to a local coordinate system, what reduces error accumulation and numerical instability; (b) localization can be achieved more efficiently since only those map points in the nearer regions are reprojected to estimate the camera position; (c) this map structure permits to approximate the global BA by the individual optimization of the different submaps, thus reducing the computational cost of the optimization process. This last advantage is of special relevance due to the demanding nature of BA, whose complexity ranges from linear to cubic in the number of keyframes depending on the particular point-keyframe structure [13]. Next, we explain the details of this approximation for the global optimization.

Having a map of n landmarks obtained from observations at m keyframes, bundle adjustment can be expressed as

$$\min_{\mathbf{a}_j,\,\mathbf{b}_i} \sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij}\, d(\mathbf{Q}(\mathbf{a}_j,\ \mathbf{b}_i),\ \mathbf{x}_{ij})^2 \tag{12.1}$$

where

- $d(\mathbf{x}, \mathbf{x}')$ denotes the Euclidean distance between the image points represented by vectors $\mathbf{x}$ and $\mathbf{x}'$,
- $\mathbf{a}_j$ is the pose of camera at keyframe j and $\mathbf{b}_i$ the position of landmark i,

- $\mathbf{Q}(\mathbf{a}_j,\ \mathbf{b}_i)$ is the predicted projection of landmark i on the image associated to keyframe j,
- $\mathbf{x}_{ij}$ represents the observation of the i-th 3D landmark on the image of keyframe j and,
- v_{ij} stands for a binary variable that equals 1 if landmark i is visible in keyframe j and 0 otherwise.

Lets now consider that the map is divided into N submaps, each submap, say k, containing m^k keyframes and n^k landmarks, with $k = \{1, \ldots, N\}$. Then, (12.1) can be rewritten as

$$\min_{\mathbf{a}_j^l,\, \mathbf{b}_i^l} \sum_{k=1}^{N} \sum_{l=1}^{N} \left(\sum_{i=1}^{n^k} \sum_{j=1}^{m^l} v_{ij}^{kl}\, d(\mathbf{Q}(\mathbf{a}_j^l, \mathbf{b}_i^k),\ \mathbf{x}_{ij}^{kl})^2 \right) \tag{12.2}$$

where the combination of subscript i and superscript k refers to the i-th landmark of the k-th submap (e.g., $\mathbf{b}_i^k$), and similarly l over j refers to the j-th keyframe of the l-th submap (e.g., $\mathbf{a}_j^l$). Taking into account the observations shared between submaps, this expression can be written as

$$\min_{\mathbf{a}_j^l,\, \mathbf{b}_i^k} \sum_{k=1}^{N} \left(\underbrace{\sum_{\substack{l=1 \\ l \neq k}}^{N} \sum_{i=1}^{n^k} \sum_{j=1}^{m^l} v_{ij}^{kl}\, d(\mathbf{Q}(\mathbf{a}_j^l,\ \mathbf{b}_i^k),\ \mathbf{x}_{ij}^{kl})^2}_{A} + \underbrace{\sum_{i=1}^{n^k} \sum_{j=1}^{m^k} v_{ij}^{kk}\, d(\mathbf{Q}(\mathbf{a}_j^k,\ \mathbf{b}_i^k),\ \mathbf{x}_{ij}^{kk})^2}_{B} \right) \tag{12.3}$$

where the term A stands for the reprojection error of those landmarks observed from keyframes of different submaps and the term B corresponds to the reprojection error of those landmarks observed form keyframes within the same submap. Both concepts are illustrated in Fig. 12.1b. The first establishes the inter-connection between submaps which is represented by arcs connecting keyframes of different submaps (e.g. arc linking KF-2 and KF-11) and the second sets the intra-connection of the submap which includes the submaps inner arcs (e.g. arc linking KF-1 and KF-2).

If we are able to divide the map in such a way that the different submaps have few common observations, and assuming that the reprojection errors are independent of the map division, then A becomes negligible with respect to B. Thus, the global optimization can be approximated by

$$\sum_{k=1}^{N} \left(\min_{\mathbf{a}_j^k,\, \mathbf{b}_i^k} \sum_{i=1}^{n^k} \sum_{j=1}^{m^k} v_{ij}\, d(\mathbf{Q}(\mathbf{a}_j,\ \mathbf{b}_i),\ \mathbf{x}_{ij})^2 \right) \tag{12.4}$$

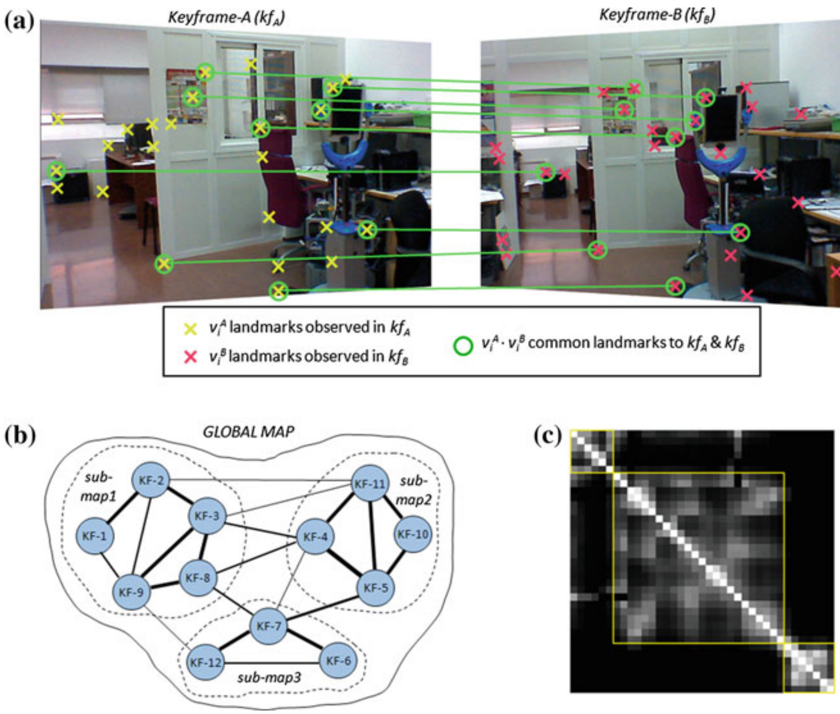


Fig. 12.1 **a** Common observations between two keyframes. This is used to calculate the Sensed Space Overlap (*SSO*) (see Eq. 12.5). **b** Graph-representation of the map where each node represents a keyframe and the arcs are weighed with the SSO calculated between keyframes (*thicker arcs* mean higher SSO). **c** Example of SSO matrix, in which the brightness of the element ij represents the SSO between the keyframes i and j

This approximation is equivalent to optimize each submap independently, which leads to a significant reduction of computational burden. In fact, this approximation is equivalent to the original expression (12.1) when there are no connections between submaps.

12.3.2 Map Partitioning Method

The approach proposed here to divide the map into coherent regions consists in grouping together those keyframes that observe the same features from the environment. For that, we consider the map as a graph whose nodes represent keyframes and the weight of the arcs are a measure of the common observations between them. There are two critical issues in this partitioning approach: first, the computation of the arc weights; and second, the criterion adopted to perform the partition itself. As for the first, the arc weights are assigned according to the Sensed-Space-Overlap (SSO), following our previous work [3], particularized for landmark observations. This simple but effective measure represents the information shared by two keyframes. It is calculated with the relation between the number of common landmark observations and the total number of landmarks observed in both keyframes (see Fig. 12.1a). This is expressed as

$$SSO\left(kf_A, kf_B\right) = \frac{\sum v_i^A \cdot v_i^B}{\sum v_i^A + \sum v_i^B - \sum v_i^A \cdot v_i^B} \tag{12.5}$$

where v_i^A and v_i^B, similarly to the definitions of the previous section, are binary variables that equal 1 if landmark i is observed in the keyframes kf_A and kf_B, respectively.

Regarding the criterion for partitioning the graph, we follow previous works [2, 19, 26] that apply the minimum normalized-cut (min-Ncut), originally introduced in [21]. The min-Ncut has the desirable property of generating balanced clusters of highly interconnected nodes, in our case clusters of keyframes that cover the same part of the environment. Figure 12.1 illustrates this concept: Fig. 12.1a shows the common observations in a pair of keyframes whose arc weight is calculated with the SSO (see Eq. 12.5), and Fig. 12.1b shows a map division into three submaps as produced by the min-Ncut procedure. Notice that the pairs of keyframes with higher SSO (thicker arcs) are grouped together. Figure 12.1c shows the symmetrical SSO matrix corresponding to a different, larger map, where the keyframes are arranged according the min-Ncut to give rise to three groups of keyframes or submaps (matrix blocks).

It is important to notice that, in order to guarantee a scalable system when applying map partitioning to visual SLAM, the size of the submaps (i.e. number of keyframes) must be kept bounded. This requirement is not demonstrated mathematically here, but it is intuitive to see that as the camera explores new parts of the scene, the new keyframes will have low SSO values (if any) with distant ones in the map. Therefore, the min-NCut will produce new partitions when the system explores unobserved regions of the environment. This can be more clearly understood with the following example: lets consider the case where there are features that are always observed (e.g. the horizon when travelling by train, or when zooming in the scene, or traversing a corridor with the camera pointing in the movement direction) as the new keyframes are selected, they will introduce new features and therefore will reduce the minimum normalized-cut, resulting in the eventual partition of the map. The last two examples represent another advantage of our partition method, which produces natural multi-scale maps when the camera zooms. This insight is supported by all the experiments we have carried out during this work.

12.4 Dynamic Division of PTAM's Metric Map

This section outlines the combination of our partition procedure and Parallel Tracking and Mapping (PTAM) [12]. PTAM is a monocular SLAM algorithm which performs online BA on keyframes, separating the tracking and mapping stages in two different threads to permit efficient real-time performance. This technique requires an initial map before it starts working automatically. Such initial map is acquired with a Structure from Motion procedure that involves user intervention to select two views with sufficient parallax. Once the initial map has been created, the system analyses the images retrieved by the camera to self-localize in the map, while the map is continuously optimized and augmented with new keyframes and landmarks. Such keyframes are selected according to some simple heuristics (see [12] for more details), and new landmarks are extracted through epipolar search between each new keyframe and its nearest keyframe in the map (Fig. 12.2).

12.4.1 Keyframes Selection in Large Environments

The keyframe selection criteria becomes an important aspect when PTAM is employed to build maps of large spaces. PTAM was designed for small environments (e.g. an office), where it works adequately with a hand-held camera which is waved sideways. PTAM employs a heuristic rule to select a new keyframe when there is a minimum separation between the current frame and the nearest keyframe in the map (i.e. Euclidean distance divided by the mean depth of the scene). This condition selects valid keyframes when the camera is moved sideways. But unlike in PTAM, we wish to explore big scenes and to construct large maps without being restricted to move sideways. Therefore, we have adapted this heuristic to select a keyframe when it provides useful information for mapping, by adding two more restrictions

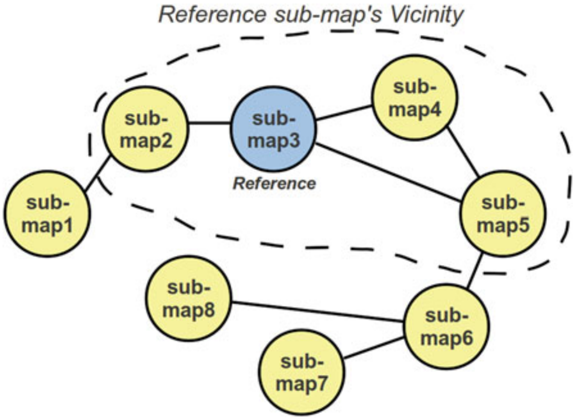


Fig. 12.2 Topological representation of the concept of submap vicinity

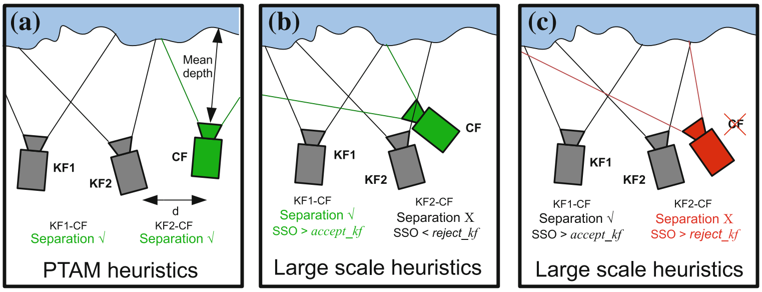


Fig. 12.3 Keyframe selection heuristics. **a** PTAM's separation condition. **b** and **c** Keyframe acceptance and rejection heuristics, respectively, for large scale mapping. The thresholds used in our experiments for $accept\_kf$ and $reject\_kf$ are 0.2 and 0.7 respectively

to the previous one for camera movement. Consequently, the current frame (CF) is selected as a new keyframe when:

- There must exist a keyframe (in the vicinity) that meets PTAM's separation condition with CF and which shares enough information of the scene ($SSO > accept\_kf$).
- There is not a keyframe (in the vicinity) that does not meet PTAM's separation condition with CF and which shares much information of the scene ($SSO > reject\_kf$).

Figure 12.3 shows the adapted heuristics to select keyframes in large scale. PTAM's separation condition is shown in Fig. 12.3a, where a keyframe is accepted when the Euclidean distance to the nearest keyframe divided by the mean depth of the scene is over some defined threshold. Figure 12.3b shows the new acceptance condition, which selects the current frame if there exist at least one keyframe that fulfils PTAM's separation and whose $SSO > accept\_kf$ (KF1-CF). Figure 12.3c shows the rejection condition, which rejects the current frame if there exist at least one keyframe that does not fulfil PTAM's separation and whose $SSO > reject\_kf$ (KF2-CF).

So, the acceptance condition prevents taking a new keyframe which shares little or no information with the map, while the rejection condition avoids selecting keyframes that are too similar to those already in the map. Hence, the combination of these two conditions permits selecting keyframes that provide new information to the map relaxing the movement constraints for nimble exploration of the scene.

12.4.2 Combination of Map Partitioning and PTAM

A scheme of the proposed partitioning method interacting with PTAM is depicted in Fig. 12.4. Our submapping procedure takes action in both of PTAM threads. In the tracking thread, it selects the current submap and the nearest keyframe to the

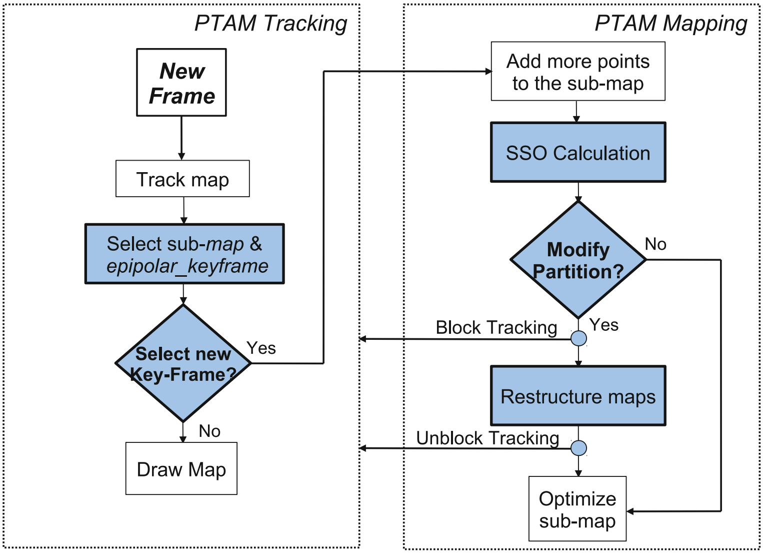


Fig. 12.4 Tracking and mapping threads of PTAM. *Blue boxes* correspond to the embedded stages to perform the map partitioning

estimated pose after a new image is analyzed. In the mapping thread, after a new keyframe is selected and new landmarks are detected in it, the SSO is evaluated with respect to all the keyframes of the vicinity. Such vicinity includes all the submaps directly connected to the current submap (see Fig. 12.2).

The partitioning procedure comes into play after the SSO has been updated, then, the min-Ncut is evaluated, and if it results in a different partition, the map is rearranged. This procedure is described in Algorithm 1. This partitioning method is applied dynamically as the map enlarges and may create new submaps as well as merge existing submaps to maintain coherency by grouping keyframes with high overlap. The result is a metric-topological map, where two different topological areas will be connected by a rigid transformation if there are common observations between them.

The partitioning process, including SSO computation, min-NCut evaluation and map rearrangement depends on the number of keyframes and landmarks in the vicinity, taking up to 100 ms. in our experiments, which supposes a short time in comparison with the map optimization time.

12.4.3 Experimental Results

In this section we present some experiments which show the advantages, in terms of efficiency and scalability, of using the proposed metric-topological arrangement of the map instead of a single metric map. The experiments have been carried out using a Philips SPC640NC webcam, connected by USB to a linux-based laptop with an Intel Core2 Duo 2.4 GHz processor, 2Gb of memory and a nVidia GeForce-9400 graphics card. Figure 12.5 shows the set up of our monocular SLAM system.

A first experiment is aimed to illustrate the increase of efficiency in localization at frame rate. For that, we compare the time needed to project map points into the current frame with and without partitioning as the map grows. Both tests have been performed in the same environment, building maps composed of about 45,000 points and 1,000 keyframes, distributed in 52 submaps for the partitioning case. Figure 12.6 shows that the time with a unique map grows linearly with the number of map points, whereas with submapping, this time is bounded since only those points in submaps close to the camera are evaluated. This improvement in efficiency becomes more relevant when the map grows nonstop (note that this process is performed with each new frame captured by the camera).

The goal of a second experiment is to quantify the efficiency in the global optimization of the map with our submapping approximation. For that, we have run BA

Algorithm 1. Map Partitioning.

M and KF are a submap and a keyframe respectively. SSO_M is the matrix containing the SSO values between all pairs of keyframes in the vicinity V. The $current\_map$ is the submap being tracked. num_KF is a keyframes counter and N_part is a parameter to control when the partition is to be reevaluated. A keyframe's $match\_map$ is the submap where it will be added, and a keyframe's $match\_KF$ is the keyframe used to find point correspondences.

After new keyframe new_KF is selected

```
$num\_KF++$
Select $match\_map$ and $match\_KF$
if $match\_map \mathrel{!}= current\_map$ then
  $num\_KF = 0$
end
Extract new map-points
Add a new row and a new column to $SSO\_M$
for all submaps $M_i \in V$ do
  for all keyframes $KF_j$ of $M_i$ do
    $SSO\_M \leftarrow SSO(new\_KF, KF_j)$
  end
end
if $(num\_KF \;\%\; N\_part) == 0$ then
  Evaluate partition
  if partition is modified then
    Lock tracking thread
    for all submaps $M_i \in V$ do
      Restructure $M_i$
    end
    Unlock tracking thread
    Update $SSO\_M$
  end
end
```

Fig. 12.5 Experimental set up: laptop with attached camera

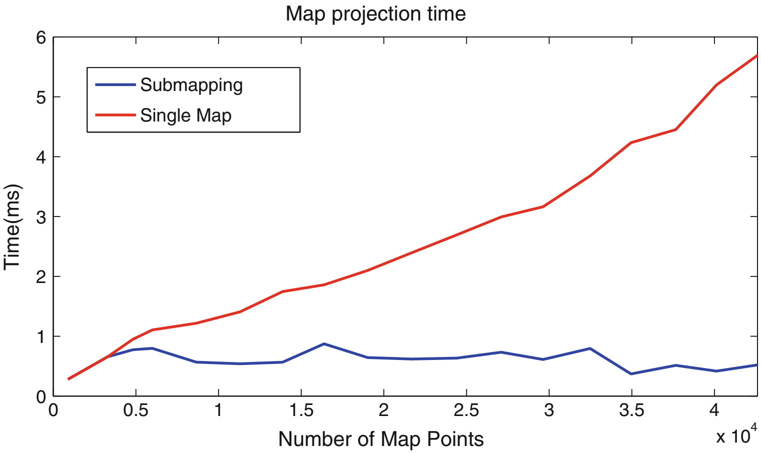


Fig. 12.6 Map projection time for localization with and without map partitioning

offline after every new keyframe is selected from a recorded video (that is, sequential SfM), measuring the times of each BA completion with and without partitioning. At the end of these tests, the maps created were composed of about 22,000 points and 400 keyframes, distributed in 9 submaps for the partitioning case. In order to compare both alternatives in the same conditions, we have included the time of partition management in the BA time for the partitioning test. Figure 12.7 shows the optimization times *versus* the number of keyframes of the whole map for both cases.

As expected, for the case without partitioning, the computational cost follows an increasing polynomial trend with the number of keyframes. Conversely, when applying map partitioning, the computational burden is bounded since the BA is applied only on the current submap. For this case, we can observe some abrupt changes in the cost which are produced when the reference submap (the one where

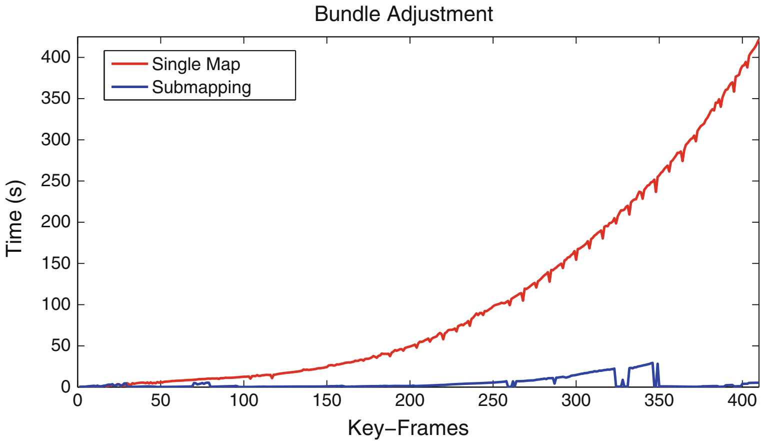


Fig. 12.7 Bundle adjustment computation time (offline) with and without partitioning

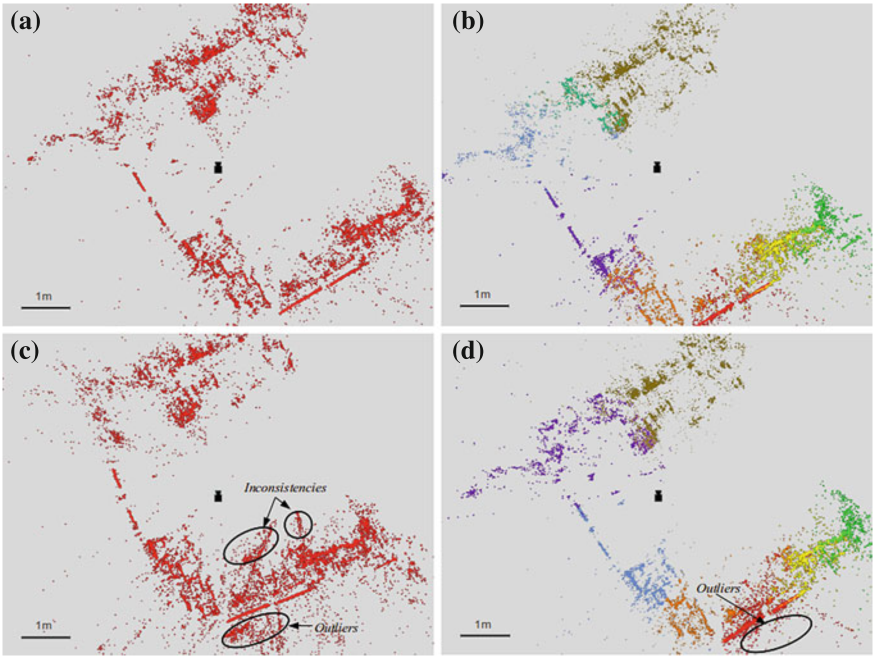


Fig. 12.8 *Top view* of maps generated in our experiments. All the maps are composed of more than 400 keyframes and 22,000 landmarks. The different colors in (**b**) and (**d**) represent different submaps

the system is localized) switches to a neighbor of different size. Figure 12.8a, b show the maps built with both alternatives (different colors represent different submaps in Fig. 12.8b). We can verify visually their high similarity, and their good alignment, as a result of the continuous optimization previous to the map partition.

Additionally, we are interested in comparing the accuracy of the generated metric map. Due to the lack of a reliable metric to evaluate the maps quality, we have compared visually the different maps considering as ground truth the map obtained offline in the previous experiment (Fig. 12.8a), which is the most accurate we can get. In the map obtained with PTAM (Fig. 12.8c), we can appreciate some regions with depth errors and many outliers (e.g. landmarks detected behind physical walls). These inconsistencies are consequence of the premature interruption of global BA that happens when a new keyframe is selected, what leads to data association errors and the subsequent accuracy decrease with the map size. On the contrary, the map obtained with our approach (Fig. 12.8d) presents no inconsistencies and considerably less outliers than the unique map solution (Fig. 12.8c). This results from the higher efficiency of the submap local optimization, which optimizes regions with highly correlated observations to produce locally accurate submaps.

The results shown in this section have been supported in several tests performed under different conditions: exploring different rooms, re-visiting previous maps, traversing a corridor, zooming to get more detail of the scene, etc. The reader may refer to http://youtu.be/-zK05EcOjX4 for a video that illustrates the operation of our submapping approach with PTAM in different environments.

12.5 Conclusions

This article presents an online submapping method which transforms a metric map into a metric-topological arrangement of it. This hybrid metric-topological structure improves the scalability of monocular SLAM in two aspects: first, the system rules out unnecessary metric information to perform more efficiently; second, it permits to use an approximation of BA to reduce computational cost while maintaining map consistency. Besides, the topological arrangement of the map is useful for other tasks, as loop closure, global localization or navigation. Experiments have demonstrated the potential of our approach to obtain efficient map representation in large environments. Future work will focus on exploiting the topological structure of the map for tasks as loop closure and relocalisation.

Acknowledgments This work has been supported by the project "TAROTH: New developments toward a robot at home", funded by the Spanish Government and the "European Regional Development Fund ERD" under contract DPI2011-25483.

References

1. Angeli, A., Doncieux, S., Meyer, J.A., Filliat, D.: Visual topological slam and global localization. In: IEEE International Conference on Robotics and Automation (2009)
2. Blanco, J.L., Fernández-Madrigal, J.A., González, J.: Towards a unified a bayesian approach to hybrid metric-topological slam. IEEE Trans. Robot. **24**(2), 259–270 (2008)
3. Blanco, J.L., González, J., Fernández-Madrigal, J.A.: Consistent observation grouping for generating metric-topological maps that improves robot localization. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 818–823 (2006)
4. Blanco, J., González-Jiménez, J., Fernández-Madrigal, J.: Sparser relative bundle adjustment (srba): constant-time maintenance and local optimization of arbitrarily large maps. In: IEEE International Conference on Robotics and Automation (2013)
5. Davison, A.J.: Real-time simultaneous localisation and mapping with a single camera. In: Proceedings of the International Conference on Computer Vision (ICCV) (2003)
6. Eade, E., Drummond, T.: Monocular slam as a graph of coalesced observations. In: International Conference on Computer Vision (2007)
7. Estrada, C., Neira, J., Tardos, J.: Hierarchical slam: real-time accurate mapping of large environments. IEEE Trans. Robot. **21**(4), 588–596 (2005)
8. Fernández-Moral, E., González-Jiménez, J., Arévalo, V.: Creating metric-topological maps for large-scale monocular slam. In: International Conference on Informatics in Control, Automation and Robotics (ICINCO) (2013)
9. Fernández-Moral, E., Mayol-Cuevas, W., Arévalo, V., González-Jiménez, J.: Fast place recognition with plane-based maps. In: IEEE International Conference on Robotics and Automation (2013)
10. Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., Fernández-Madrigal, J., González, J.: Multi-hierarchical semantic maps for mobile robotics. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2278–2283 (2005)
11. Holmes, S., Sibley, G., Klein, G., Murray, D.: A relative frame representation for fixed-time bundle adjustment in monocular sfm. In: IEEE International Conference on Robotics and Automation (2009)
12. Klein, G., Murray, D.W.: Parallel tracking and mapping for small ar workspaces. In: Proceedings of the International Symposium on Mixed and Augmented Reality (2007)
13. Konolige, K.: Sparse sparse bundle adjustment. In: British Machine Vision Conference (2010)
14. Lim, J., Pollefeys, M., Frahm, J.M.: Online environment mapping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011)
15. Newman, P., Leonard, J., Soika, M., Feiten, W., Teller, S.: An atlas framework for scalable mapping. IEEE Int. Conf. Robot. Autom. **2**, 1899–1906 (2003)
16. Ni, K., Dellaert, F.: Multi-level submap based slam using nested dissection. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2010)
17. Ni, K., Steedly, D., Dellaert, F.: Tectonic sam: exact, out-of-core, submap-based slam. In: IEEE International Conference on Robotics and Automation (2007)
18. Nistér, D., Naroditsky, O., Bergen, J.R.: Visual odometry. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 652–659 (2004)
19. Rogers, J.G., Christensen, H.I.: Normalized graph cuts for visual slam. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2009)
20. Savelli, F., Kuipers, B.: Loop-closing and planarity in topological map-building. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, pp. 1511–1517 (2004)
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
22. Sibley, G., Mei, C., Reid, I., Newman, P.: Adaptive relative bundle adjustment. In: Robotics Science and Systems Conference (2009)
23. Strasdat, H., Davison, A., Montiel, J., Konolige, K.: Double window optimisation for constant time visual slam. In: IEEE International Conference on Computer Vision (ICCV) (2011)

24. Strasdat, H., Montiel, J.M.M., Davison, A.J.: Scale drift-aware large scale monocular slam. Robot.: Sci. Syst. **2**(3), 5 (2010)
25. Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation. Artif. Intell. **99**(1), 21–71 (1998)
26. Zivkovic, Z., Bakker, B., Krose, B.: Hierarchical map building using visual landmarks and geometric constraints. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2480–2485 (2005)

Chapter 13
Novel Virtual Training System for Learning the Sway Suppression of Rotary Crane by Presenting Joystick Motion or Visual Information

Tsuyoshi Sasaki, Shoma Fushimi, Yong Jian Nyioh and Kazuhiko Terashima

Abstract In this paper, we propose a novel virtual training system capable of shortening the training period of unskilled crane operators. First, a simulator representing the motion behavior of load and boom during transfer operation in crane's cockpit is newly built. Second, referring to such the sway suppression skill taught in crane driving school, sway suppression control input is theoretically derived. Thirdly, a learning support method with ideal operation of joystick motion or visual sensory information to facilitate acquisition of the sway-suppression skill for unskilled operators is proposed. Finally, a lot of experiments were performed to validate the effectiveness of the proposed learning support method.

Keywords Rotary crane · Oscillations · Human-machine interface · Virtual reality · Teaching

13.1 Introduction

Rotary cranes are widely used at factories, harbors, and construction sites to load and unload cargo. Figure 13.1 shows a rotary crane. A rotary crane performs boom rotation, boom hoisting and load hoisting. Owing to its simple structure, a rotary crane can be easily disassembled, transported, and reassembled. Another big advantage of a rotary crane is that the very large workspace is achieved with a relatively small footprint. However, owing to acceleration or deceleration and centrifugal force, load sway is often generated during transport operations. When load sway is generated, it brings the problems on the accuracy of load to target position, work efficiency and safety. To solve these problems, it becomes important for crane operators to acquire sway suppression skill, and furthermore, acquisition of the skill in short training

T. Sasaki · S. Fushimi (✉) · Y.J. Nyioh · K. Terashima
The Mechanical Engineering Department, System and Control Laboratory, Toyohashi University of Technology, Hibarigaoka 1-1, Tempaku, Toyohashi, Japan
e-mail: fushimi@syscon.me.tut.ac.jp

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_13

Fig. 13.1 A rotary crane in the construction site

period is also needed. In regard to how this skill is acquired, training methods are frequently employed in which actual crane are used, but such methods involve a risk of accidents during the training period. In view of the safety hazard, various virtual crane simulators have been developed that enable training to be conducted without the training of actual cranes. In Huang and Gau [1] the development of the training simulator with high realistic sensation where the beginner operator could learn the skill of crane operation beforehand is attracting much attention. In Daqaq and Nayfeh [2], a virtual simulation of a ship-mounted crane is carried out in Cave Automated Virtual Environment (CAVE). A six degrees of freedom motion base was used to simulate the motion of a ship. The simulation serves as a platform for studying the dynamics of ships and ship-mounted cranes under dynamic sea environments, and also as a training platform for operators of ship-mounted cranes. Although those simulators perform well in terms of realistic sensation, the training function is insufficient. Thus, unskilled operators have to acquire the sway suppression skill by trial and error. As a result, a long period of training is needed to acquire the sway suppression skill. Terashimas group researches about shipboard crane training simulator for beginners [3, 4]. However, these basic studies are unmatched with the actual cockpit view of a crane, because the simulator is built as the operator view position fixed around a crane. In addition to this fact, the interface of the training system was only the one with the presence of ideal operation of joystick information, and therefore the comparison and consideration of interface with the presence of other information were highly demanded.

The purpose of the present study is to develop a simulator capable of shortening the training period for unskilled crane operators for rotary cranes. First, we create a virtual crane simulator using a rotary crane model. Display on crane of simulator is given by viewing from the cockpit which is rotated by crane, while the display on crane of simulator in the authors former researches [3, 4] was given by the fixed cockpit such that cockpit was set on the ground. Next, sway suppression control input is derived theoretically. Thirdly, using this control input, we propose novel two learning support methods that present ideal operation of joystick or visual sensory information to facilitate acquisition of the sway suppression skill for unskilled operators. For the former presentation of ideal operation of joystick information, control input with anti-sway against centrifuged force is reproduced by using an active joystick. Active

Joystick is automatically moved by using the inverse kinematics of joysticks motor model, and operators can naturally learn the ideal operation by holding joystick. On the other hand, for the latter presentation of visual information, control input is shown by an indicator on computer display. Unskilled crane operators are able to acquire the sway suppression skill by spontaneously operating the joystick following to the visual guidance from the indicator. The usefulness of the proposed method is demonstrated through various simulation experiments.

13.2 Dynamics of the Rotary Crane

The motion of rotary crane is different from the linear motion of an overhead crane or a gantry crane. In the case of a rotary crane, the motion of the load has an arc-like trajectory, and considering the effect of centrifugal force, it is necessary to model the load sway as a circular cone pendulum. A diagrammatic illustration of a rotary crane is shown in Fig. 13.2. In addition, the system is simplified by the following assumption. A crane is a rigid body and, considering the load is a mass point, the rope's weight, deflection and elasticity are ignored. The friction and backlash for the power transmission device are ignored. Boom tip position and load position are represented by Eqs. (13.1) and (13.2). The equation of swing angle of a load is represented by Eqs. (13.3) and (13.4) [5]. Boom tip trajectory:

$$\begin{cases} \tilde{x} = L_B \cos\theta \cos\phi \\ \tilde{y} = L_B \sin\theta \cos\phi \\ \tilde{z} = H + L_B \sin\theta \end{cases} \tag{13.1}$$

Load position:

$$\begin{cases} x = \tilde{x} + l \sin\alpha \\ y = -\tilde{y} - l \cos \sin\beta \\ z = \tilde{z} - l \cos\alpha \cos\beta \end{cases} \tag{13.2}$$

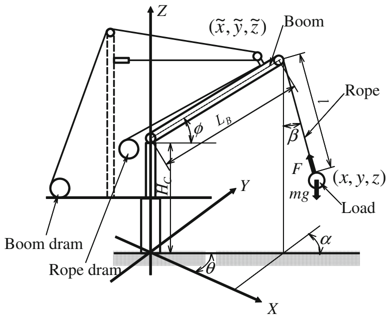


Fig. 13.2 Schematic of rotary crane for a load position model

Model of swing angle of the load:

$$\begin{aligned}\ddot{\alpha}l(\cos\alpha + \sin\alpha\tan\alpha) + \ddot{\beta}l\sin\alpha\tan\beta \\ + \dot{\beta}^2 l\sin\alpha - 2\dot{\alpha}\dot{\beta}l\sin\alpha\tan\alpha\tan\beta + g\sec\beta\tan\alpha \\ = -\ddot{\theta}l\cos\alpha\sin\beta + \dot{\theta}^2(L_B + l\sin\alpha) + 2\dot{\theta}l(\dot{\alpha}\sin\alpha\sin\beta - \dot{\beta}\cos\alpha\cos\beta),\end{aligned} \tag{13.3}$$

$$\begin{aligned}\ddot{\beta}l(\cos\alpha\cos\beta + \sin\beta\cos\alpha\tan\beta) - \dot{\alpha}\dot{\beta}(\cos\beta\sin\alpha + \sin\alpha + \sin\beta\cos\alpha\tan\beta) \\ + g\tan\beta + \dot{\beta}^2 l\sin\alpha - 2\dot{\alpha}\dot{\beta}l\sin\alpha\tan\alpha\tan\beta + g\sec\beta\tan\alpha \\ = \ddot{\theta}(L_S + l\sin\alpha) + \dot{\theta}^2 l\cos\alpha\sin\beta + 2\dot{\theta}\dot{\alpha}l\cos\alpha.\end{aligned} \tag{13.4}$$

where $\tilde{x}$, $\tilde{y}$, $\tilde{z}$ [m] is three-dimensional coordinate of Boom tip position, x, y, z [m] is

three-dimensional coordinate of load position, L_B [m] is length of the boom, θ[rad] is rotary angle, ϕ [rad] is Boom angle, l[m] is length of rope, α [rad] is sway angle of radius direction, and β [rad] is sway angle of slew direction.

13.3 Construction of Virtual Simulator

In this section, a virtual crane simulator using crane model is built. The present visual simulator consists of displayed graphics of crane boom and load, and joystick displayed graphics for operation. The graphics on computer display is created using OpenGL. Operational view of this simulator configures from a cockpit of a crane, and translates its view position with slew motion of crane cockpit. Operational interface (device) for virtual simulator uses Active Joystick (will be explained later). Flow of this simulator is shown in Fig. 13.3. First, read velocity input from Active Joystick. Second, it calculates operational amounts of rotary, boom hoisting, and load hoisting from velocity input. Thirdly, it calculates the states of Boom tip and load using crane model. Finally, it makes a static graphic of a crane, and renders its graphics at 30msec intervals. By repeating this flow, it is displayed crane graphic naturally on real-time. Figure 13.4 shows virtual crane simulator built with Active Joystick.

13.4 Derivation of Sway Suppression Control Input

The load sway of a rotary crane is affected by acceleration or deceleration of the boom and centrifugal force, because the boom is rotated. Thus, load sway becomes two-dimensional sway consisting of radius direction sway and slew direction sway

Fig. 13.3 Flow of simulator

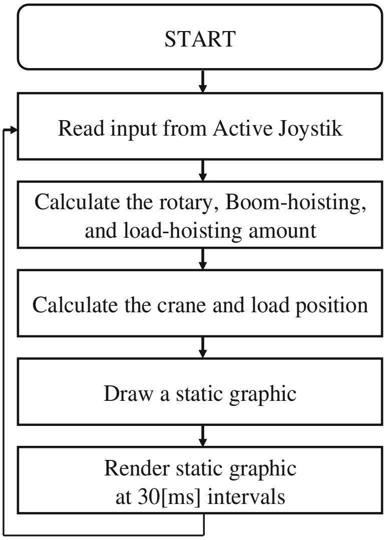
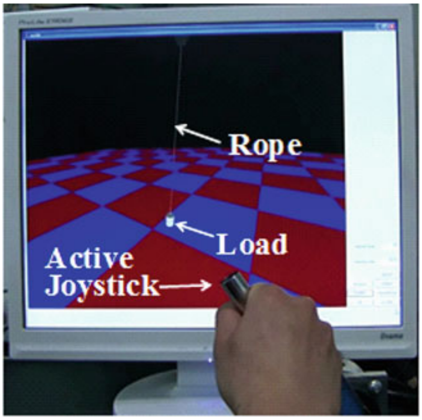

START
Read input from Active Joystik
Calculate the rotary, Boom-hoisting, and load-hoisting amount
Calculate the crane and load position
Draw a static graphic
Render static graphic at 30[ms] intervals

Fig. 13.4 Visual crane simulator



(see Fig. 13.5). Additionally, in the case that the boom rotates 90 degrees, the slew direction sway mutates into the radius direction sway from initial position to target position in absolute coordinates. Given these facts, we use the 2-Mode Input Shaping method by Shighose et al. [6], in which velocity variation changes in three steps for the anti-sway. Because the Input shaping control method is very intuitive one, it is considered that it is easier for operators to train the anti-sway control input compared with other methods (see Fig. 13.6). The optimal velocity A_i[rad/s] and the timing of velocity variation t_i[s] are derived to minimize residual sway. This method can control the residual radius direction sway and slew direction sway by only rotary actuator. Time t_p[s] is the arbitrary time at which the crane is commanded to begin

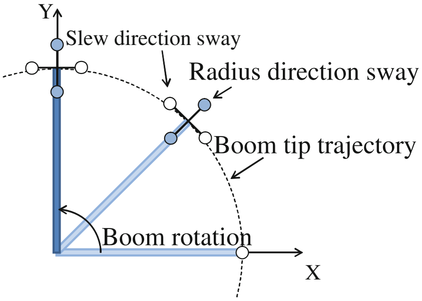


Fig. 13.5 Characteristics of load sway for a rotary crane

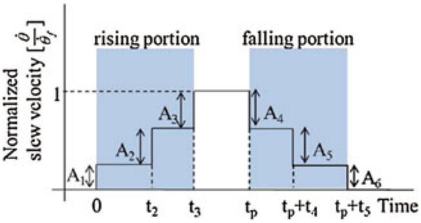


Fig. 13.6 2-mode input shaping command template

decelerating. The vertical axis is normalized by the final setpoint slew velocity $\dot{\theta}_f$ [rad/s] yielding

$$\begin{cases} A_1 + A_2 + A_3 = 1 \\ A_4 + A_5 + A_6 = 1 \end{cases} \tag{13.5}$$

This means that only amplitudes A_1, A_2, A_4 and A_5 need to be derived since A_3 and A_6 can be found directly from Eqs. (13.5) and (13.6). The relational expression of slew velocity and sway angle becomes Eq. (13.7).

$$\begin{bmatrix} \dot{\beta} \\ \ddot{\beta} \\ \dot{\alpha} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ (\dot{\theta}^2 - \omega_0^2) & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & -2\dot{\theta} & (\dot{\theta}^2 - \omega_0^2) & 0 \end{bmatrix} \begin{bmatrix} \beta \\ \dot{\beta} \\ \alpha \\ \dot{\alpha} \end{bmatrix} \tag{13.6}$$

Here, α and β have high-frequency modes and low-frequency modes respectively from characteristic value of state equation. Here, the sum of high-frequency modes defines v_{jk} [rad], and the sum of low-frequency modes defines y_{jk} [rad], ω_0 [rad/s] is natural frequency. These modes can be expressed by Eqs. (13.8) and (13.9):

$$\begin{cases} v_{jk} = -\dfrac{L_B}{2l\omega_0}\left[\dfrac{\dot{\theta}_k\omega_0}{\omega_k^+} + \dot{\theta}_j\left(-1 + \dfrac{\dot{\theta}_j\omega_k^-}{\omega_j^+\omega_j^-}\right)\right] \\ y_{jk} = \dfrac{L_B}{2l\omega_0}\left[\dfrac{\dot{\theta}_k\omega_0}{\omega_k^-} - \dot{\theta}_j\left(1 + \dfrac{\dot{\theta}_j\omega_k^-}{\omega_j^+\omega_j^-}\right)\right] \end{cases} \tag{13.7}$$

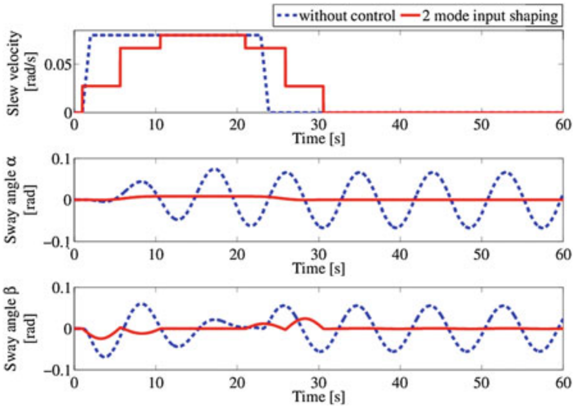


Fig. 13.7 Simulation result for 2-mode input shaping

where,

$$\omega_j^+ = \omega_0 + \dot{\theta}_j, \quad \omega_k^+ = \omega_0 + \dot{\theta}_k,$$
$$\omega_j^- = \omega_0 - \dot{\theta}_j, \quad \omega_k^- = \omega_0 - \dot{\theta}_k$$

Here, $L_B = 17.5$ [m], $l = 20$ [m], $\dot{\theta}_j$ is slew velocity before input A_i, $\dot{\theta}_j$ is slew velocity before input A_i, ω_j^+ [rad/s] and ω_j^- [rad/s] are the high and low modes at $\dot{\theta}_f$ respectively, while ω_k^+ [rad/s] and ω_k^- [rad/s] are the high and low modes at $\dot{\theta}_k$ respectively. By recursively applying Eqs. (13.8) and (13.9) to the command template in Fig. 13.6, the complex valued residual sway of two modes, v_{tot} [rad], and y_{tot} [rad] can be derived as follow:

$$\begin{cases} v_{tot} = v_{01}\exp[i\{\omega_1^+ t_2 + \omega_2^+(t_3 - t_2)\}] + v_{12}\exp[i\omega_2^+(t_3 - t_2)] + v_{23} \\ y_{tot} = y_{01}\exp[i\{\omega_1^+ t_2 + \omega_2^+(t_3 - t_2)\}] + y_{12}\exp[i\omega_2^+(t_3 - t_2)] + y_{23} \end{cases} \tag{13.8}$$

where v_{tot} and y_{tot} are found from Eqs. (13.8) and (13.9) (Eqs. (13.5)–(13.11) details; see the original papers of Singhose, et al. [7].) These terms are the change in the complex amplitudes of first and second caused by a step transition from $\dot{\theta}_f$ to $\dot{\theta}_k$,

$$\dot{\theta}_0 = 0, \dot{\theta}_1 = \dot{\theta}_f A_1, \dot{\theta}_2 = \dot{\theta}_f (A_1 + A_2), \dot{\theta}_3 = \dot{\theta}_f \tag{13.9}$$

where A_1, A_2, t_2, and t_3 refer to the step amplitudes and times shown in Fig. 13.6.

In rising portion (acceleration interval), it needs to derive A_1, A_2, t_2, and t_3 such that v_{tot} and y_{tot} are minimized. Similarly, it needs to derive A_4, A_5, t_4, and t_5 in falling portion (deceleration interval). This study minimizes using conjugate gradient method, and run a simulation. By simulation result in Fig. 13.4, it was able to confirm that 2-Mode Input Shaping reduce residual sway of the load. And that, the timing of changing velocity in second step and third step was turn out when the sway angle and a crane become vertical (Fig. 13.7).

13.5 Proposed Learning Assyst System

Instructors verbally explain the sway suppression techniques at the crane driving school. Beginners receive the explanation, and then practice crane operation by themselves. However, this training method will be not the sufficient training for beginners. In this section, using the 2-Mode Input Shaping method described in the previous section, a novel training system for beginners that teaches the amount and timing of acceleration or deceleration is presented. By teaching these operational skills, we hope that the training effect will be enhanced and the training period shortened for beginners. So, we propose the training system by giving sensory information of humans such as ideal operation of joystick or visual information. In this paper, two methods of teaching operational skills are proposed. One is ideal operation of joystick guidance training by presenting ideal operation of joystick information, and the other is visual guidance training by presenting visual information.

One training method we propose is often used in sports training and skills education. Such training through hands-on coaching is known to be effective in many situations. This study focused attention on this point, and proposes ideal operation of joystick guidance training by the joystick of operational interface for obtaining the sway suppression skill. Principle of this learning method is as follows. Namely, active joystick interface is automatically moved by using inverse kinematics of motor model from anti-sway reference velocity obtained from 2-Mode Input Shaping method. Then, beginners are able to learn a sense of the skill by touching its joystick with his or her hand and feeling the motion.

Figure 13.5 shows the Active Joystick that is the operational interface used in this study. This joystick is equipped with a 6-axis force sensor and AC servo motors. The joystick rotates on the X_J-axis and Y_J-axis. If the joystick is tilted on X_J-axis, it rotates the rotary crane, and if it is tilted on Y_J-axis, the boom is hoisted. In order to drive the joystick, 2 AC servo motors with harmonic drive (speed reduction ratio = 1:100) are utilized. A force/torque sensor is attached on the joystick to measure the force that the operator applies to the joystick (Fig. 13.8).

Furthermore, the joystick incorporates a spring mass damper model so that an operator can move the joystick using relatively little force and when the operator removes his hand from the joystick, the joystick automatically returns to its starting point. The joystick's motion equation is expressed as follows:

$$J_r\ddot{\theta}_J + d_r\dot{\theta}_J + k_r\theta_J = M_y \tag{13.10}$$

where, θ_J: joystick's inclination angle from original point, M_y: force applied on joystick by operator, $J_r = 0.1$ kg m^2: inertia moment, $d_r = 0.7$ Nms/rad: viscous friction coefficient, and $k_r = 1.65$ N/m: spring constant.

Figure 13.5 shows outline of learning assist system. This system converts the sway suppression control input into driving voltage of motors, and replicates its input by its joystick. Beginners can learn to the sway suppression skill sensuously, because

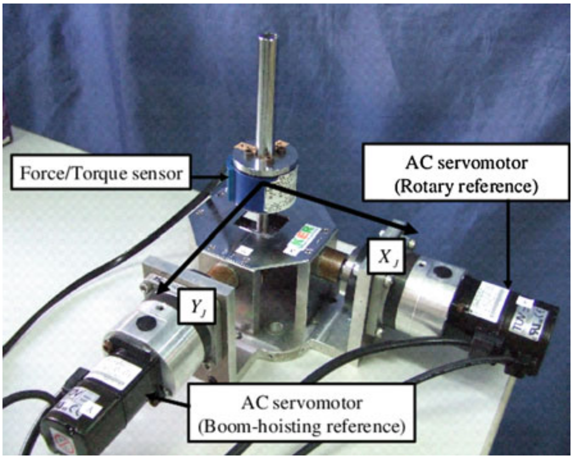


Fig. 13.8 Active joystick

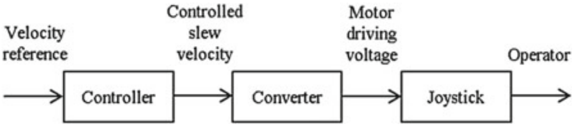


Fig. 13.9 Diagram of joystick motion guidance

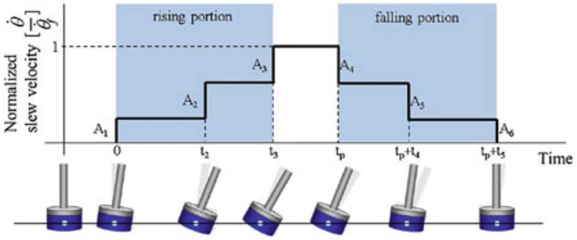


Fig. 13.10 Motion of active joystick for each time

they feel maneuvering feeling like getting coaching from expert. Thus, they will be able to achieve its operational amount and the timing by using this training (see Fig. 13.5, 13.9 and 13.10).

The other training method that we propose is visual guidance training involving the presentation of visual information. An indicator is displayed on the screen of the crane simulator shown in Fig. 13.5. This indicator shows the amount of acceleration or deceleration A_i on a three-step scale for control of the load sway. The height of the meter changes in proportion to joystick angle. Thus, beginners will be able to learn the amount and timing of acceleration or deceleration required by spontaneously manipulating the joystick in accordance with the scale.

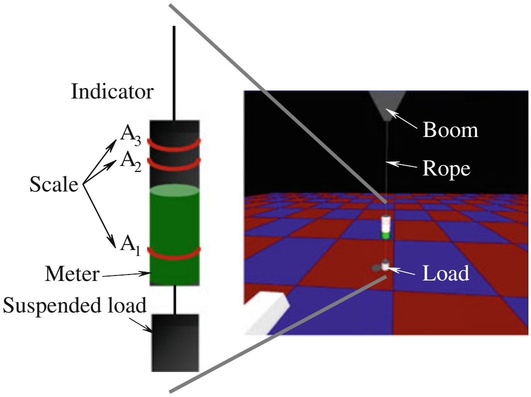


Fig. 13.11 Indicator for visual guidance

13.6 Result of Virtual Training Experiments and Discussion

The effectiveness of the proposed novel learning assist system was evaluated by means of an experiment. In this experiment, training was conducted for 13 men who had no experience of crane operation. The subjects were divided into the following four groups:

- Group A: 2 men, self-training, without oral presentation
- Group B: 3 men, self-training, with oral presentation
- Group C: 4 men, ideal operation guidance training
- Group D: 4 men, visual guidance training.

Before starting the experiment, the sway suppression skill was explained verbally to each group, excluding Group A. Using the crane simulator that we developed, each group transports a load (height: 0.9 m, radius: 0.3 m) several times. Each group transports the load to a circular target position (radius: 0.4 m) by turning the rotary crane 100deg. For automatic transportation using 2-Mode Input Shaping, transfer time is 29.6 s. However, for transportation by manual operation, it is almost certain that human error will occur. Thus, in consideration of human error, transfer time is set to within 35.0 s. Parameters of the rotary crane simulator are listed in Table 13.1, and parameters of 2-Mode Input Shaping shown in Table 13.2.

The training schedule is shown in Fig. 13.12. First, subjects of each group transport the load three times without assistance, and then they transport the load three

Table 13.1 Parameter of rotary crane for training use

Parameter	Symbol	Value	Units
Rope length	l	20.0	m
Boom length	L_B	17.5	m
Boom hoisting angle	ϕ	45.0	deg
Max slew velocity	θ	0.08	rad/s

Table 13.2 Parameters of 2-mode input shaping

A_1	A_2	t_2	t_3	A_4	A_5	t_4	t_5
0.3442	0.4902	4.6476	9.5657	0.1673	0.4912	4.9161	9.5701

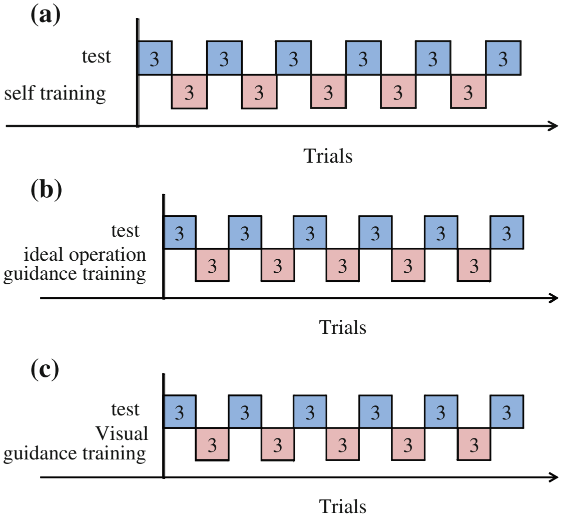


Fig. 13.12 Training schedule for each group

times with assistance. This flow is treated as one set, and this set is conducted five times. Finally, one set is conducted again without assistance, and residual sway of all test sets is evaluated. Training time of one set is about 3 minutes, all training time is about 40 minutes with break time. The learning effect is evaluated on the basis of residual sway. Figure 13.13 shows the average residual sway angle for trials of each group. Figure 13.13a shows that subjects in Group A were unsure about the crane operation because they were not given information about it. Figure 13.13b shows that the average residual sway angle for Group B was on a modest declining trend. However, because subjects in Group B were not informed of the amount and timing of acceleration or deceleration required, they had to ascertain it by trial and error. Therefore, the training effect for Group A and Group B was low. Figure 13.13c shows that the average residual sway angle for Group C steeply trended downward. As the subjects in Group C were able to experience the ideal sway suppression skill through ideal operation guidance, they were able to replicate it well. Therefore, we conclude that the ideal operation guidance training is effective. However, as shown in Figure 13.13d Group D's results were superior to those of the other groups, which is considered to be attributable to the superior effect of visual guidance training because it allows the subjects to recognize the disparity between actual and ideal input and rectify it by spontaneous joystick operation in real time.

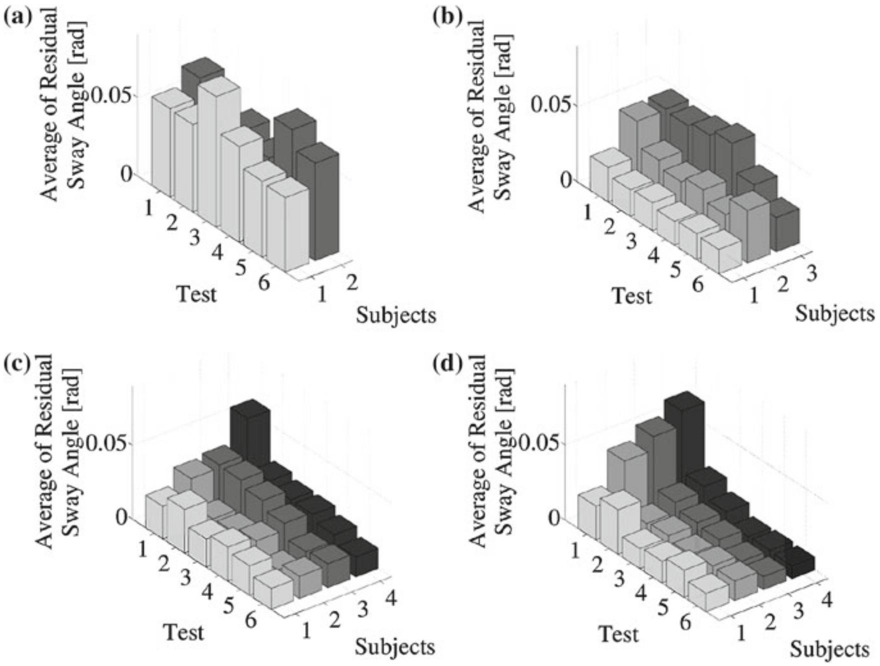


Fig. 13.13 Training result for each group. **a** Group A. **b** Group B. **c** Group C. **d** Group D

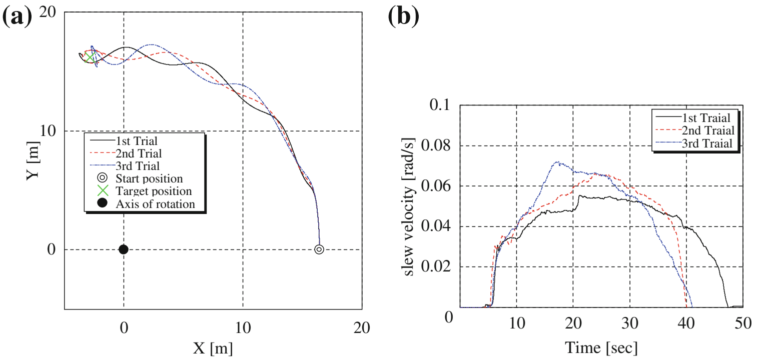


Fig. 13.14 Result of group A. **a** Load trajectory. **b** Slew velocity input

Figure 13.14 through Fig. 13.17 shows the slew velocity input and the load trajectory of the 6th test involving certain subjects of each group. Slew velocity of Group D is reproduced stably three times and load trajectory is comparatively smooth. Figure 13.6 shows the diminishing rate of residual sway by comparing the results of the first test with those of the final test. As can be seen from these results, the learning assist system will shorten the training period for beginner crane operators. In

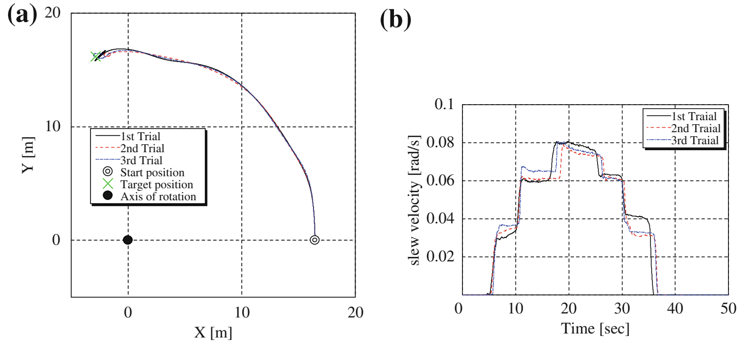


Fig. 13.15 Result of group B. **a** Load trajectory. **b** Slew velocity input

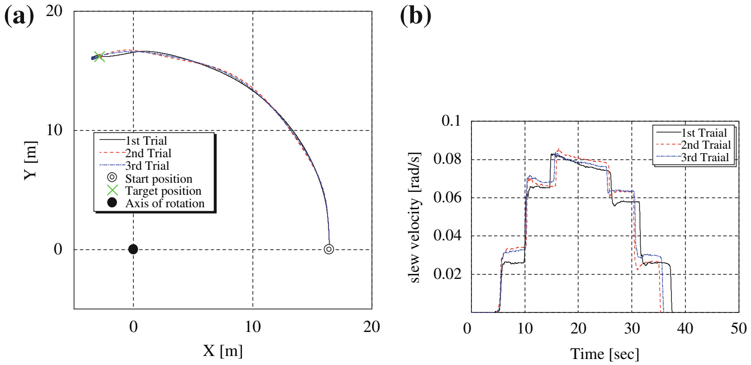


Fig. 13.16 Result of group C. **a** Load trajectory. **b** Slew velocity input

addition, the efficiency of work will increase because of a decrease in residual sway. Furthermore, we conducted simulation experiments for various lope length such as l =15 m, 25 m and l =30 m. In any cases, we obtained the results of learning effects as almost same as the results of l =20 m if we learn how to operate for anti-sway by using 2-Mode Input Shaping method for each order rope length. Now, an alternative method as training way will be discussed here. A way to present ideal movements of joystick proposed in this paper can certainly teach the motion of joystick, but not the force to push joystick. After learning how to move the joystick without force information, operator must operate joystick by grasping it. Then, operator needs not only motion, but also force reference information. Therefore, through the results of this study, a learning system is expected such that ideal force for anti-sway is memorized in computer, and a motion feedback is worked against operator's actual force input. Furthermore, a hybrid training system constructed of motion feedback and visual indicator proposed in this paper may be better. We will also present them in near future (Fig. 13.18).

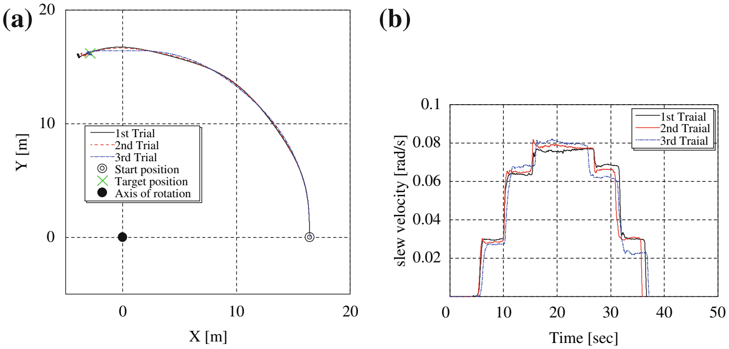
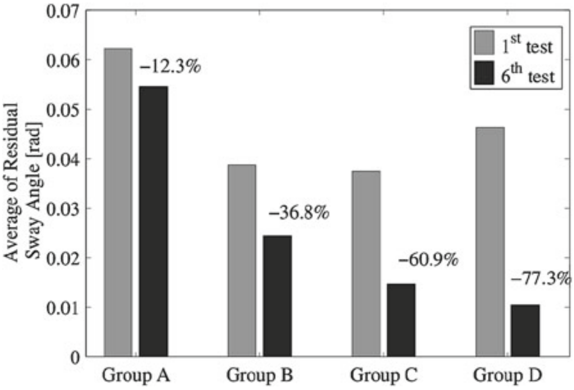


Fig. 13.17 Result of group D. **a** Load trajectory. **b** Slew velocity input

Fig. 13.18 Comparasion of residual sway before and after training



13.7 Conclusions

In this study, we built a virtual crane training simulator such as presenting sway suppression skill. The results are as follows.

1. A crane training simulator using rotary crane model has been built.
2. As operational interface for simulator, Active Joystick presenting ideal operation of joystick information is developed.
3. The sway suppression control using 2-Mode Input Shaping method is applied in this study, and sway suppression is well achieved.
4. Guidance training methods by presenting ideal operational joystick or visual information are proposed.
5. It was clear that visual guidance training decreased 77% of residual sway than conventional training in the same training time through many simulation experiments.

As future work, we apply this result to real experimental apparatus. Furthermore, we plan to extend to shipboard crane with the present training function. The operation of shipboard crane is highly required to get operational skill, because crane operator must consider complicated ship sway in addition to anti-sway on boom and load of crane.

References

1. Huang, J.Y., Gau, C.Y.: Modelling and designing a low-cost hight-fidelity mobile crane simulator. Int. J. Hum. Comput. Stud. **58**(2), 151–176 (2003)
2. Daqaq, M.F., Nayfeh, A.H.: Virtual reality simulation of ships and ship-mounted cranes. Master thesis, Virginia Polytechnic Institute and State University (2003)
3. Iwasa, T., Terashima, K., Jian, N.Y., Noda, Y.: Operator assistance system of rotary crane by gain-scheduled H-inf controller with reference governor. In: 2010 IEEE International Conference on Control Applications (CCA), Yokohama, Japan, pp. 1325–1330, 8–10 Sept 2010
4. Yong Jian, N., Noda, Y., Terashima, K.: Simulator building for agile control design of shipboard crane and its application to operational training. In: 18th World Congress of the International Federation of Automatic Control (IFAC), Universita Cattolica del Sacro Cuore, Milano, 28 Aug–2 Sept (2011)
5. Shen, Y., Terashima, K., Yano, K.: Optimal control of rotary crane using the straight transfer transformation method to eliminate residual vibration (2003)
6. Lawrence, J., Singhose, W.: Command shaping slewing motions for tower crane. J. Vib. Acous. **132**(1) (2010)
7. Feygin, D., Keehner, M., Tendick, F.: Haptic guidance: experimental evaluation of a haptic training method for a perceptual motor skill. In: Proceeding of the 10th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 40–47 (2002)
8. Morris, D., Tan, H., Barbagli, F., Chang, T., Salisbury, K.: Haptic feedback enhances force skill learning. In: Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 21–26 (2007)
9. Al-Hussein, M., Niaz, M.A., Yu, H., Kim, H.: Integrating 3D visualization and simulationnext term for tower previous termcranenext term operations on construction sites. Autom. Construct. **15**(5), 554–562 (2006)
10. Feygin, D., Keehner, M., Tendick, F.: Haptic guidance: experimental evaluation of a haptic training method for a perceptual motor skill. In: Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 40–47 (2002)
11. Terashima, K., Shen, Y., Yano, K.: Modeling and optimal control of a rotary crane using the straight transfer transformation method. Int. J. Control Eng. Pract. **15**, 1179–1192 (2007)
12. Shen, Y., Terashima, K., Yano, K.: Minimum time control of a rotary crane using straight transfer transformation method. Trans. Soc. Instrum. Control Eng. **3**(10), 70–79 (2004)
13. Yano, K., Ogura, N., Terashima, K.: Starting control with vibration damping by hybrid shaped approach considering time and frequency specifications. Trans. Soc. Instrum. Control Eng. **137**, 403–410 (2001)

Chapter 14
Navigation of Autonomous Mobile Robots with Unscented HybridSLAM

Jurek Z. Sasiadek, Amir H. Monjazeb and Dan Necsulescu

Abstract Simultaneous localization and mapping is a situation in which a mobile robot travels through an environment and concurrently makes a momentary map of the environment and uses that map to localize. The simultaneous localization and mapping is currently one of the most challenging problems in the field of autonomous mobile robots and providing a solution to SLAM may open doors to the world of truly autonomous robots. This paper provides a novel approach to Simultaneous Localization and Mapping problem based on estimation approach. The new approach is called Unscented HybridSLAM filter which addresses the linearization process of an autonomous mobile robot utilizing the second order Sterling polynomial interpolation specifically used for Unscented HybridSLAM algorithm. Using computer simulations, Unscented HybridSLAM and the associated theoretical interpolation is examined for a double-loop scenario and the efficacy of the Unscented HybridSLAM is validated.

Keywords Simultaneous Localization and Mapping (SLAM) problem · EKF · FastSLAM · HybridSLAM · Unscented hybridSLAM · Cluttered environment · Double loop closing · Absolute error

14.1 Introduction

The main task of a feature-based SLAM algorithm is to estimate the path of the robot and map of the environment as accurate as possible. There are many methods in which the robot uses different sensors to measure positions of landmarks as well as

J.Z. Sasiadek (✉) · A.H. Monjazeb
Department of Mechanical and Aerospace Engineering, Carleton University,
1125 Colonel by Drive, Ottawa, Canada
e-mail: jurek_sasiadek@carleton.ca

A.H. Monjazeb
e-mail: amonjaze@connect.carleton.ca

D. Necsulescu
Department of Mechanical Engineering,
Ottawa University, 161 Louis Pasteur, Ottawa CBY A205, Canada
e-mail: necsu@uottawa.ca

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_14

pose of the robot [12]. Sensor readings are analyzed in these methods to extract data from the active or passive features in the environment to match it with a-priori known information in order to determine the current position of the robot. Usually, the task of extracting and matching data with a-priori information is easy for a domestic environment in which landmarks are distributed evenly. If the robot has a notation of evenly distribution of landmarks, the extracting of such data would be rather easier. For some SLAM cases in which the robot is equipped with restricted sensors, a uniform distribution of landmarks would considerably reduce the ambiguity of data association in the environment [2]. The advantage in such cases would be the elimination of data extracted from wrongly observed landmarks. Since the robot is aware of a uniform set of landmarks, sensor readings that result more than a specific threshold would be automatically deleted from the estimation process as a result of the Maximum Likelihood Rule [11].

14.2 Sterling Polynomial Interpolation

The formulation of the second order Sterling Polynomial Interpolation (SPI) is the basis of derivation of the Divided Deference Filter (DDF) and the Central Difference Filter (CDF) [5]. To formulate the equations of the system in a linear form, the second order SPI will be discussed in this section to indicate how a non-linear system can be approximated in a linear form. Then, the mean and covariance of the system in the posterior state will be discussed. Based on Taylor series of a non-linear function [4], a random variable x around a statistical point $\bar{x}$ as its mean, can be expressed by

$$h(x) = h(\bar{x}) + D_{\delta_x} h + \frac{1}{2!} D^2_{\delta_x} h + \cdots = h(\bar{x}) + (x - \bar{x}) \frac{dh(x)}{dx} + \frac{1}{2!} (x - \bar{x})^2 \frac{d^2 h(x)}{dx^2} \tag{14.1}$$

The SPI formula [6] uses a finite number of functional evaluations to approximate the above non-linear function with $\tilde{D}_{\Delta_x}$ as the first and $\tilde{D}^2_{\Delta_x}$ as the second order central divided difference operators acting on h(x), ℓ is the interval length or central difference step size and $\bar{x}$ is the prior mean of x around which the expansion is done. The resulting formula can be expressed as

$$h(x) = h(\bar{x}) + \tilde{D}_{\Delta_x} h + \frac{1}{2!} \tilde{D}^2_{\Delta_x} h \tag{14.2}$$

$$\tilde{D}_{\Delta_x} = (x - \bar{x}) \frac{h(\bar{x} + \ell) - h(\bar{x} - \ell)}{2\ell} \tag{14.3}$$

$$\tilde{D}^2_{\Delta_x} = (x - \bar{x})^2 \frac{h(\bar{x} + \ell) + h(\bar{x} - \ell) - 2h(\bar{x})}{\ell^2} \tag{14.4}$$

In some cases [4], the SPI formula can be interpreted as the Taylor series. If this formula is extended to the multi dimensional case, the function $h(\boldsymbol{x})$ may be obtained by first stochastically decoupling the prior random variable x by the linear transformation as

$$\mathbf{y} = \mathbf{S}_x^{-1}\mathbf{x} \tag{14.5}$$

$$\tilde{h}(\mathbf{y}) = h(\mathbf{S}_x\mathbf{y}) = h(\mathbf{x}) \tag{14.6}$$

where $\mathbf{S}_x$ is called Cholesky factor [7] of the covariance matrix P_x of x such that $\mathrm{P}_x = S_x \mathrm{S}_x^T$. It should be noted that Taylor series expansion of $h(\cdot)$ and $\tilde{h}(\cdot)$ is identical if the expected value of vector x is E[x] and the covariance of the system is the expected value of P_x $=\mathrm{E}[(\mathbf{x}-\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}})^T]$, the transformation stochastically decouples variables in $\mathbf{x}$ so that the interval components of $\mathbf{y}$ becomes mutually uncorrelated.

$$\mathbf{P}_y = \mathrm{E}[(\mathbf{y}-\bar{\mathbf{y}})(\mathbf{y}-\bar{\mathbf{y}})^T] = \mathbf{I}. \tag{14.7}$$

Assuming that L is the dimension of $\mathbf{x}$ and $\mathbf{y}$ with $\Delta_{y_i} = (\mathbf{y}-\bar{\mathbf{y}})_{\mathbf{i}}$ as the ith component of $\mathbf{y}-\bar{\mathbf{y}}(i = 1, \ldots,\mathrm{L})$, e_i is the ith unit vector, $\boldsymbol{d}_i$ is the partial first order difference, $\boldsymbol{d}_i^2$ is the partial second order difference, and $\mathbf{m}_i$ is the mean operator [13]. Therefore,

$$\tilde{\mathbf{D}}_{\Delta_y}\tilde{h} = \left(\sum_{i=1}^{L} \Delta_{y_i}\boldsymbol{m}_i\boldsymbol{d}_i\right)\tilde{h}(\bar{\mathbf{y}}) \tag{14.8}$$

$$\tilde{\mathbf{D}}^2_{\Delta_y}\tilde{h} = \left(\sum_{i=1}^{L} \Delta^2_{y_i}\boldsymbol{d}_i^2 + \sum_{j=1}^{L}\sum_{q=1}^{L} \Delta_{y_j}\Delta_{yq}(\boldsymbol{m}_j\boldsymbol{d}_j)(\boldsymbol{m}_q\boldsymbol{d}_q)\right)\tilde{h}(\bar{\mathbf{y}}) \tag{14.9}$$

$$\boldsymbol{d}_i\tilde{h}(\bar{\mathbf{y}}) = \frac{1}{2\ell}\left[\tilde{h}(\bar{\mathbf{y}}+\ell\mathbf{e}_i) - \tilde{h}(\bar{\mathbf{y}}-\ell\mathbf{e}_i)\right] \tag{14.10}$$

$$\boldsymbol{d}_{\boldsymbol{i}}^2\tilde{h}(\bar{\mathbf{y}}) = \frac{1}{2\ell^2}\left[\tilde{h}(\bar{\mathbf{y}}+\ell\mathbf{e}_i) + \tilde{h}(\bar{\mathbf{y}}-\ell\mathbf{e}_i) - 2\tilde{h}(\bar{\mathbf{y}})\right] \tag{14.11}$$

$$\boldsymbol{m}_i\tilde{h}(\bar{\mathbf{y}}) = \frac{1}{2}\left[\tilde{h}(\bar{\mathbf{y}}+\ell\mathbf{e}_i) + \tilde{h}(\bar{\mathbf{y}}-\ell\mathbf{e}_i)\right] \tag{14.12}$$

using Eqs. (14.5) and (14.6) and considering that s_{x_i} is the ith column of the Cholesky factor of covariance matrix of $\mathbf{x}$ we can induce

$$\tilde{h}(\bar{\mathbf{y}} \pm \ell\mathbf{e}_i) = h(\mathbf{S}_x\left[\bar{\mathbf{y}} \pm \ell\mathbf{e}_i\right] = h(\mathbf{S}_x\bar{\mathbf{y}} \pm \ell\mathbf{S}_x\mathbf{e}_i) = h(\bar{\mathbf{x}} \pm \ell\mathbf{s}_{x_i}) \tag{14.13}$$

$$\mathrm{s}_{x_i} = \mathbf{S}_x\mathbf{e}_i = (\mathbf{S}_x)_i = (\sqrt{\mathbf{P}_x})_i \tag{14.14}$$

Set of vectors defined in Eq. (14.13) is equivalent so that that the UKF generates its set of sigma-points with only the difference in the value of the weighting term [8].

14.3 Posterior Mean and Covariance Estimation

The observation function can be expressed through a non-linear function $h(\cdot)$ and with considering non-linear transformation of an L dimensional random variable $\mathbf{x}$ with covariance $\mathbf{P}_x$ and mean $\bar{\mathbf{x}}$ as follows

$$\mathbf{z}_k = h(\mathbf{x}_k) = \tilde{h}(\mathbf{y}_k) \approx \tilde{h}(\mathbf{y}_k) + \tilde{\mathbf{D}}_{\Delta_y}\tilde{h} + \frac{1}{2}\tilde{\mathbf{D}}^2_{\Delta_y}\tilde{h} \tag{14.15}$$

$$\mathbf{y} = \mathbf{S}_x\mathbf{x} \tag{14.16}$$

The posterior mean of $\mathbf{y}$ and its covariance and cross covariance are defined as

$$\bar{\mathbf{z}}_k = \mathrm{E}[\mathbf{z}_k] \tag{14.17}$$

$$\mathbf{P}_{\mathrm{z_k}} = \mathrm{E}[(\mathbf{z}_k - \bar{\mathbf{z}}_k)(\mathbf{z}_k - \bar{\mathbf{z}}_k)^T] \tag{14.18}$$

$$\mathbf{P}_{\mathrm{x_k}z_k} = \mathrm{E}[(\mathbf{x}_k - \bar{\mathbf{x}}_k)(\mathbf{z}_k - \bar{\mathbf{z}}_k)^T] \tag{14.19}$$

Assuming that $\Delta_y = (\mathbf{y} - \bar{\mathbf{y}})$ is a zero-mean unity variance random variable which is symmetric [5] as defined in Eq. (14.5), the mean is approximated as

$$\bar{\mathbf{z}}_k \approx E[\tilde{h}(\mathbf{y}_k) + \tilde{\mathbf{D}}_{\Delta_y}\tilde{h} + \frac{1}{2}\tilde{\mathbf{D}}^2_{\Delta_y}\tilde{h}] \tag{14.20}$$

$$= \tilde{h}(\mathbf{y}_k) + E[\frac{1}{2}\tilde{\mathbf{D}}^2_{\Delta_y}\tilde{h}] \tag{14.21}$$

$$= \tilde{h}(\mathbf{y}_k) + E(\frac{1}{2\ell^2}\left(\sum_{i=1}^{L}\Delta^2_{y_i}\boldsymbol{d}_i^2)\tilde{h}(\mathbf{y}_k)\right) \tag{14.22}$$

$$= \tilde{h}(\mathbf{y}_k) + \frac{1}{2\ell^2}\sum_{i=1}^{L}[\tilde{h}(\bar{\mathbf{y}}_k + \ell\mathbf{e}_i) + \tilde{h}(\bar{\mathbf{y}}_k - \ell\mathbf{e}_i) - 2\tilde{h}(\mathbf{y}_k)] \tag{14.23}$$

$$= \frac{\ell^2 - \boldsymbol{L}}{\ell^2}\tilde{h}(\mathbf{y}_k) + \frac{1}{2\ell^2}\sum_{i=1}^{L}[\tilde{h}(\bar{\mathbf{y}}_k + \ell\mathbf{e}_i) + \tilde{h}(\bar{\mathbf{y}}_k - \ell\mathbf{e}_i)] \tag{14.24}$$

By rewriting the posterior mean in terms of motion vector [9] we will have

$$\bar{\mathbf{z}}_k = \frac{\ell^2 - L}{\ell^2}\tilde{h}(\mathbf{x}_k) + \frac{1}{2\ell^2}\sum_{i=1}^{L}[\tilde{h}(\bar{\mathbf{x}}_k + \ell s_{x_i}) + \tilde{h}(\bar{\mathbf{y}}_k - \ell s_{x_i})] \tag{14.25}$$

Using the identity

$$\begin{aligned}\bar{\mathbf{z}}_k = E[\mathbf{z}_k] &= E[\mathbf{z}_k] + h(\bar{\mathbf{x}}_k) - h(\bar{\mathbf{x}}_k) = E[\mathbf{z}_k] + h(\bar{\mathbf{x}}_k) - E[h(\bar{\mathbf{x}}_k)] \\ &= h(\bar{\mathbf{x}}_k) + E[\mathbf{z}_k - h(\bar{\mathbf{x}}_k)] \end{aligned} \tag{14.26}$$

$$\begin{aligned}
\mathbf{P}_{z_k} &= E[(\mathbf{z}_k - \bar{\mathbf{z}}_k)(\mathbf{z}_k - \bar{\mathbf{z}}_k)^T] \\
&= E[(\mathbf{z}_k - h(\mathbf{x}_k))(\mathbf{z}_k - h(\mathbf{x}_k))^T] - E[(\mathbf{z}_k - h(\mathbf{x}_k))] \\
E[(\mathbf{z}_k - h(\mathbf{x}_k))]^T &= E[(\mathbf{z}_k - \tilde{h}(\mathbf{y}_k))(\mathbf{z}_k - \tilde{h}(\mathbf{y}_k))^T] \\
&\quad - E[(\mathbf{z}_k - \tilde{h}(\mathbf{y}_k))]E[(\mathbf{z}_k - \tilde{h}(\mathbf{y}_k))]^T
\end{aligned} \tag{14.27}$$

From Eq. (14.15), the second order approximation of $\mathbf{z}_k - \tilde{h}(\mathbf{y}_k) = \tilde{\mathbf{D}}_{\Delta_y}\tilde{h} + \frac{1}{2}\tilde{\mathbf{D}}^2_{\Delta_y}\tilde{h}$ can be substituted into Eq. (14.27) and therefore,

$$\begin{aligned}
\mathbf{P}_{\mathrm{z_k}} \approx \mathrm{E}[(\tilde{D}_{\Delta_\mathrm{y}}\tilde{h} + \frac{1}{2}\tilde{\mathbf{D}}^2_{\Delta_\mathrm{y}}\tilde{h}) \times (\tilde{\mathbf{D}}_{\Delta_\mathrm{y}}\tilde{h} + \frac{1}{2}\tilde{\mathbf{D}}^2_{\Delta_\mathrm{y}}\tilde{h})^T] \\
- \mathrm{E}[(\tilde{\mathbf{D}}_{\Delta_\mathrm{y}}\tilde{h} + \frac{1}{2}\tilde{\mathbf{D}}^2_{\Delta_\mathrm{y}}\tilde{h})] \times E[(\tilde{\mathbf{D}}_{\Delta_\mathrm{y}}\tilde{h} + \frac{1}{2}\tilde{\mathbf{D}}^2_{\Delta_\mathrm{y}}\tilde{h})]^T
\end{aligned} \tag{14.28}$$

$\Delta_y = (\mathbf{y} - \bar{\mathbf{y}})$ is symmetric, therefore, all resulting odd-order expected moments have zero value. Since the number of terms in this calculation grows rapidly with the dimension of **y,** the inclusion of such terms leads the computation highly complex. As a result all components of the resulting fourth order term, $\mathrm{E}[\frac{1}{4}(\tilde{\mathbf{D}}^2_{\Delta_y}\tilde{h})\,(\tilde{\mathbf{D}}^2_{\Delta_y}\tilde{h})^T]$, that contains cross differences in the expansion of Eq. (14.28) are discarded. The extra effort worthwhile is not considered since it is not possible to capture all fourth order moments [3]. The approximation of the covariance and cross-covariance matrices are expressed as below.

In Eq. (14.31) the odd-order moment terms are all zero since $(\mathbf{y}_k - \bar{\mathbf{y}}_k)$ is symmetric. The optimal setting of the central difference interval parameter, ℓ, is dictated by the prior distribution of $\mathbf{y} = \mathbf{S}_x^{-1}\mathbf{x}$. For Gaussian priors, the optimal value of h is thus $h = \sqrt{3}$. For more details see [5].

$$\begin{aligned}
\mathbf{P}_{z_k} \approx \frac{1}{4\ell^2}\sum_{i=1}^{L}&[h(\bar{\mathbf{x}}_k + \ell \mathrm{s}_{x_i}) - h(\bar{\mathbf{x}}_k - \ell \mathrm{s}_{x_i})] \\
&\times [h(\bar{\mathbf{x}}_k + \ell \mathrm{s}_{x_i}) - h(\bar{\mathbf{x}}_k - \ell \mathrm{s}_{x_i})]^T \\
&+ \frac{\ell^2 - 1}{4\ell^4}\sum_{i=1}^{L}[[h(\bar{\mathbf{x}}_k + \ell \mathrm{s}_{x_i}) + h(\bar{\mathbf{x}}_k - \ell \mathrm{s}_{x_i}) - 2h(\bar{\mathbf{x}}_k)] \\
&\times [h(\bar{\mathbf{x}}_k + \ell \mathrm{s}_{x_i}) + h(\bar{\mathbf{x}}_k - \ell \mathrm{s}_{x_i}) - 2h(\bar{\mathbf{x}}_k)]^T]
\end{aligned} \tag{14.29}$$

$$\begin{aligned}
\mathbf{P}_{x_k z_k} &= E[(\mathbf{x}_k - \bar{\mathbf{x}}_k)(\mathbf{z}_k - \bar{\mathbf{z}}_k)^T] \\
&\approx E[(\mathbf{S}_x(\mathbf{y}_k - \bar{\mathbf{y}}_k)[\tilde{\mathbf{D}}_{\Delta_y}\tilde{h} + \frac{1}{2}\tilde{\mathbf{D}}^2_{\Delta_y}\tilde{h} - E[\frac{1}{2}\tilde{\mathbf{D}}^2_{\Delta_y}\tilde{h}]]^T] \\
&= E[(\mathbf{S}_x(\mathbf{y}_k - \bar{\mathbf{y}}_k)[\tilde{\mathbf{D}}_{\Delta_y}\tilde{h}]^T] \\
&\quad + \frac{1}{2}E[(\mathbf{S}_x(\mathbf{y}_k - \bar{\mathbf{y}}_k)[\tilde{\mathbf{D}}^2_{\Delta_y}\tilde{h}]^T]
\end{aligned} \tag{14.30}$$

$$-\frac{1}{2}E[(\mathbf{S}_x(\mathbf{y}_k-\bar{\mathbf{y}}_k)]\times E[\frac{1}{2}\tilde{\mathbf{D}}^2_{\Delta_y}\tilde{h}]^2$$

$$=E[(\mathbf{S}_x(\mathbf{y}_k-\bar{\mathbf{y}}_k)[\tilde{\mathbf{D}}_{\Delta_y}\tilde{h}]^T] \tag{14.31}$$

$$=\frac{1}{2\ell}\sum_{i=1}^{L}\mathbf{s}_{x_i}[\tilde{h}(\bar{\mathbf{y}}_k+\ell\mathbf{e}_i)-\tilde{h}(\bar{\mathbf{y}}_k-\ell\mathbf{e}_i)]^T \tag{14.32}$$

$$=\frac{1}{2\ell}\sum_{i=1}^{L}\mathbf{s}_{x_i}[h(\mathbf{x}_k+\ell\mathbf{s}_{x_i})-h(\mathbf{x}_k-\ell\mathbf{s}_{x_i})]^T \tag{14.33}$$

14.4 Simulations and Results

14.4.1 Landmark Estimation Threshold

Figure 14.1a shows a path in an environment with a non-uniform distribution of landmarks. Figure 14.1b depicts the range of position estimation of landmark at x = 30 and y = 20 m. The error in this case indicates that the estimated location of the landmark is within ±0.40 m. In this particular scenario, the level of data ambiguity does not arise exponentially when the distribution of landmarks change from uniform to random. Figure 14.2 compares the ambiguity of data with the use of EKF-SLAM as well as using 3000 particles resulted by FastSLAM, HybridSLAM, and Unscented HybridSLAM. Hundreds of dots that make different formations around in the range are depicted in this figure for each specific algorithm. The threshold range (oval) is obtained using a standard EKF under Gaussian conditions. The true position of

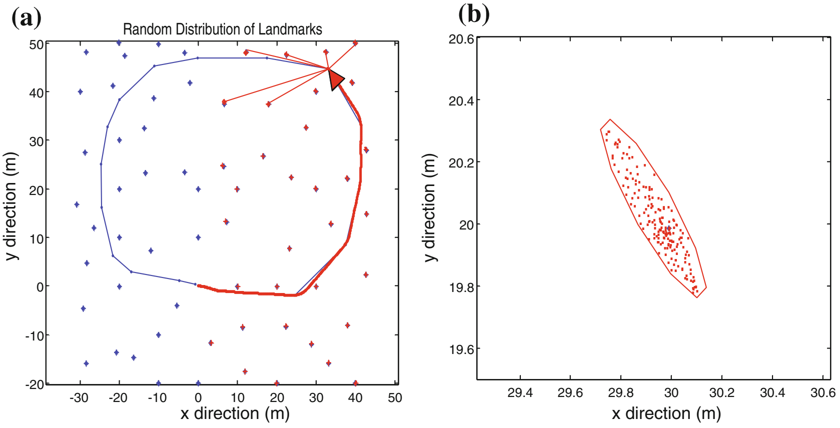


Fig. 14.1 Random distribution of landmarks **a** Non-uniform distribution of landmarks in the environment. **b** Estimated position of the landmark located at (x = 30, y = 20)

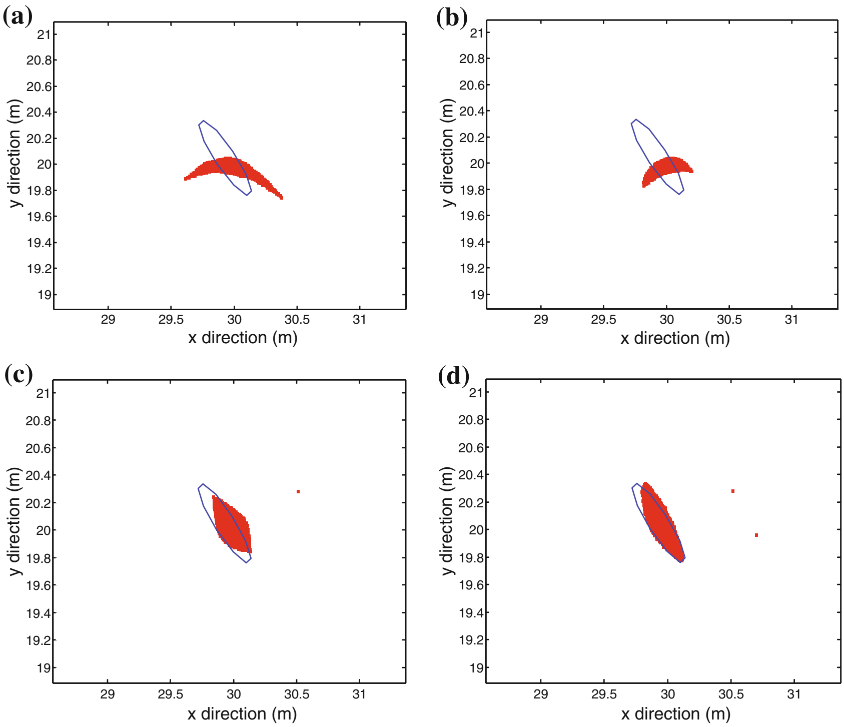


Fig. 14.2 Estimated position of the landmarks **a** EKF-SLAM under non-Gaussian conditions. **b** FastSLAM. **c** HybridSLAM. **d** Unscented HybridSLAM

the landmark is at x = 30 and y = 20 m. The banana shape in Fig. 14.2a, shows the estimation result using the first order Taylor series in EKF under non-Gaussian conditions which appears to be highly inaccurate.

The banana shape in Fig. 14.2b, illustrates a reduction of error in the location estimation of the landmark using FastSLAM and as a result less ambiguity in data. However, estimated points do not fit in the standard oval and there are about 60 % of estimated points off the standard threshold. HybridSLAM has relatively less ambiguity in data association as shown in Fig. 14.2c. As shown in the picture, there are only 30 % of points outside the range. Moreover, the estimation dots are mostly inside the standard range. Nonetheless, it is still far from the standard threshold and may not be an acceptable result for SLAM applications. The estimation of the landmark with Unscented Kalman Filter creates an oval shape around the true location of the landmark and is the one with the least ambiguity in data association. As demonstrated in Fig. 14.2d, about 15 % of estimated points are outside the standard range which proves that HS has the most acceptable result amongst all other algorithms. As a result, UHS is the only algorithm which is a recursive filter based on sterling approximation and has the least tendency to diverge. Figures 14.3, 14.4, and 14.5

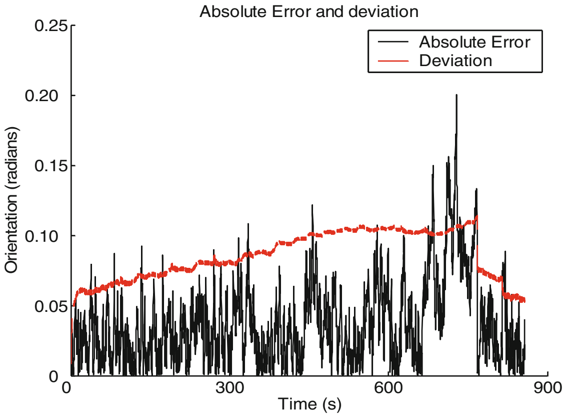


Fig. 14.3 Orientation absolute error and deviation

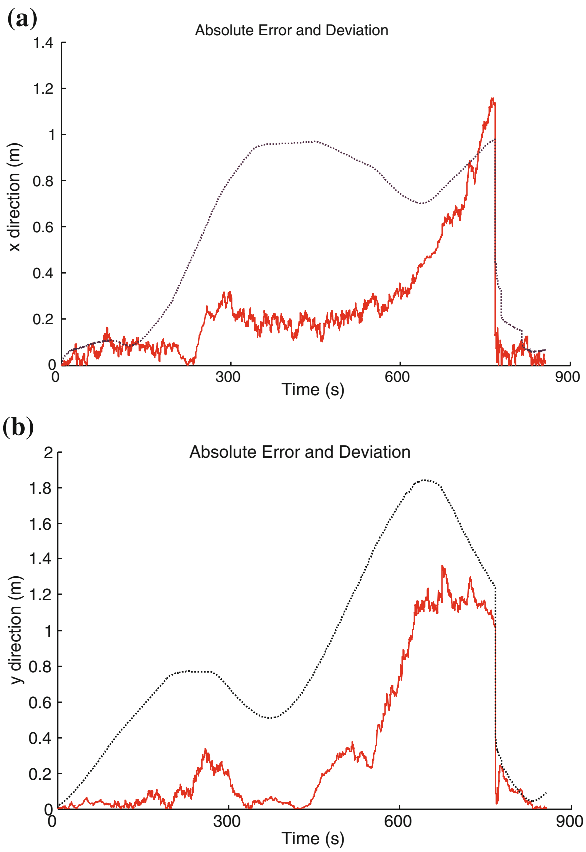


Fig. 14.4 Deviation along **a** x axis, **b** y axis

Fig. 14.5 Landmarks deviation (x = 10, y = 0) and (x = 20, y = 0) using 3,000 particles

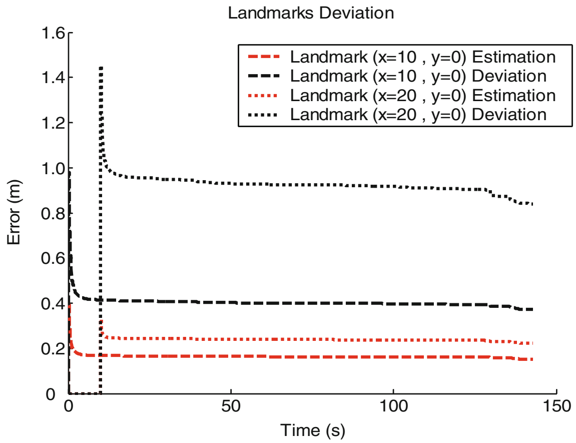
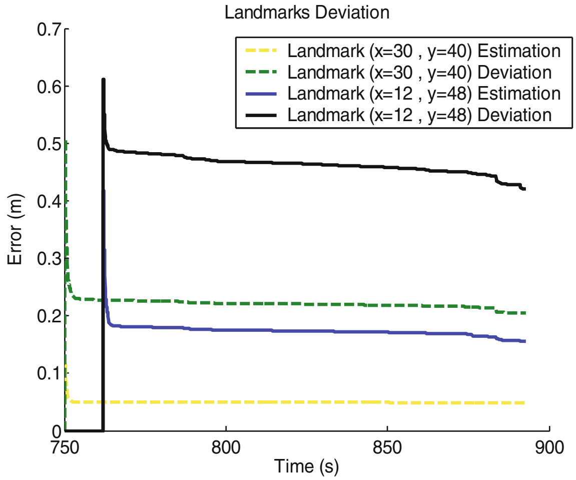


Fig. 14.6 Landmarks deviation (x = 30, y = 40) and (x = 12, y = 48) using 3,000 particles



demonstrate the performance of Unscented HybridSLAM for the scenario depicted in Fig. 14.1. In Fig. 14.5 the location estimation error of landmark (x = 10, y = 0) is approximately 0.2 m. In Fig. 14.6 the error of location estimation of landmark (x = 30, y = 40) is approximately 0.25 m.

14.4.2 Double Loop Closing Scenario

In this section, simulation results of a double loop scenario using Unscented HybridSLAM algorithm are presented. Here, the double loop closing case is exemplified in order to analyze the performance of the algorithm while the robot is travelling across

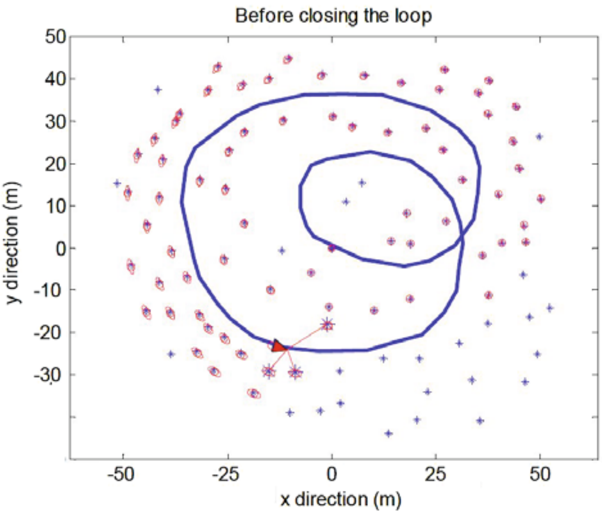
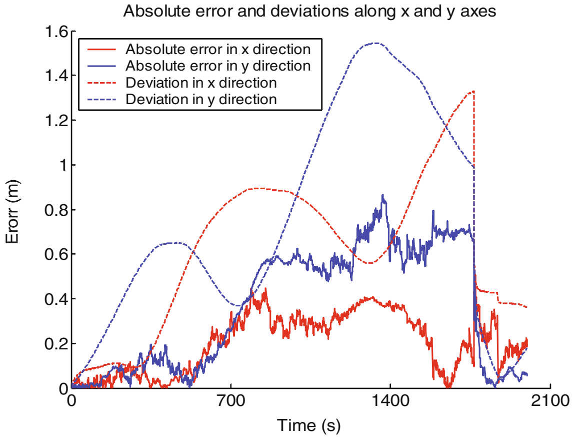


Fig. 14.7 True map of the environment with 94 observable landmarks

Fig. 14.8 Absolute error and deviations



more complex terrain. Figure 14.7 shows a map of the environment that contains an uneven distribution of landmarks.The figure also shows the true path of the robot. The speed of the mobile robot is assumed 3.5 m/s. The robot completes the whole loop inapproximately 2,800 s. Number of particles used in this experiment is 500. In Fig. 14.7 the true map of the environment and observation results before closing the loop are depicted. The vehicle starts at the centre of the test area (x = 0, y = 0) and travels counter clock wise. During the navigation process landmarks are observed and the uncertainty increases slightly. The uncertainty in the observations is at the largest value on the third part of path. Figure 14.8 demonstrates the actual error and standard deviations of the process when the robot is at the third part of the path. Simulation

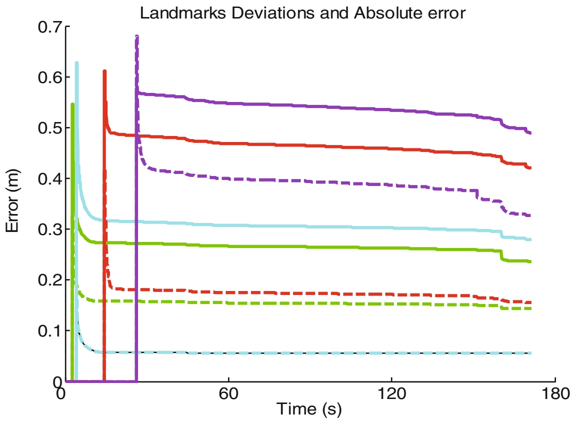


Fig. 14.9 Landmark deviation and absolute error (a double loop case) using 500 particles

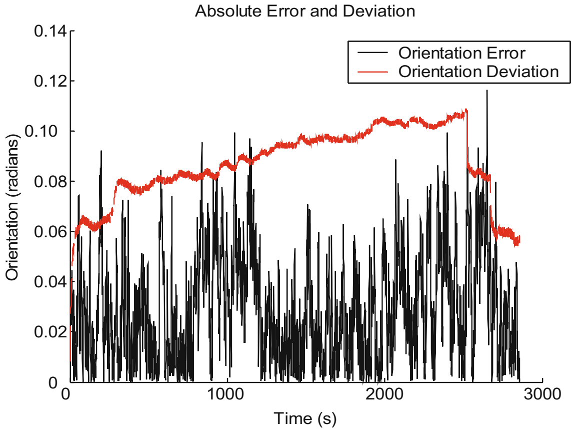


Fig. 14.10 Orientation absolute error and deviation (double loop case) using 500 particles

results illustrate the actual location error along x and y axes respectively. Dashed lines represent the 1-sigma estimated uncertainty. The simulated result indicates that UHS is a consistent method with the actual error.

Figure 14.9 shows the evolution of the uncertainty for 4 out of 6 landmarks located in the smaller loop at the beginning of the process. All solid lines represent the deviations and dashed lines represent the location estimation error. Comparing the error between actual landmarks positions and those estimated with the 2-sigma deviations indicate that the UHS algorithm is consistent, specifically with respect to landmarks location error. As expected, the actual landmarks error and uncertainty have been reduced. Two out of six landmarks were not observed due to the scanner range limitations. Figure 14.10 shows the result in regard to the orientation deviation and absolute error right after the loop is closed and indicates that the map becomes more correlated at the end of the first run. Figure 14.11 depicts the situation in which the loop is closed and the robot is at one third of the path again. The

Fig. 14.11 After the completion of the loop

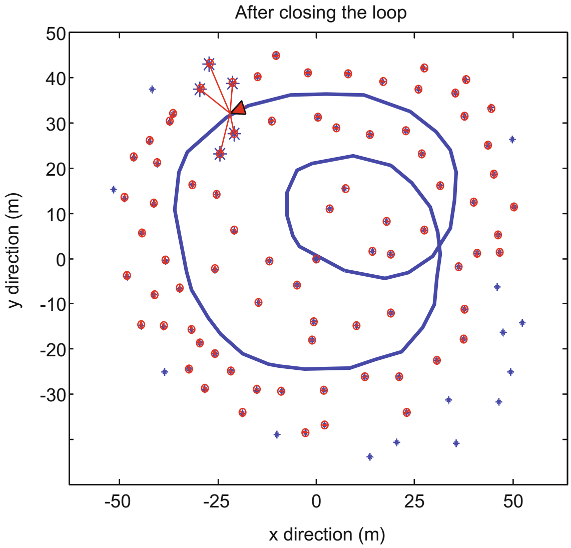
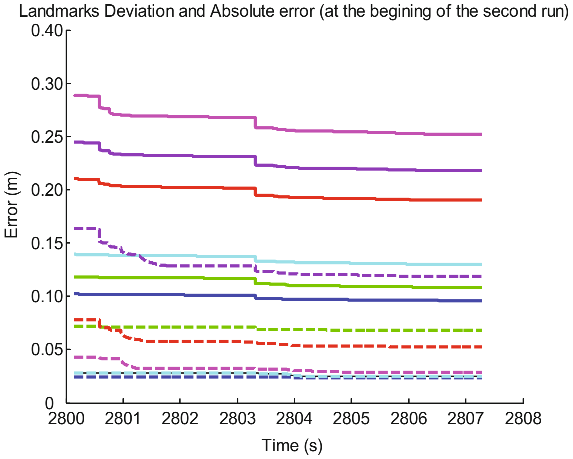


Fig. 14.12 Landmark deviation after closing the loop



robot is at point (x = −20, y = 34) and heading to complete the second loop.The uncertainty in the observation of landmark sat this point is considerably reduced, meaning that the outcome of loop closing is successful and the filter converges. Moreover, all observable landmarks have been estimated correctly following the completion of the first run. Figure 14.12 demonstrates absolute error and deviations along x and y axes, the orientation, and for six landmarks inside the internal loop after the robot completes the loop and is at one third of its path during completion of the second loop. The evolution of the uncertainty for all six landmarks in the map indicate that the map correlation in maintained and leads the final map to be consistent. These results show that the estimated uncertainty is consistent with the actual error along both axes and the orientation of the vehicle. The orientation error is around 0.02 radians which confirms the consistency of Unscented HybridSLAM algorithm.

14.5 Conclusions

The major shortcoming of most simultaneous localization and mapping algorithms is their limitation to the first order accuracy of propagated the mean and covariance as a result of first order truncated Taylor series linearization technique. Unscented HybridSLAM can address this issue with the use of a deterministic sampling approach to approximate the optimal gain and prediction terms in a linear Bayesian form. Unscented HybridSLAM, with its derivative-free Gaussian random variable propagation technique, is able to calculate the posterior mean and covariance of the system to the second order of Taylor series. In order to show how the model robot dynamics can be approximated, a derivative-free technique based on Sterling's polynomial interpolation formula was derived and presented in this paper. Derived equations were linearized due to the high non-linearity of the system. The second order Sterling Polynomial Interpolationwas employed to approximate a non-linear function with first and second order central divided difference operators acting on the observation function expressed in a non-linear form. Simulation results indicated that with the second order Sterling polynomial linearization, Unscented HybridSLAM gained enough accuracy and stability in performance for double-loop scenarios in a non-domestic environment.

References

1. Sasiadek, J.Z., Monjazeb, A., Necsulescu, D.: Navigation of an autonomous mobile robot using EKF-SLAM and FastSLAM. In: Proceedings of 16th Mediterranean Conference on Control and Automation, pp. 517–522. Ajaccio, France (2008).
2. Thrun, S., Montemerlo, M., Koller, D., Wegbreit, B., Nieto, J., Nebot, E.: FastSLAM: an efficient solution to the simultaneous localization and mapping problem with unknown data association. J. Mach. Learn. Res. (2004).
3. Norgard, M., Poulsen, N., Ravn, O.: New development in state estimation for nonlinear systems. J. Autom. **3611**, 1627–1638 (2000)
4. Dahlquist, G., Bjorck, A.: Numerical Methods. Prentice-Hall, NJ, Englewood Cliffs (1974)
5. Julier, S.J., Uhlmann, J.K.: Unscented filtering and nonlinear estimation. Proc. IEEE J. **2**(3), 401–422 (2004)
6. Smith, R., Self, M., Cheesman, P.: Estimating uncertain spatial relationships in robotics. In: Autonomous Robot Vehicles; Coxand&Wilforn Ed., pp. 167–193. Springer, Heidelberg (1974).
7. Monjazeb, A., Sasiadek, J. Z., Necsulescu, D.: Autonomous navigation of an outdoor mobile robot in a cluttered environment using a combination of unscented Kalman filter and a Rao-Blackwellised particle filter. In: Proceedings of 9th International Conference on Informatics in Control Automation and Robotics (ICINCO), pp. 485–488. Rome, Italy, July 2012.

8. Julier, S.J., Uhlmann, J.K.: A counter example to the theory of simultaneous localization and map building. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 4238–4243. March 2001.
9. Brooks, A., Bailey, T.: HybridSLAM: combining FastSLAM and EKF-SLAM for reliable mapping. Springer Tracts Adv. Robot. **57**, 647–661 (2009)
10. Monjazeb, A., Sasiadek, J.Z., Necsulescu, D.: Autonomous navigation among large number of nearby landmarks using FastSLAM and EKF-SLAM; a comparative study. In: Proceedings of 16th International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 369–374. Miedzyzdroje, Poland (2012).
11. Norgard, M., Poulsen, N., Ravn, O.: Advances in derivative-free state estimation for nonlinear systems, Tech. Rep. IMM-REP, pp. 1998–2015, Deptartment of Modelling, Technical University of Denmark (2000).
12. Williams, S.B., Dissanayake, G., Durrant-Whyte, H.: An efficient approach to the simultaneous localisation and mapping problem, In: Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Washington DC, (2002).
13. Monjazeb, A., Sasiadek, J.Z., Necsulescu, D.: An approach to autonomous navigation based on Unscented HybridSLAM. In: Proceedings of 17^{th} International Conference on Methods and Models in Automation and Robotics, pp. 369–374. Miedzyzdroje, Poland (2012).

Chapter 15
A New Relational Geometric Feature for Human Action Recognition

M. Vinagre, J. Aranda and A. Casals

Abstract Pose-based features have demonstrated to outperform low-levelappearance features in human action recognition. New RGB-D cameras provide locations of human joints with which geometric correspondences can be easily calculated. In this article, a new geometric correspondence between joints called Trisarea feature is presented. It is defined as the area of the triangle formed by three joints. Relevant triangles describing human pose are identified and it is shown how the variation over time of the selected Trisarea features constitutes a descriptor of human action. Experimental results show a comparison with other methods and demonstrate how this Trisarea-based representation can be applied to human action recognition.

Keywords Pose-based feature · Action descriptor · Action recognition

15.1 Introduction

In recent years, the study of computational methods that allow identifying and understanding human actions has been a field of interest in research and industry. The interest in this topic is motivated by its potential application in a large variety of systems, such as surveillance, patient monitoring, robotics, games, intelligent user interfaces, and in general, those activities that involve some kind of interaction between users and systems. This research is commonly known as gesture, motion or activity recognition, and several surveys have been published related to this topic in the last years. In [13] a detailed overview of current advances in the field is pro-

M. Vinagre (✉) · J. Aranda · A. Casals
Robotics Group, Institute for Bioengineering of Catalonia, Baldiri Reixac 10-12, 08028 Barcelona, Spain
e-mail: mvinagre@ibecbarcelona.eu

J. Aranda · A. Casals
Universitat Politècnica de Catalunya, BarcelonaTech, Jordi Girona 1-3, 08034 Barcelona, Spain
e-mail: joan.aranda@upc.edu

A. Casals
e-mail: alicia.casals@upc.edu

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_15

vided. The work in [1] presents recognition methodologies developed for simple human actions, as well as, for more complex high-level activities. The study presents an approach-based taxonomy that compares the advantages and limitations of each approach.

Despite the efforts of a large number of researchers, action recognition still remains an unsolved problem due to the variability of input data in intra-classes and similarity in inter-classes. In the real world a given action can be performed by subjects anthropomorphically different and, a given subject can perform a determined action with different absolute parameters of velocity or trajectory, affecting its appearance. This fact difficult the extraction of discriminant patterns that describes a particular action and the resulting loss of accuracy in the classification and recognition process.

In general, human action recognition approaches exploit appearance and/or human body parts information by defining suitable features of an image sequence. From appearance features, some methods recognize actions as a sequence of local low-level features in images such as optical flow, gabor filter or Harris corners [4, 9]. Other methods use body parts information from human pose estimation to extract features of human actions [3, 5]. A recent work [22] discusses about both approximations for action recognition in home-monitoring scenarios, depicting that pose-based features outperform low-level appearance features, even when data are heavily corrupted by noise. The study also suggests that a combined approach of both techniques can be beneficial for action recognition.

Different pose-based features extracted from positions of human joints have been used, which can be mainly classified into two methodologies. The first relies on obtaining features from joint parameters as orientation, position, velocity or acceleration. Many previous works use this kind of features to represent human poses and action [5, 21]. However, the problem of extracting a reliable similarity measure between the same type of actions or poses from these individual properties of joints is still unresolved. In the second methodology this problem is reduced obtaining features from correspondences between joints, so called relational geometric features. Features are geometric relations between joints, as the Euclidean distance between two joints or the distance between a joint to a line spanned by other two joints [23].

The good prospects of relational geometric features as pose and motion representation [2] and the short number of such features proposed up to now, has motivated this work. Thus, this research attempts to contribute with a new relational geometric feature and its use for action recognition. We propose a relational geometric feature called Trisarea, which describes the geometric correspondence between joints by means of the area of the triangle that they define. We demonstrate how the variation of a set of Trisarea features on an action sequence contains useful information for its application in human action recognition.

The rest of the paper is organized as follows. Section 15.2 gives a short review of the recent advances in pose-based features for their use in action recognition. Section 15.3 presents a new pose-based feature called Trisarea, which is extracted from a geometric relation between three joints. In Sect. 15.4, an action representation based on the variation of a set of Trisarea features over time is presented. In Sect.

15.5, a setup of action recognition integrating our action representation is presented. Experimental results about with this setup are given in Sect. 15.6 and conclusion is presented in Sect. 15.7.

15.2 Related Work

Human action recognition from pose-based features requires an inherent procedure for extracting human pose. Vision-based pose estimation faces the difficult problem of estimating kinematic parameters of a body model, either from static frame or a frame sequence. However, despite this complex initial processing, this approach has several advantages over action recognition from appearance based features since it is invariant to the point of view and to appearance variations produced by environment conditions. It is also less sensitive to noise from intra-class variances in contrast to recognition from appearance based features.

Previous promising methods for interactive human pose estimation and tracking are those that use a volumetric model of the body as [8, 10] or utilize depth information extracted from structured light sensors, as the newly Microsoft Kinect camera or the Asus Wavi Xtion as [15, 16].

Motivated by the current progress in real-time pose estimation, some recent works in human action/motion recognition have been performed based on such information. The work in [14] presents a real-time dance gestures classification system from pose representation. It uses a cascaded correlation-based classifier for multivariate time-series data and distance metric based on dynamic time-warping. This approach has an average accuracy of 96.9 % for approximately 4 s motion recordings. In [11] a method is introduced for real-time gesture recognition from a noisy skeleton stream extracted from Kinect depth sensors. They identify key poses through a multi-class classifier derived from support vector learning machines and gestures are labelled on-the-fly from key pose sequences through a decision forest tree.

These and other works as [5, 17, 18, 21] recognize human action from direct measures of joint parameters of the human body as angles, instantaneous position, orientation, velocity, acceleration, etc. Such approaches have as inconvenient that different repetitions of a same action must be numerically similar, and, due to the irregularity in the periodicity of human actions and intra-person motion variability this assumption is not always true.

Other methods accept more flexibility by using relational geometric features, describing correspondences between joints in a single pose or a short sequence of poses. In [12] different relational geometric features are introduced, which have been used for single human action recognition [20, 22] and for two-people interaction activities detection [23], with good results. In [2] different of these type of features are proposed, as:

- Distance feature. It is defined as the Euclidean distance between all pairs of joints of a human pose, at time t.

- Rotation feature. It is the rotational angle of the line spanned by two joints with respect to the reference pose.
- Line-Joint feature. It is defined as the Euclidean distance between a joint and a line spanned by other two joints.
- Plane feature. It computes the correspondence between the plane spanned by some joints with respect to a single joint, as the distance from this joint to the referred plane.
- Normal plane feature. Similar to plane feature, but here the plane is defined by its normal spanned by two joints and a joint belonging to the plane.
- Angle feature. It is the angle between two lines spanned by two pairs of different joints.

Features based on geometric relations between joints are easy to calculate and they have demonstrated to be discriminative for pose classification problems [2]. This result highlights the potential effectiveness of these features for action recognition. However, little work has been done on the creation of new relational geometric features, as well as their use as a basis for action recognition processes.

15.3 Trisarea Feature

In this work, a particular pose of human body is seen as a complete graph of joints where edges are the distances between them. With the aim to characterize this pose with the spatial relationship of a group of selected joints, a new feature called Trisarea is proposed. Trisarea is defined as the area of the triangle formed by three joints and it encodes in just one number the geometric relation between them. Although we know Trisarea does not encode an unequivocal spatial relation between joints (different triangles can produce the same area), a join of specific Trisarea features seems to do it as it will be demonstrated in this work.

Mathematically, let p_1, p_2, p_3 be the coordinates of joints j_1, j_2, j_3 in a Euclidean space Given a pose P, the Trisarea feature between j_1, j_2, j_3 joints is defined by:

$$\triangle\,(j_1, j_2, j_3, P) = \frac{1}{2} \cdot \|\overrightarrow{p_1 p_2} \times \overrightarrow{p_1 p_3}\| \tag{15.1}$$

Figure 15.1 shows an example of Triarea features in a given pose where there are three geometric relations between eight joints.

15.4 Action Representation

In the previous section, Trisarea feature has been presented as a geometric relation which can be calculated from joints of a human pose. In this section, a represen-

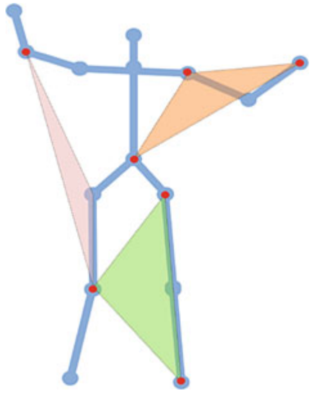


Fig. 15.1 Example of Trisarea features

tation of a human action is built from the evolution of different Trisarea features along human poses in the action sequence. The existence of some irrelevant Trisarea features was realized by observation, thus, an automatic method to filter and select the most important features is applied. Finally, the action representation as a single feature vector encoding the variation of selected Trisarea features is presented.

15.4.1 Extracting Information About Actions

In order to extract information of an action from pose changes of human body, the variation of Trisarea features from the sequence of poses in an action is calculated. The number of possible Trisarea features depends on the number of joints in the pose representation. Being J the number of joints of a pose representation, the number of possible Trisarea features F from a pose is:

$$F = \frac{J!}{3! \cdot (J-3)!} \tag{15.2}$$

Thus, given an action Act with a sequence of poses P_{seq} the action information $Info(Act)$ is a matrix of $F \times |P_{seq}|$ defined as:

$$Info(Act) = \begin{bmatrix} \triangle_1 (j_i, j_j, j_k, P_0) & \cdots & \triangle_1 (j_i, j_j, j_k, P_{|P_{seq}|}) \\ \vdots & \ddots & \vdots \\ \triangle_F (j_r, j_p, j_q, P_0) & \cdots & \triangle_F (j_r, j_p, j_q, P_{|P_{seq}|}) \end{bmatrix} \tag{15.3}$$

Figure 15.2 shows the action information of an *arm wave* action. In this example, the pose representation contains $J = 15$ joints as shown in Fig. 15.1. So, the number of Trisarea features by applying the Eq. 15.2 is $F = 455$.

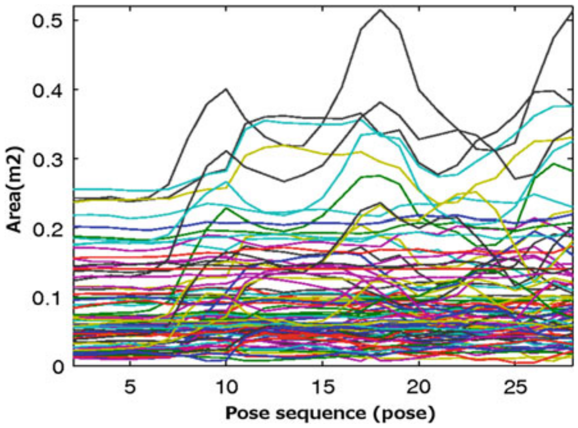


Fig. 15.2 Information about a '*arm wave*' action represented by the variation of all Trisarea features

As shown, Trisarea features contribute to encode useful information about action. However, many of these features do not provide any information and can be obviated without loss of discrimination performance.

The selection of relevant features is not immediately intuitive, but few of them are clearly irrelevant, as those define invariant areas, formed by mutually constrained joints (i.e. the relation beween torso, right shoulder and left shoulder).

In order to eliminate non informative Trisarea features, we perform an unsupervised feature selection procedure, as shown in next section.

15.4.2 Feature Selection

The feature selection process is a common preprocessing filter step used for classification and pattern recognition applications. In this process, the goal is to detect the most informative features as well as to reduce computational cost and avoid undesired noise discarting non-informative ones.

Hence, we have used a computationally feasible unsupervised feature selection algorithm called Principal Feature Analysis (PFA) [7]. From the initial representation with a feature vector with all possible Trisarea features, this method exploits the information that can be inferred from a reduced principal component space in order to obtain the optimal subset of salient features.

This feature selection methodology differs from common feature extraction methods as principal component analysis (PCA), independent component analysis (ICA) and Fisher Linear Discriminate Analysis (LDA). These methods apply a mapping from the original feature space to a lower dimensional feature space, having as disadvantage that all components of the original feature are needed in the projection to the lower dimensional space, so they must be always calculated. Instead, in *PFA* only a subset of relevant components in the original feature is selected, thus lessening

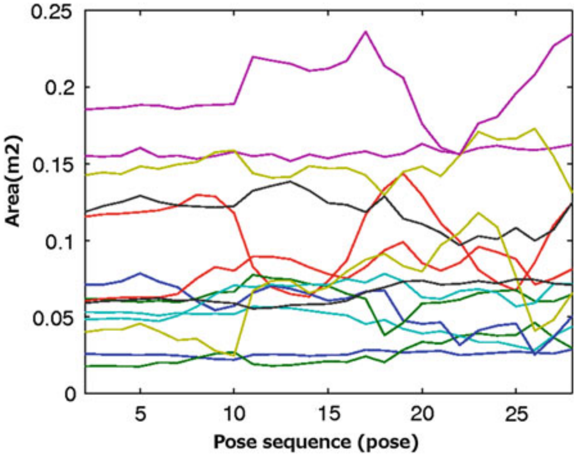


Fig. 15.3 Most important information about '*arm wave*' example

computation time. In this case there is no mapping process and it is possible to work directly in a reduced feature space. Detailed information of this method can be read in [7].

PFA method is applied on a large input set of Trisarea-based feature vectors of poses selected at random overall available poses from actions to be recognized. Here, the amount of information to be retained from the input set is selected in order to extract most important features. As a result, we obtain a reduced feature vector representing poses with the most informative Trisarea features (with cardinality F_{filt}, where $F >> F_{filt}$). This feature vector is used to obtain most relevant information about action $(Info_{filt})$.

As an example, Fig. 15.3 shows the remaining features ($F_{filt} = 14$) as a result of the *PFA* process that retains the 95 % of data input information from the original set of features($F = 455$) of the *arm wave* action information obtained in Fig. 15.2.

15.4.3 Action Representation as Trisarea Feature Variations

Our main hypothesis in this work is that the variation of Trisarea features is useful to represent an action. In order to define an action feature as the variation of Trisarea features, a variation parameter from Trisarea feature evolution along action sequence is extracted. For that, we use the descriptive statistic parameter called Pearson's variation coefficient. This coefficient summarizes variations of a Trisarea feature ($\delta_{\triangle_i}$) over the sequence of poses in the action , calculated as:

$$D_{var}(\delta_{\triangle_i}) = \frac{\sigma(\delta_{\triangle_i})}{\mu(\delta_{\triangle_i})} \tag{15.4}$$

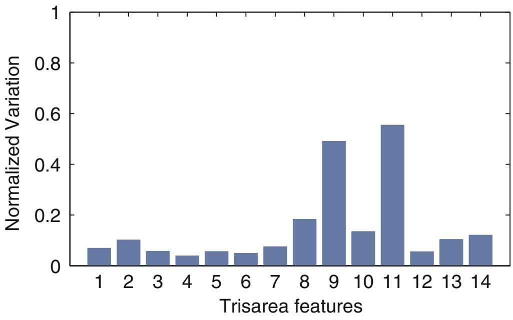


Fig. 15.4 Action representation of '*arm wave*' example

where $\sigma(\cdot)$ and $\mu(\cdot)$ perform the standard deviation and the mean of Trisarea feature values along the sequence of poses. Finally, the Eq. 15.4 is applied to every selected features. As a result, a single vector with dimension $1 \times F_{filt}$ is calculated:

$$\varphi = \langle D_{var}(\delta_{\triangle_1}), \ldots, D_{var}(\delta_{\triangle_{F_{filt}}}) \rangle \tag{15.5}$$

As an example, Fig. 15.4 shows the instance of the motion *wave arm* example shown in Sect. 15.4.2. This action is instantiated with the 14 Trisarea features selected by the *PFA* pre-process.

15.5 Human Action Recognition

The core of an action recognition process is exploring input data in order to identify it. In this work, the meaningful features about action are represented in the action representation explained in Sect. 15.4.3. Since the action representation is defined as a feature vector, classification methods using machine learning techniques can be used. The performance of the majority of these techniques depends on the feature space and the quality of data used in learning. In fact, our action representation is not useful with techniques which deal with the temporal order of data patterns because this data has been reduced.

In order to evaluate the contribution of Trisarea features as a feature space in a recognition process, two classification techniques were performed. The first classification technique was a Nearest Centroid classifier. This classification is a supervised neighbors-based learning method that obtains a prototype class by the centroid of its training instances. Let T_{set}^{ζ} be the set of training instances of a certain class ζ used in the learning phase, the action prototype or centroid of a class is calculated as:

$$Prot_{\zeta} = Mean(T_{set}^{\zeta}) \tag{15.6}$$

In order to determine the class of an unknown input action, the minimal similarity against all action prototypes using an Euclidean distance measurement is calculated. Let φ an unknown action instance, which must be classified, and let K be the set of action types (*classes*). The resulting action class ζ of φ is given by:

$$\zeta = argmin_{\zeta}(\|\varphi - Prot_{\zeta}\|)_{\forall \zeta \in K} \tag{15.7}$$

The second classification method used was the Naïve Bayes classifier. This classifier is a supervised learning method based on applying Bayes' theorem with the *assumption* of independence between every pair of Trisarea features. It requires a small amount of training data to estimate necessary parameters. In order to classify an unknown action instance φ from K types of actions, the maximum a posteriori (MAP) decision rule is applied. The estimation of the action class ζ of φ is calculated as:

$$\zeta = \text{argmax}_{\zeta}\ P(\zeta|\varphi)_{\forall \zeta \in K} \tag{15.8}$$

In this work, the likelihood function of the features given for each class was modeled as Gaussian mixtures.

15.6 Experimentation

In this section, a test of the action recognition approaches explained in Sect. 15.5 is presented. For this test, a public dataset is used in order to compare the obtained results with other recognition methods in the literature. Finally, the obtained results are discussed.

15.6.1 Experimentation Setup

Dataset. We selected the public MSR Action 3D dataset [6] which supplies the sequences of depth maps captured by a depth camera similar to a Kinect device, with a frame rate of 15 fps and down-sampled resolution of 320 × 240. This dataset contains 20 different actions that cover various movement of arms, legs, torso and their combinations without human-object interactions: high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hands wave, side -boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing and pick up and throw. Each action was performed by 9–10 subjects two or three times. The subjects were advised to use their right arm or leg whenever an action is to be performed by a single limb. Altogether, 567 actions sequences were used, those provided by the dataset.

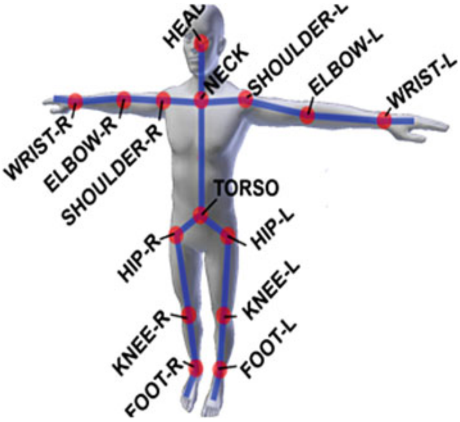


Fig. 15.5 Pose model

Pose Model. The pose estimation was extracted from the original depth maps. After that, pose estimation results were inspected manually to filter possible pose estimation errors.

Our human pose model was a pose representation of 15 joints, as shown in Fig. 15.5. As some approaches explained in Sect. 15.2, we normalized the position of joints relative to the torso position to make the pose description independent to pose changes with respect to the world space and camera changes (e.g. point of view changes) and tolerant to anthropomorphic differences.

15.6.2 Results

In order to contrast our analysis, the dataset actions were divided into three subsets, suggested in [6]. This division has been followed in different works in order to obtain a public benchmarking. Concretely, every of subsets contain 8 actions, as shown in Table 15.1. The action sets AS1 and AS2 were intended to group actions with similar movements, while the action set AS3 was intended to group complex actions together.

For each experiment/subset, actions were represented as the Trisarea evolution representation explained in Sect. 15.4.1. After that, the filtering step called Principal Feature analysis (PFA) and explained in Sect. 15.4.2 was applied. For the *PFA*, a random set of poses from available actions to be recognized was calculated. Concretely, this set was built with a set of 1,500 randomized poses spread evenly from actions.

In the *PFA* process, the number of features to be filtered was calculated taking into account the ratio of the amount of original input information to be retained (*retained variation ratio*). The behaviour of the retained information against the number of selected features is shown in Fig. 15.6. The inflection points of each experiment were around 90 %. So, we chose 90 % as the retained variation to preserve a good

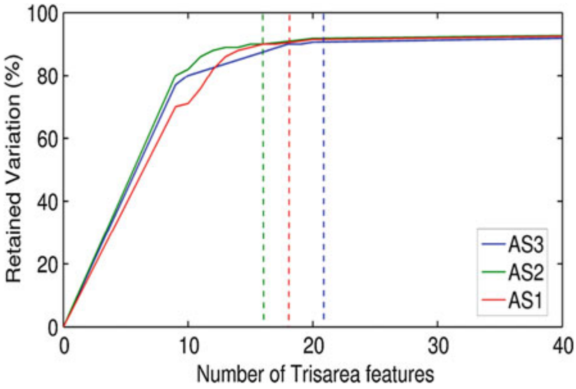


Fig. 15.6 Retained variation against number of selected Trisarea features

information-feature ratio. The number of selected Trisarea features was 18, 16 and 21 for AS1, AS2 and AS3 respectively.

With the results of the filtering step, the selected Trisarea features were used to deal action data into a feature space. The data transformation was performed by applying Eq. 15.5 over data of subsets. The new data representation of actions in a single feature vector allow us to perform the action recognition process based on the two classification methods introduced in Sect. 15.5. Since these classification methods have a training step in a supervised learning way, 3/4 parts of action instances were selected as training data. The rest of data were used for the classification test.

The accuracy of both classification approaches for each experiment/subset are shown in Table 15.2.

Table 15.1 Data subsets from MSR Action 3D dataset

Subset AS1	Subset AS2	Subset AS3
Horiz. arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw x	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend	Two hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup and throw	Side boxing	Pickup and throw

Table 15.2 Results of the classification performance

	AS1 (%)	AS2 (%)	AS3 (%)
Nearest centroid	75.4	73.5	79.6
Naive bayes	88.6	86.3	94.0

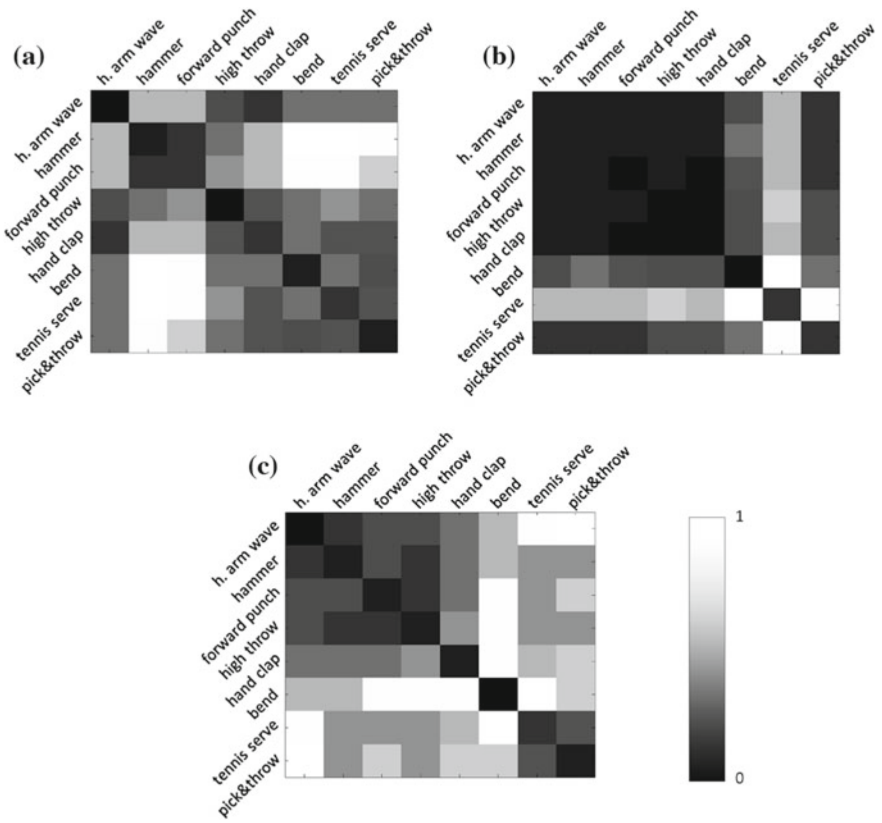


Fig. 15.7 Normalized sum of distances between all tested instances belonging to a class against the class prototypes in **a** AS1, **b** AS2 and **c** AS3 action sets

We can observe a better accuracy of the Naïve Bayes (NB) classifier than the Nearest Centroid (NC) technique. One of the reason is that the NC classifier performed was a non-generative model and it did not take into account the variability of the distances to the centroid within a class. Nevertheless, the Naïve Bayes classifier is a generative model where classes are modeled by probability distributions which are generated by training data. This probabilistic framework accommodates asymmetric misclassification and class priors.

In order to analyze the NC classifier performance, similarity between test samples and learned action prototypes has been calculated. Figure 15.7 shows the normalized mean distance between test samples belonging to the same class against the action prototypes. Low values was expected along the diagonal indicating high similarity of the samples with the prototype of their class. Higher out of diagonal values indicate low similarity to other prototypes so lower probability of misclassification.

The dispersion of the test samples of the same class were not significant (*with stdev around 0.2*). This fact depicts good results of action representation space with the variation of Trisarea features. Figure 15.7 shows that some actions like *high arm wave* and *hammer*, similar actions present similar distance between their test samples and their prototypes. Indeed, in Fig. 15.7b we can observe that half of the actions are very similar and it remarks the poor classification accuracy in AS2. These results show that NC classifier decreases its accuracy when similar actions exist.

In the case of high amount of training data and non-similar real actions, the NC classifier with our action instances can be applied. This approach constitutes a fast recognition task with moderate recognition accuracy. For better accuracy results, the Naïve Bayes classifier can be applied. The result of NB classifier provides an average of 89.6 % while NC classifier reaches an average of 76 %.

In general, both classifiers results are good and they depict that our Trisarea feature seems to be useful feature for human motion recognition. Relevant features retain sufficient information to describe and discriminate different human actions. Since NB shows a better performance than NC classifier, NB classifier was selected as our action recognition method and a specific comparison against others approaches in the literature is presented in the next section.

15.6.3 Comparison

We compared our action recognition approach with three different state of the art methods [6, 11, 19]. These approaches uses different recognition strategies and they use the same dataset as benchmark. So, we can perform a testbed in order to compare and evaluate our action recognition performance. In [6] a method to recognize human actions from sequences of depth maps is presented. They obtain a projection of representative sampled 3D points to characterize a set of salient postures which are used as nodes to describe an action graph. In [19] depth maps images are used too. They present Space-Time Occupancy Patterns (STOP) which depth information sequence is represented in a 4D space-time grid and an action graph based system is used to learn a statistical model for each action class. On the other hand, [11] presents a real-time action gesture recognition from a pose stream with an angular representation. They capture key poses through a multi-class classifier and a gesture/motion is labelled from a key pose sequence through a decision forest algorithm

For the testbed comparison, the partition of dataset were performed in the same way that the others approaches. This partition consisted in performing a training set with half of the samples and the rest of samples for the test part. Accuracies of our approach and others with datasets AS1, AS2 and AS3 are shown in Table 15.3.

In general, the results in Table 15.3 depict how the dynamics of the Trisarea feature is able to give useful information in terms of human action recognition. The results of the feature selection of Trisarea features show a reduction from 455 possible features to a maximum of 21, retaining around 90 % of the original input information. This fact confirms that only few Trisarea features are relevant to the recognition processes. Statistics on selected features was performed and this analysis

Table 15.3 Comparison of recognition accuracies (%)

	Li et al. (%)	Vieira et al. (%)	Miranda et al. (%)	Our (%)
AS1	72.9	84.7	93.5	76.2
AS2	71.9	81.3	52.0	72.3
AS3	79.2	88.4	95.4	81.0
Avg.	74.7	84.8	80.3	76.5

shows that relevant triangles are, in most cases, formed by a swing of non-adjacent joints with unconstrained movements in 3D space (i.e. elbows, knees or wrist) with respect to a joint with a constrained position from torso reference (i.e. shoulder, hip or neck).

The comparison with other recognition approaches denotes that our approach performs reasonably well. For this testbed, our results outperform [6] approach. On the other hand, [11, 19] have better results because they have a more accurate temporal information about motion. However, our approach have the most compact and presumably better than previous pose-based methods for an online action recognition process.

As conclusion, the results show the usefulness of Trisarea features extraction to perform an action feature space that permits us the use of classical machine learning algorithms to perform human action recognition.

15.7 Conclusion

The extraction of human pose features based on a new geometric relation between joints, called Trisarea feature, has been presented. A low dimensional feature vector of human pose formed by a set of Trisarea features that holds the most useful information for a posterior action recognition process has been obtained. For the selection of the set of Trisarea features, the Principal Feature Analysis filter has been applied.

An action representation based on the variation of selected Trisarea features overall poses in the action sequence has been proposed. The continuous variation of each Trisarea feature has been described as its Pearson's relative coefficient of variation, obtaining a single value per feature and a compact vector representing an action (about 20 scalar values).

In order to verify our action representation accuracy in an action recognition procedure, several action representation instances have been generated from three different datasets. As a result, the experiments have demonstrated the usefulness of Trisarea features for human action recognition tasks.

A comparison with other approaches in the same scenario has revealed that our recognition results have not been far from other methodologies results. In addition, the presented approach has got a good accuracy/speed ratio because it has less preprocessing calculations than the compared pose-based approaches.

Although we have obtained promising results with the Pearsons coefficient of variation, this measure is ambiguous for similar movements and we have to explore other options to encode the dynamic behavior of Trisarea features in motion sequences.

Acknowledgments This work has been done under project IPRES, DPI2011-29660-C04-01 of the Spanish National Research Program, with partial *FEDER* funds. It has also been partially founded by *Fundació La Caixa* inside the *Recercaixa 2012* research program.

References

1. Aggarwal, J., Ryoo, M.: Human activity analysis: A review. ACM Comput. Surv. **43**(3), 16:1–16:43 (2011)
2. Chen, C., Zhuang, Y., Xiao, J., Liang, Z.: Perceptual 3D pose distance estimation by boosting relational geometric features. Comput. Animation Virtual Worlds **20**(2–3), 267277 (2009)
3. Ellis, C., Masood, S.Z., Tappen, M.F., Laviola Jr, J.J., Sukthankar, R.: Exploring the trade-off between accuracy and observational latency in action recognition. Int. J. Comput. Vis. **101**(3), 420–436 (2013)
4. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. Trans. Pattern Anal. Mach. Intell. **29**(12), 2247–2253 (2007)
5. Gu, J., Ding, X., Wang, S., Wu, Y.: Action and gait recognition from recovered 3-d human joints. Trans. Sys. Man Cyber. Part B **40**(4), 1021–1033 (2010)
6. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 9–14. IEEE Computer Society, Washington, DC, USA (2010)
7. Lu, Y., Cohen, I., Zhou, X. S., Tian, Q.: Feature selection using principal feature analysis. In: Proceedings of the 15th International Conference on Multimedia, pp. 301–304. ACM, New York, NY, USA (2007)
8. Luo, X., Berendsen, B., Tan, R.T., Veltkamp, R.C.: Human pose estimation for multiple persons based on volume reconstruction. In: Proceedings of the 2010 20th International Conference on Pattern Recognition, pp. 3591–3594. IEEE Computer Society, Washington, DC, USA (2010)
9. Matikainen, P., Hebert, M., Sukthankar, R.: Representing pairwise spatial and temporal relations for action recognition. In: Proceedings of the 11th European Conference on Computer Vision: Part I, pp. 508–521. Springer, Berlin, Heidelberg (2010)
10. Straka, M., Hauswiesner, S., Rüther, M., Bischof, H.: Skeletal graph based human pose estimation in real-time. In: Proceedings of the British Machine Vision Conference, pp. 69.1–69.12. BMVA Press, Aberystwyth, Wales
11. Miranda, L., Vieira, T., Morera, D.M., Lewiner, T., Vieira, A.W., Campos, M.F.M.: Real-time gesture recognition from depth data through key poses learning and decision forests. In: SIBGRAPI, pp. 268–275. IEEE Computer Society, Washington, DC, USA (2012)
12. Müller, M., Baak, A., Seidel, H.-P.: Efficient and robust annotation of motion capture data. In: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 17–26. ACM, New York, NY, USA (2009)
13. Poppe, R.: A survey on vision-based human action recognition. Image Vis. Comput. **28**(6), 976–990 (2010)
14. Raptis, M., Kirovski, D., Hoppe, H.: Real-time classification of dance gestures from skeleton animation. In: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 147–156. ACM, New York, NY, USA (2011)
15. Schwarz, L., Mateus, D., Castaneda, V., Navab, N.: Manifold learning for tof-based human body tracking and activity recognition. In: Proceedings of the British Machine Vision Conference, pp. 80.1–80.11. BMVA Press, Aberystwyth, Wales (2010)

16. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1297–1304. IEEE Computer Society, Washington, DC, USA (2011)
17. Sung, J., Ponce, C., Selman, B., Saxena, A.: Unstructured human activity detection from rgbd images. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 842–849. IEEE, Washington, DC, USA (2012)
18. Uddin, M.Z., Thang, N.D., Kim, J.T., Kim, T.-S.: Human activity recognition using body joint-angle features and hidden markov model. ETRI J. **33**(4), 569–579 (2011)
19. Vieira, A.W., Nascimento, E.R., Oliveira, G.L., Liu, Z., Campos, M.F.M.: Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. In: Proceedings of the 17th Iberoamerican Congress, pp. 252–259. Springer, Berlin, Heidelberg (2012)
20. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1290–1297. IEEE Computer Society, Washington, DC, USA (2012)
21. Xia, L., Chen, C.-C., Aggarwal, J.K.: View invariant human action recognition using histograms of 3d joints. In: CVPR Workshops, pp. 20–27. IEEE, Washington, DC, USA (2012)
22. Yao, A., Gall, J., Fanelli, G., Van Gool, L.: Does human action recognition benefit from pose estimation? In: Proceedings of the British Machine Vision Conference, pp. 67.1-67.11. BMVA Press, Aberystwyth, Wales (2011)
23. Yun, K., Honorio, J., Chattopadhyay, D., Berg, T., Samaras, D.: Two-person interaction detection using body-pose features and multiple instance learning. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 28–35. IEEE (2012)

Chapter 16
Geometry Estimation of Urban Street Canyons Using Stereo Vision from Egocentric View

Tobias Schwarze and Martin Lauer

Abstract We investigate the problem of estimating the local geometric scene structure of urban street canyons captured from an egocentric viewpoint with a small-baseline stereo camera setup. We model the facades of buildings as planar surfaces and estimate their parameters based on a dense disparity map as only input. After demonstrating the importance of considering the stereo reconstruction uncertainties, we present two approaches to solve this model-fitting problem. The first approach is based on robust planar segmentation using random sampling, the second approach transforms the disparity into an elevation map from which the main building orientations can be obtained. We evaluate both approaches on a set of challenging inner city scenes and show how visual odometry can be incorporated to keep track of the estimated geometry in real-time.

Keywords Environment perception · Geometry estimation · Robust plane fitting

16.1 Introduction

Robotic systems aiming at autonomously navigating public spaces need to be able to understand their surrounding environment. Many approaches towards visual scene understanding have been made, covering different aspects such as object detection, semantic labeling or scene recognition. Also the extraction of geometric knowledge has been recognised as important high-level cue to support scene interpretation from a more holistic viewpoint. Recent work for instance demonstrates the applicability in top–down reasoning [1, 2].

Extracting geometric knowledge appears as hard task especially in populated outdoor scenarios, because it requires to tell big amounts of unstructured clutter apart

T. Schwarze (✉) · M. Lauer
Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany
e-mail: tobias.schwarze@kit.edu

M. Lauer
e-mail: martin.lauer@kit.edu

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_16

from the basic elements that make up the geometry. This problem can be approached from many sides, clearly depending on the input data. In recent years the extraction of geometric knowledge from single images has attracted a lot of attention and has been approached in different ways, e.g. as recognition problem [3], as joint optimization problem [4], or by geometric reasoning e.g. on line segments [5]. Other than the extensive work in this field, here we investigate the problem based on range data acquired from a stereo camera setup as only input, which is in principle replaceable by any range sensor like LIDAR systems or TOF cameras. We aim at extracting a local geometric description from urban street scenarios with building facades to the left and right ("street canyon"). Rather than trying to explain the environment as accurately as possible, our focus is a simplified and thus very compact representation that highlights the coarse scene geometry and provides a starting point for subsequent reasoning steps. To this end our goal is a representation based on geometric planes, in the given street canyon scenario one plane for each building facade, which are vertical aligned to the groundplane.

Such representation can basically be found in two ways. The 3D input data can be segmented by growing regions using similarity and local consistency criteria between adjacent data points that lead to planar surface patches, or surfaces can be expressed as parametric models and directly fitted into the data. Either way has attracted much attention. Studies on range image segmentation have been conducted, but usually evaluating range data that differs strongly from outdoor range data obtained by a stereo camera in terms of size of planar patches and level of accuracy [6]. Variants of region growing can be found in e.g. [7, 8].

The combination of short-baseline stereo, large distances in urban scenarios and difficult light conditions due to a free moving and unconstrained platform poses challenging conditions. Additionally we can not assume free view on the walls since especially traffic participants and static infrastructure often occlude large parts of the images—a key requirement is hence robustness of the fitting methods.

Region growing alone does not guarantee to result in connected surfaces when occlusions visually split the data, a subsequent merging step would be necessary. This does not occur when parametric models are fitted directly. Most popular methods here are random sampling and 3D Hough transformations [9].

A large body of literature focuses specifically on the task of groundplane estimation, in case of vision systems planes have been extracted using v-disparity representations [10] and robust fitting methods [11], often assuming fixed sensor orientation [12].

We start with estimating the groundplane using random sampling. Based on the groundplane parameters we constrain the search space to fit two planes to the left and right building facade. In Sects. 16.2.2 and 16.2.3 we present two robust methods to fulfil this task. In Sect. 16.3 we evaluate both methods using a dataset of inner city scenes and show how visual odometry data can be integrated to keep track of the estimated geometry.

16.2 Plane Fitting

Estimating planar structures from an egocentric viewpoint in urban environments has to deal with a huge amount of occlusions. Especially the groundplane is often only visible in a very small part of the image since buildings, cars or even pedestrians normally occlude free view onto the ground. Hence, robustness of the methods is a key requirement. Therefore, we developed an approach based on the RANSAC scheme [13], which is known to produces convincing results on model fitting problems even with way more than 50 % outliers. In a scenario with fairly free view and cameras pointing towards the horizon with little tilt a good heuristic is to constrain the search space to the lower half of the camera image space to find an initial estimate of the groundplane.

A plane described through the equation

$$aX + bY + cZ + d = 0$$

can be estimated using the RANSAC scheme by repeatedly selecting 3 random points and evaluating the support of the plane fitting these points. A fit is evaluated by counting the 3D points with point-to-plane distance less than a certain threshold. In our case, we had to extend the RANSAC scheme by an adaptive threshold to cope with the varying inaccuracy of 3D points determined from a stereo camera. To account for the uncertainty, the covariance matrices of the XYZ points can be incorporated into the distance threshold. In case of reconstructing from stereo vision one obtains the 3D coordinates (X, Y, Z)[1] through:

$$F(u, v, \delta) = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{B(u-c_x)}{\delta} \\ \frac{B(v-c_y)}{\delta} \\ \frac{Bf}{\delta} \end{bmatrix} \tag{16.1}$$

where B is the baseline, f the focal length, δ the disparity measurement at image point (u, v), and (c_x, c_y) the principal point. The covariance matrix C can be calculated by (also see [14]) $C = J \cdot M \cdot J^T$ with J the Jacobian of F

$$J = \begin{bmatrix} \frac{dF_X}{du} & \frac{dF_X}{dv} & \frac{dF_X}{d\delta} \\ \frac{dF_Y}{du} & \frac{dF_Y}{dv} & \frac{dF_Y}{d\delta} \\ \frac{dF_Z}{du} & \frac{dF_Z}{dv} & \frac{dF_Z}{d\delta} \end{bmatrix} = \begin{bmatrix} \frac{B}{\delta} & 0 & \frac{-B(u-c_x)}{\delta^2} \\ 0 & \frac{B}{\delta} & \frac{-B(v-c_y)}{\delta^2} \\ 0 & 0 & \frac{-Bf}{\delta^2} \end{bmatrix}$$

Assuming a measurement standard deviation of 1px for the u and v coordinates and a disparity matching error of 0.05 px we obtain as measurement matrix $M = diag(1, 1, 0.05)$. A world point on the optical axis of the camera in 15 m distance is subject to a standard deviation of ~1 m (640 × 480 px, focal length 5 mm, baseline

[1] Our Z-axis equals the optical axis of the camera, X-axis pointing right and Y-axis towards the ground. Compare Fig. 16.1.

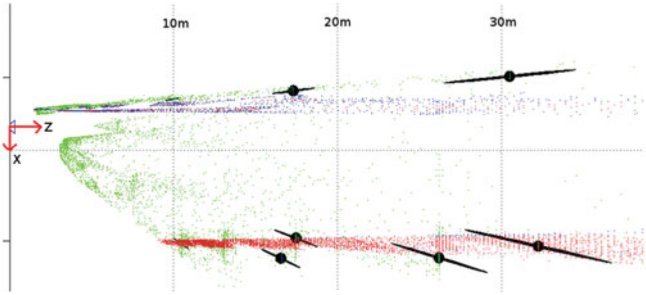


Fig. 16.1 Stereo covariances

12 cm). While the Z uncertainty of reconstructed points grows quadratically with increasing distance, the uncertainty of reconstructed X and Y components remains reasonable small (see Fig. 16.1).

With the covariance matrices we can determine the point to plane Mahalanobis distance and use it instead of a fixed distance threshold to count plane support points. This way the plane margin grows with increasing camera distance according to the uncertainty of the reconstruction. Calculating the point to plane Mahalanobis distance essentially means transforming the covariance uncertainty ellipses into spheres. A way to do so is shown in [15].

However, calculating the covariance matrix for every XYZ-point is computationally expensive. We can avoid this by fitting planes in the $uv\delta$-space and transforming the plane parameters into XYZ-space afterwards.

Fitting planes in $uv\delta$-space can be done in the same way as described above for the XYZ-space. We obtain the plane model satisfying the equation

$$\alpha u + \beta v + \gamma - \delta = 0$$

Expressing u, v, δ through Eq. (16.1) yields the $uv\delta$ to XYZ-plane transformation

$$a = \alpha; \quad b = \beta; \quad c = \frac{\alpha c_x + \beta c_y + \gamma}{f}; \quad d = -B$$

and vice-versa

$$\alpha = \frac{aB}{d}; \quad \beta = \frac{bB}{d}; \quad \gamma = \frac{B(cf - ac_x - bc_y)}{d}$$

In the following section we demonstrate the importance of considering the reconstruction uncertainty when setting the plane distance threshold.

16.2.1 Sweep Planes

For the purpose of estimating facades of houses we construct a plane vertical to the groundplane and sweep it through the XYZ-respectively $uv\delta$-points. For most urban scenarios it is a valid assumption that man-made structures and even many weak and cluttered structures like forest edges or fences are strongly aligned vertical to the floor. Knowledge of the groundplane parameters $\{\boldsymbol{n}_{gp}, d_{gp}\}$ with groundplane normal $\boldsymbol{n_{gp}}$ and distance d_{gp} from the camera origin allows us to construct arbitrary planes perpendicular to the ground.

We construct a sweep plane vector perpendicular to $\boldsymbol{n_{gp}}$ and the Z-axis (compare Fig. 16.1) by

$$\boldsymbol{n}_{sweep}(\alpha) = R_{\boldsymbol{n}_{gp}}(\alpha)\left(\boldsymbol{n}_{gp} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right),$$

where rotation matrix $R_{\boldsymbol{n}_{gp}}(\alpha)$ rotates α degrees around the axis given by $\boldsymbol{n}_{gp}$.

We sweep the plane $\{\boldsymbol{n}_{sweep}(\alpha), d_{sweep}\}$ through the XYZ respectively $uv\delta$-points by uniformly sampling α and d_{sweep} and store the number of support points for every sample plane in the parameter space (α, d_{sweep}). The result for a sampling between $-10° < \alpha < 10°$ and -3 m $< d_{sweep} <$ 15 m with a stepsize of 1° resp. 0.5 m is shown in Fig. 16.2. Peaks in the parameter space correspond to good plane support.

A fixed threshold in XYZ-space (Fig. 16.2a) overrates planes close to the camera (left facade), while planes further away are underrated (right wall). The Mahalanobis threshold (Fig. 16.2b) can compensate for this. Fitting planes in $uv\delta$-space leads to the same result (Fig.16.2c), without the computational expense.

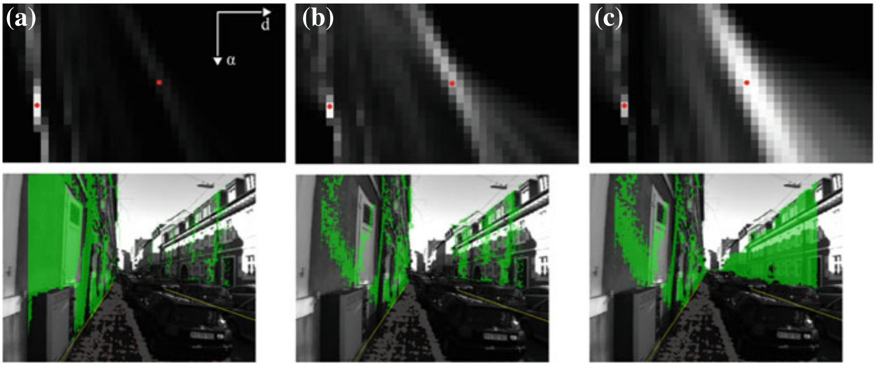


Fig. 16.2 Plane estimation based on point-to-plane distance thresholds using **a** a fixed distance in XYZ coordinates, **b** the Mahalanobis distance in XYZ coordinates and **c** a fixed distance in $uv\delta$ coordinates. The *top row* visualizes with gray values the number of support points for a vertical plane swept through the XYZ- resp. $uv\delta$-points in distance and angle steps of 0.5 m resp. 1°. The true plane parameters are marked in *red*. The *bottom row* shows support points for these plane parameters as overlay

The method can be used to extract planar surfaces from the points, regardless whether input data is present in XYZ- or $uv\delta$-space. It is easy to incorporate prior scene knowlegde like geometric cues to select planes and suppress non-maxima. Parallel planes are mapped to the same rows in parameter space, while a minimal distance between selected planes can be enforced by the column gap.

In practice, when knowledge about the rough orientation of the scene is unavailable, the computational cost of building the parameter space is very high. In our strongly constrained example scene with ±10° heading angle and 18 m distance already 777 planes had to be evaluated. The required size and subsampling of parameter space is hard to anticipate and would have to be chosen much bigger, which makes the approach less attractive in this simple form.

Due to the fact that the plane sweeping is not a data-driven approach, many planes are evaluated that are far off the real plane parameters. Simplifying the data does not affect the number of evaluated planes. In its data-driven form this algorithm resembles the Hough transform [16], which has also been studied and extended for model-fitting problems in 3D data, e.g. [17].

In the following sections we present two data-driven approaches for wall estimation. First, we show how RANSAC can be used to achieve a planar scene segmentation, from which we extract the street geometry. In Sect. 16.2.3 we use a 2D Hough transform to find the left and right facade.

16.2.2 Planar RANSAC Segmentation

RANSAC based plane fitting can obviously not only be used to fit the groundplane but any other planar structure in the scene. Assuming the groundplane parameters are known, we first remove the groundplane points from $uv\delta$-space. In the remaining points we iteratively estimate the best plane using RANSAC. Since we are interested in vertical structures, we can reject plane hypotheses that intersect the groundplane at smaller angles than 70° by comparing their normal vectors. After every iteration we remove the plane support points from $uv\delta$-space. This way we generate five unique plane hypotheses, out of which we select the two most parallel planes ($\alpha = \frac{180}{\pi}\arccos(\boldsymbol{n}_1 \cdot \boldsymbol{n}_2)$) with distance $d = |\boldsymbol{n}_1 d_1 + \boldsymbol{n}_2 d_2| > 5$m by pairwise comparison to obtain the left and right building facade. Figure 16.3 shows an example scene with five plane hypotheses (left) out of which the red and blue one are selected since they are the most parallel and also exceed the minimal distance threshold.

The top row in Fig. 16.4 presents the output for some challenging scenarios, some of which feature considerable occlusions. In every iteration we evaluate 50 planes with a valid groundplane intersection angle, 5 iterations hence summing to 250 evaluated plane hypotheses.

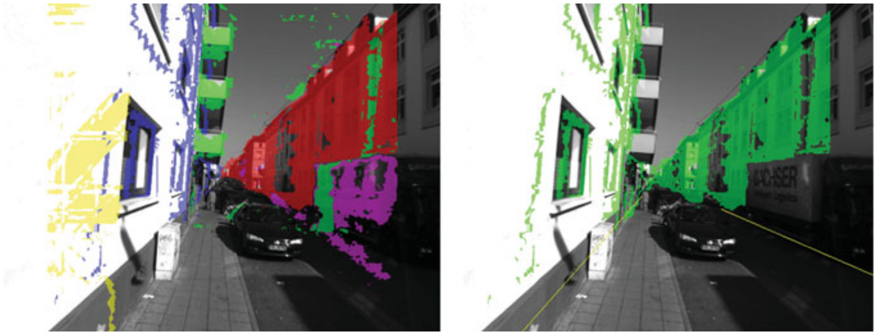


Fig. 16.3 RANSAC based planar segmentation. The two most parallel planes (*right*) are selected from five hypotheses (*left*)

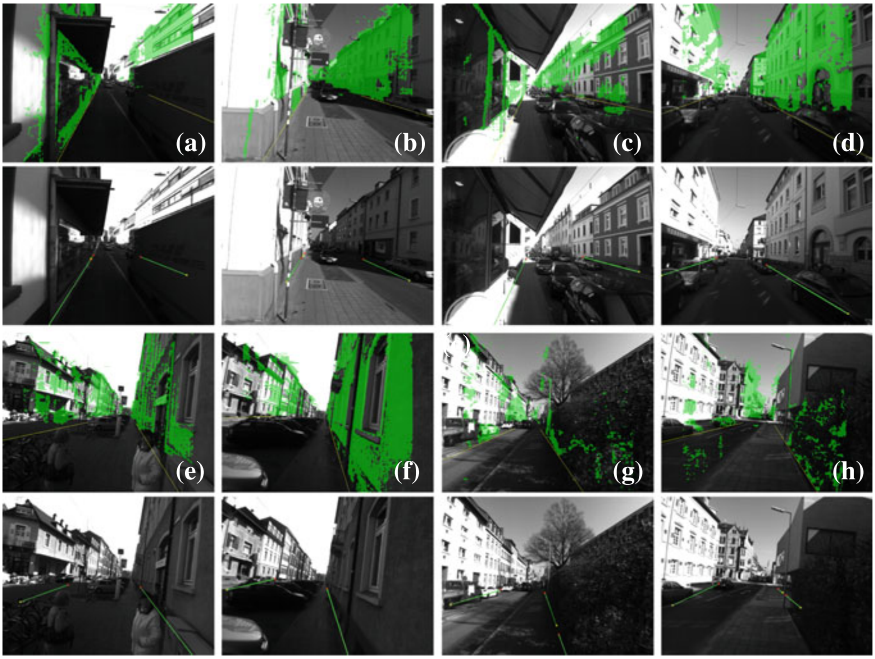


Fig. 16.4 Results of RANSAC based planar segmentation (*top row*) and estimation of facade orientations using elevation maps (*bottom row*), which can be used in a subsequent step to generate an according surface

16.2.3 Elevation Maps

The strong vertical alignment of man-made environments can be exploited by transforming the 3D point data into an elevation map. We do this by discretizing the groundplane into small cells (e.g. 10 × 10 cm) and projecting the 3D points along the

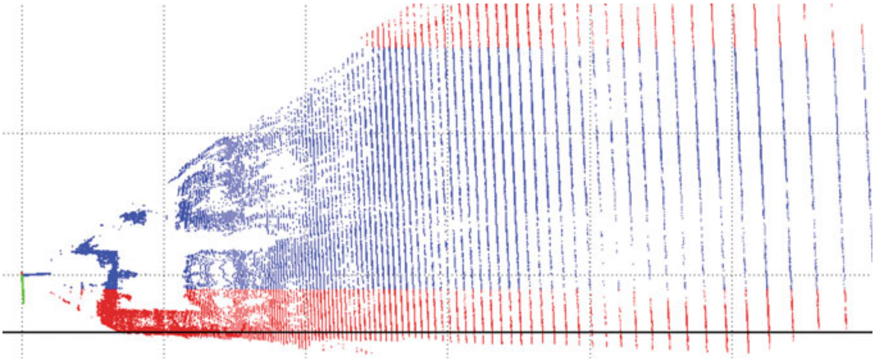


Fig. 16.5 Lateral view onto the 3D points of Fig. 16.6 (*top row*) with groundplane estimate. Blue points are considered for elevation map, red points are cropped

groundplane normal onto this grid. The number of 3D points projecting onto a cell provides a hint about the elevation over ground for the cell. Grid cells underneath high vertical structures will count more points than grid cells underneath free-space.

To make the approach stable with respect to objects and clutter in groundplane proximity we crop 3D points that are closer than 1.5 m and further than 10 m from the estimated groundplane. The wall structures we are interested in generally fill this range. A lateral view is shown in Fig. 16.5. Two exemplary elevation maps are depicted in Fig. 16.6.

The resulting elevation map can be analysed using 2D image processing tools. In case of a street scenario with expected walls left and right we apply a Hough transform to discover two long connected building facades. Because of geometric plausibility we again enforce a minimal distance of 5 m between walls when selecting the Hough peaks.

The approach also works with weak structures like forest edges (e.g. bottom image of Fig. 16.6), though overhanging trees are obviously causing a deviation from the real forest ground line here and the assumed street model with two walls does not hold in this view. The approach will further benefit from integrating elevation maps over multiple frames.

16.2.4 Iterative Least-Squares

The random nature of the RANSAC segmentation and the assumption of perfectly vertical buildings in case of the elevation maps prevent either approach to produce perfectly robust results. Nevertheless, both approaches normally yield an approximation of the main orientation of the buildings, which is accurate enough to optimize the estimated surface with a least-squares estimator. To deal with the remaining outliers we optimize the plane hypotheses iteratively. While shrinking the plane to

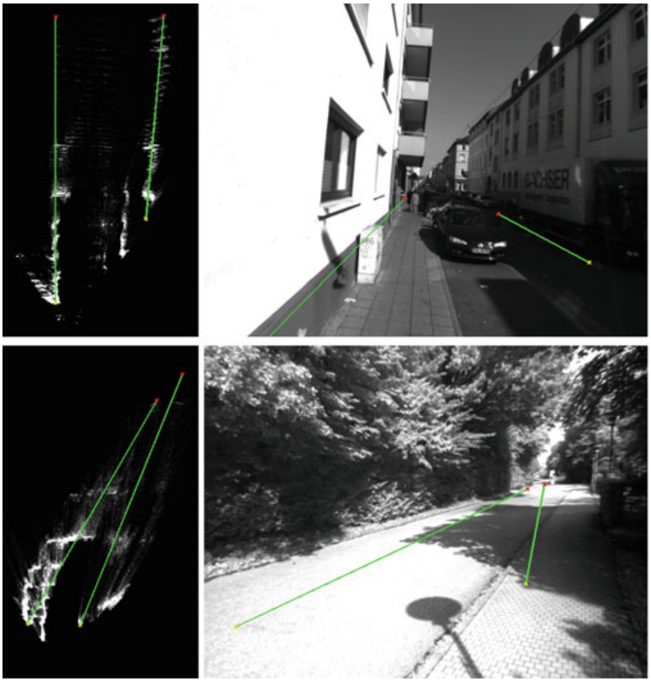


Fig. 16.6 *Left column* Elevation maps. Connected elements are found by Hough transformation and projected into the camera image (*right column*)

point distance threshold in every iteration, the optimization converges within a few iterations.

We verify the plane by comparing the normal angle deviation between the initial fit and the optimized fit. A false initial fit will lead to big deviations and can be rejected in this way.

16.3 Experiments

In a set of experiments we compare the different approaches for plane estimation. Our experimental setup consists of a calibrated stereo rig with a short baseline of around 10 cm and a video resolution of 640 × 480 px. To enlarge the field-of-view we deploy wide-angle lenses of 5 mm focal length. We obtain the dense disparity estimation using an off-the-shelf semi-global-matching approach (OpenCV).

16.3.1 Evaluation

We ran some tests on a dataset consisting of inner city scenes captured from ego-view to evaluate the applicability of the proposed approaches in some challenging scenarios. Processing video data from ego-view perspective especially has to be robust against occlusion and the high degree of clutter caused by parked cars or bikes, trees, or other dynamic traffic participants. Another issue in real life scenarios are the challenging light conditions, that often lead to over- and underexposed image parts in the same image. Figure 16.4 shows 8 scenes with the result of RANSAC segmentation in the top row, and the resulting facade orientation drawn from elevation maps in the bottom row.

16.3.1.1 Planar RANSAC Segmentation.

The random plane selection in the RANSAC segmentation approach makes it difficult to draw quantitative conclusions about the performance. To underlay some numbers we picked five of the more difficult scenarios and evaluated the repeatability of the output. We ran the algorithm 50 times on each scenario and evaluated the number of correct wall estimations by manual supervision. We consider a wall as missed when the estimated orientation deviates so strongly, that a subsequent optimization step will not converge close to the optimum.

Figure 16.7 shows the results for some scenarios taken from Fig. 16.4. Scenario (c) is challenging in that the left building wall is hardly visible due to occlusion, and the visible part is overexposed. The right wall is found very robustly. The large amount of errors in detecting the right wall in scenario (d) can be explained by clutter, which often leads to planes fitted to the sides of the parked cars. Increasing the number of sampled planes per iteration would probably prevent this. The substantial gap on the right hand side in (h) explains the often missed right wall. In scenarios with mostly free view on the walls a rate of around 90 % for both walls is realistic.

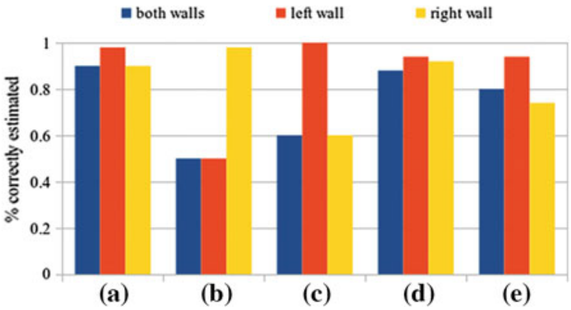


Fig. 16.7 Evaluation of repeatability of planar RANSAC segmentation. Shown is the percentage of correctly determined walls in 50 repetitions, scenes correspond to Fig. 16.4

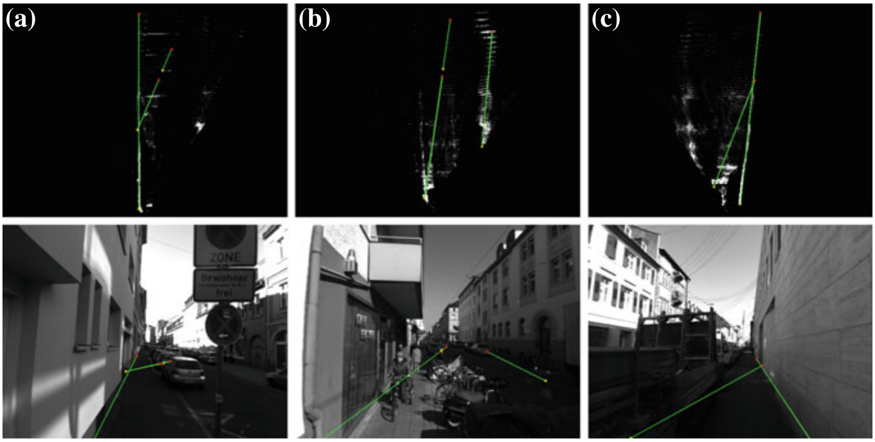


Fig. 16.8 Failures in orientation estimation using elevation maps. See text for details

16.3.1.2 Elevation Maps

To rate the stability of wall estimation based on elevation maps we investigate two sequences of 200 and 500 frames, taken while travelling down a street canyon. In each frame we estimate the orientation of both walls, independent of the previous frame.

The first sequence consists of 200 frames and is mostly free of wall occlusions. The algorithm finds the correct wall orientation in all but 8 frames, that were taken while passing a street sign (see Fig. 16.8a). The second sequence consists of 500 frames and is more cluttered. The algorithm fails in around 10 % of all frames to estimate one of the walls correctly. Reasons are always related to obstacles that were not filtered because they exceed the groundplane cut-off height, or obstacles that occlude the wall. See Fig. 16.8b, c for two examples.

16.3.2 Real-Time Geometry Tracking

The wall estimation is embedded as part of a real-time system, which also contains a module to estimate the camera motion between consecutive frames by running the visual odometry estimation taken from the LIBVISO2 library [18]. Visual odometry provides the egomotion in all six degrees of freedom such that the camera poses of frame $n-1$ and n are related via a translation vector $\boldsymbol{t}$ and a rotation matrix R.

Knowledge of the camera transformation allows to predict the current groundplane and wall parameters from the previous frame. The XYZ-plane $\boldsymbol{p}_{t-1} = \{\boldsymbol{n}, d\}$, with surface normal $\boldsymbol{n}$ and distance d from the camera origin, transforms into the current frame via

$$\boldsymbol{p}_t = \left([R|\boldsymbol{t}]^{-1}\right)^T \boldsymbol{p}_{t-1}$$

In a street canyon scenario we proceed as follows: We initialize the groundplane and planes for left and right wall with the methods described above. For the following frames we use the prediction as starting point for the iterative least-squares optimization to compensate the inaccuracy of the egomotion estimation. To stabilize the process over time we store the best fitting support points in a plane history ring buffer and incorporate them with a small weighting factor into the subsequent least-squares optimization. We reject the optimization when the plane normal angles of prediction and optimization deviate by more than 5° in either direction, or the groundplane angle becomes smaller than 80°. Reasons for this to happen are normally related to a limited view onto the wall, which either occurs when the wall is occluded by some close object (e.g. truck), or the cameras temporarily point away from the wall. If the optimization was rejected we carry over the prediction as current estimate and continue like that until the wall is in proper view again.

The estimated plane parameters for both walls in a sequence over 450 frames are plotted in Fig. 16.9. The upper diagram shows the angle between groundplane and

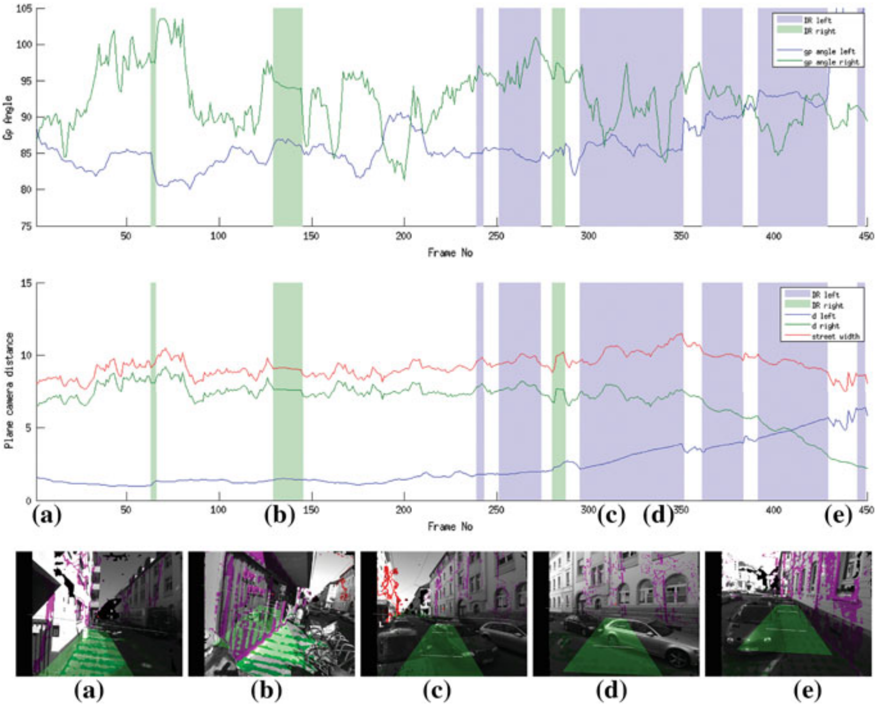


Fig. 16.9 Plane parameters tracked over a sequence of 450 frames. The *top diagram* shows the groundplane angle, the *bottom diagram* the normal plane distance. Predicted parts due to walls being out of view are *shaded*

walls, the bottom diagram shows the plane distance parameters. The distances add up to the street width, for this sequence with a mean of 9.3 m.

The sequence begins on the left sidewalk and ends on the right sidewalk after crossing the street. It contains several parts in which the walls are out of view due to the camera heading, some are shown in the screen shots. As explained earlier, these parts are bridged by predicting the parameters using the ego-motion and are shaded in the diagram.

We run the visual odometry in a thread parallel to the disparity estimation. The visual odometry is updated with 30 Hz on our platform (dual-core i7 2,7 GHz), while new disparity estimations in VGA resolution are available with around 10 Hz. The subsequent iterative least-squares optimization of the predicted geometry adds around 10 ms to the overall processing time, but can equally be run in its own thread while the next disparity map is prepared. The update rate of the visual odometry is sufficient to track a head-worn camera setup exposed to typical head motion, which consequently allows to keep an updated model of the geometry in real-time.

16.4 Conclusion

We have demonstrated two approaches towards estimating the local geometric structure in the scenario of urban street canyons. We model the right and left building walls as planar surfaces and estimate the underlying plane parameters from 3D data points obtained from a passive stereo-camera system, which is replaceable by any kind of range sensor as long as the uncertainties of reconstructed 3D points are known and can be considered.

The presented approaches are not intended as a standalone version. Their purpose is rather to separate a set of inlier points fitting the plane model to initialize optimization procedures as we applied in form of the iterative least-squares. By taking visual odometry in combination with a prediction and update step into the loop we are able to present a stable approach to keep track of groundplane and both walls.

Future work includes integrating the rich information offered by the depth-registered image intensity values and relaxing the assumptions implied by the street canyon scenario.

Acknowledgments The work was supported by the German Federal Ministry of Education and Research within the project OIWOB. The authors would like to thank the "Karlsruhe School of Optics and Photonics" for supporting this work.

References

1. Geiger, A., Lauer, M., Urtasun, R.: A generative model for 3d urban scene understanding from movable platforms. In: CVPR'11, Colorado Springs, USA (2011)
2. Cornelis, N., Leibe, B., Cornelis, K., Gool, L.V.: 3d urban scene modeling integrating recognition and reconstruction. Int. J. Comput. Vis. **78**(2–3), 121–141 (2008)
3. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. Int. J. Comput. Vis. **75**, 151–172 (2007)
4. Barinova, O., Lempitsky, V., Tretiak, E., Kohli, P.: Geometric image parsing in man-made environments. In: ECCV'10, pp. 57–70. Springer, Berlin (2010)
5. Lee, D.C., Hebert, M., Kanade, T.: Geometric reasoning for single image structure recovery. In: CVPR'09 (2009)
6. Hoover, A., Jean-baptiste, G., Jiang, X., Flynn, P.J., Bunke, H., Goldgof, D.B., Bowyer, K., Eggert, D.W., Fitzgibbon, A., Fisher, R.B.: An experimental comparison of range image segmentation algorithms. Pattern. Anal. Mach. Intell. IEEE. Trans. On. **18**(7), 673–689 (1996)
7. Gutmann, J.S., Fukuchi, M., Fujita, M.: 3D perception and environment map generation for humanoid robot navigation. Int. J. Robot. Res. **27**(10), 1117–1134 (2008)
8. Poppinga, J., Vaskevicius, N., Birk, A., Pathak, K.: Fast plane detection and polygonalization in noisy 3D range images. In: IROS'08 (2008)
9. Iocchi, L., Konolige, K., Bajracharya, M.: Visually realistic mapping of a planar environment with stereo. ISER **271**, 521–532 (2000)
10. Labayrade, R., Aubert, D., Tarel, J.P.: Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In: Intelligent Vehicle Symposium, Vol. 2, pp. 646–651 (2002)
11. Se, S., Brady, M.: Ground plane estimation, error analysis and applications. Robot. Autonom. Syst. **39**(2), 59–71 (2002)
12. Chumerin, N., Van Hulle, M.M.: Ground plane estimation based on dense stereo disparity, pp. 209–213. In: ICNNAI'08, Minsk, Belarus (2008)
13. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
14. Murray, D.R., Little, J.J.: Environment modeling with stereo vision. In: IROS'04 (2004)
15. Schindler, K., Bischof, H.: On robust regression in photogrammetric point clouds. In: Michaelis, B., Krell, G., (Eds.) DAGM-Symposium, Vol. 2781 of Lecture Notes in Computer Science, pp. 172–178. Springer, Berlin (2003)
16. Hough, P.: Method and means for recognizing complex patterns. U.S. Patent 3.069.654 (1962)
17. Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A.: The 3d hough transform for plane detection in point clouds: A review and a new accumulator design. 3D Res. **2**(2), 32:1–32:13 (2011)
18. Geiger, A., Ziegler, J., Stiller, C.: Stereoscan: dense 3d reconstruction in real-time. In: IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany (2011)

Chapter 17
Hand Motion Detection for Observation-Based Assistance in a Cooperation by Multiple Robots

Toyomi Fujita and Tetsuya Endo

Abstract In a cooperation task by multiple robots, it may happen that a working robot can not detect a target object for handling due to a sensor occlusion. In this situation, if another cooperative robot observes the working robot with the target object and detects their positions and orientations, it will be possible for the working robot to complete the handling task. Such behavior is a kind of indirect cooperation. This study proposes a method for such an indirect cooperation based on an observation by the partner robot. The observing robot obtains corresponding points of Scale-Invariant Feature Transformation (SIFT) on the working robot with hand and the target object from multiple captured images. The 3-D position of the target object and hand motion of the working robot can be detected by applying stereo vision theory to the points. The working robot is then able to get the relation between its hand and the target object indirectly from the observing robot. We describe each process to establish the indirect cooperation. Fundamental experiments confirmed the validity of presented method.

Keywords Robot vision · Cooperation by observation · Scale-Invariant Feature Transformation (SIFT) · Stereo vision

17.1 Introduction

Indirect cooperation is an important function in a working environment with multiple robots, especially if a robot observes another robot and assists the movement indirectly. Specifically, the authors consider a situation in which a mobile working robot that has a manipulator can not detect a target object to handle due to a sensor occlusion. Figure 17.1a shows an example of the case; the working robot can not

T. Fujita (✉) · T. Endo
Department of Electronics and Intelligent Systems,
Tohoku Institute of Technology, Sendai, Japan
e-mail: t-fujita@tohtech.ac.jp

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_17

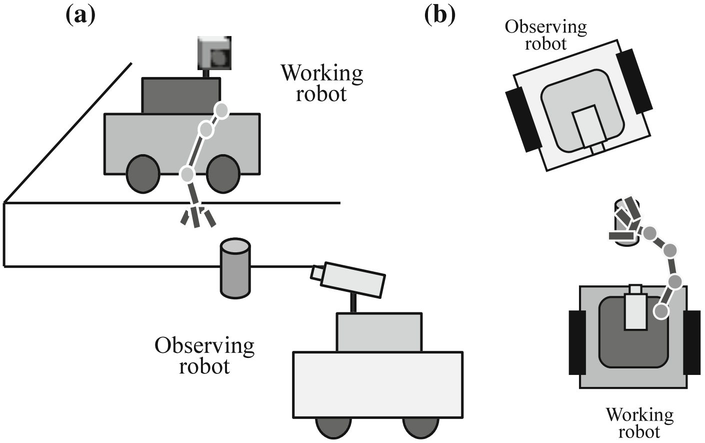


Fig. 17.1 Situations in which a working robot can not detect the target object: **a** the working robot can not detect the target object which is outside of visual angle of its camera, **b** the working robot occludes the visual field by its arm itself in manipulation.

detect the target object because it is outside of the visual angle of the camera mounted on the robot. Figure 17.1b shows another example; the working robot occludes the visual field by its arm itself due to manipulation. In these situations, if another robot that has a camera observes the working robot with the target object and detects their positions and orientations, it can assist the handling of the working robot indirectly by sending the information to the robot.

The aim of this study is to establish the functions to achieve such an indirect assistance. More specifically, this study considers how the observing robot detects the working robot with its position, the motion of its hand, and the position and orientation of the target object by vision. We propose a method in which Scale-Invariant Feature Transformation (SIFT) is applied for detecting them and computing their positions. SIFT can generate feature points which become useful correspondences for computing the 3-D position of an object from multiple view images. Therefore, the observing robot will be able to get valid 3-D information to assist the working robot by applying stereo vision theory.

In earlier studies on such an assistance, a pioneering work by [1] has presented cooperation tasks based on an observation for multiple robot. However, a kind of indirect cooperation for assistance of handling task has not been considered yet.

The following sections describe a method for assistance in indirect cooperation. Section 17.2 explains an overview of cooperation and each process for assistance including computations of SIFT features and 3-D information. Section 17.3 shows results of fundamental experiments. Finally, conclusion and future works are provided.

17.2 Observation-Based Assistance

17.2.1 Overview of Cooperation

Figure 17.2 shows an overview of cooperation with observation-based assistance for the object handling. The cooperation is made by the following procedure.

(a) At first, the observing robot finds the working robot with its hand and the target object in the surrounding environment. This study assumes that the observing robot obtains SIFT features for the working robot in advance. Therefore, the observing robot can detect regions for SIFT features matching to those for the working robot. The region that has a larger number of the matching features than a threshold can be extracted as the working robot in an observed image. The target object can be detected in the same way.

(b) After detecting the working robot with its hand and the target object, the observing robot changes the viewing location. The observing robot can obtain SIFT features for them in the same way to (a), and their correspondences. Their 3-D positions are able to be computed from the correspondences in two or more views based on stereo vision theory. The observing robot then sends the relative position information of the target object to the working robot by a communication.

(c) When the working robot receives the relative position data of the target object to itself, it starts planning the path of the hand to the target object.

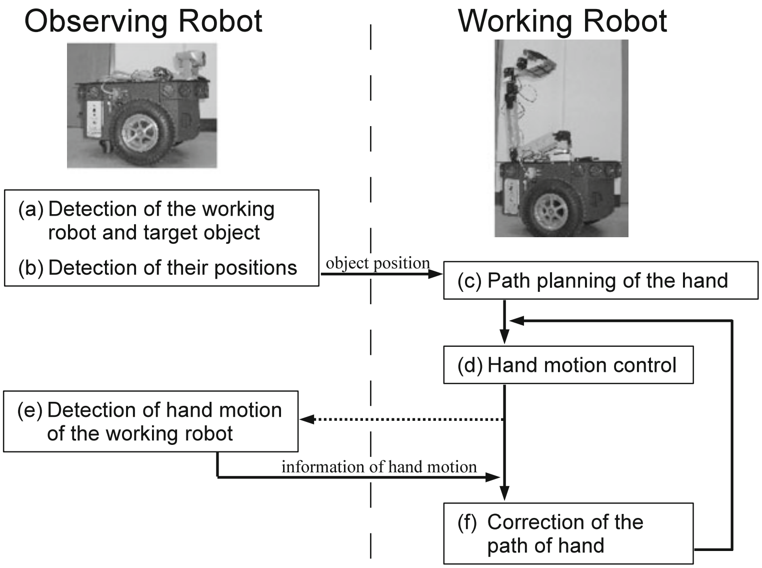

Observing Robot
Working Robot
(a) Detection of the working robot and target object
(b) Detection of their positions
object position
(c) Path planning of the hand
(d) Hand motion control
(e) Detection of hand motion of the working robot
information of hand motion
(f) Correction of the path of hand

Fig. 17.2 Procedure for the object handling based on assistance by observing robot

(d) Then the working robot can control hand motion based on the planned path. The joint angles at each time can be calculated using Jacobian which is computed from movement of the hand in a sampling cycle time.

(e) The observing robot continuously observes hand movement of the working robot. During the working robot moves the hand, the observing robot detects 3-D information of the hand motion; that is possible even though the observing robot stays at the same position. Then the hand movement information is sent to the working robot by a communication.

(f) According to the hand movement information received from the observing robot, the working robot corrects the path of hand motion if the current path has an error to reach the target object. Then it updates the control of hand motion as described in (d).

In this procedure, the working robot continuously moves its hand with the correction of the path in the loop of (d) and (f). The observing robot also keeps tracking the working robot and its hand motion to obtain more correct motion data in the process of (e). Updated data are then sent to the working robot.

The following sections describe key functions for these processes: the object and position detection for the observing robot by the use of SIFT, and the hand motion control for the working robot.

17.2.2 Detection of SIFT Features

SIFT is capable of robust detection of feature points in an image. It is also able to describe quantities of detected features to the change of scale, illumination, and rotation of image robustly. Therefore, it is useful for object detection and recognition.

The processes of the detection of SIFT features consist of *extraction of feature points*, *localization*, *computation of orientation*, and *description of quantities of features* [2]. The followings paragraphs explain these processes.

17.2.2.1 Extraction of Feature Points

In order to obtain candidate points of the feature points, we used a difference between smoothness images $L(u, v, \sigma)$ at a position (u, v) in an input image, whose scale σ is different. The difference $D(u, v, \sigma)$ between scales $k\sigma$ and σ is defined as

$$D(u, v, \sigma) = L(u, v, k\sigma) - L(u, v, \sigma). \tag{17.1}$$

$L(u, v, \sigma)$ is given by a convolution between Gaussian image $G(x, y, \sigma)$ and input image $I(u, v)$. $G(x, y, \sigma)$ at a position (x, y) in Gaussian window is defined as

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2 + y^2}{2\sigma^2}). \tag{17.2}$$

Therefore,

$$L(u, v, \sigma) = G(x, y, \sigma) * I(u, v), \tag{17.3}$$

where $*$ represents the convolution.

Three DoG images are picked from consecutive scales: $k\sigma$, σ, and $\frac{\sigma}{k}$. $D(u, v, \sigma)$ values are compared in 27 pixels: surrounding 8 pixels and 9 pixels at the same regions for the scales. If $D(u, v, \sigma)$ is extreme value in the 27 pixels, the point is picked as a candidate of the feature point. In this study, we gave 1.6 as the initial scale of σ and $k = 2^{1/3}$.

17.2.2.2 Localization

In this process, some possible feature points are picked in the candidate points by eliminating the points which may come from noise. In order to do that, principal curvature is calculated by computing two-dimensional Hessian matrix $\boldsymbol{H}$ as follows,

$$\boldsymbol{H} = \begin{pmatrix} g_{xx} & g_{xy} \\ g_{xy} & g_{yy} \end{pmatrix}, \tag{17.4}$$

where g is 2nd order derivative of D on a candidate point on each direction; for example g_{xy} is that on x and y directions.

The trace $\mathrm{Tr}(\boldsymbol{H})$ and determinant $\mathrm{Det}(\boldsymbol{H})$ are used to decide feature point. The points that meet

$$\frac{\mathrm{Tr}(\boldsymbol{H})^2}{\mathrm{Det}(\boldsymbol{H})} < \frac{(\gamma_{th} + 1)^2}{\gamma_{th}} \tag{17.5}$$

are picked as the feature points, where right equation shows threshold. γ_{th} indicates the ratio of the first eigenvalue of $\boldsymbol{H}$ to the second eigenvalue of $\boldsymbol{H}$.

17.2.2.3 Computation of Orientation

The orientation of the feature is computed in order to obtain a quantity of feature which is invariable to a rotation.

First, the gradient intensity $m(u, v)$ and gradient direction $\theta(u, v)$ are calculated by

$$m(u, v) = \sqrt{f_u(u, v)^2 + f_v(u, v)^2}, \tag{17.6}$$

and

$$\theta(u, v) = \tan^{-1} \frac{f_v(u, v)}{f_u(u, v)}, \tag{17.7}$$

where

$$\begin{cases} f_u(u, v) = L(u+1, v) - L(u-1, v) \\ f_v(u, v) = L(u, v+1) - L(u, v-1). \end{cases} \tag{17.8}$$

Next, a histogram with weights on the gradient intensity and gradient direction at surrounding points (x, y) are computed. The weight $w(x, y)$ is given by

$$w(x, y) = G(x, y, \sigma)m(x, y), \tag{17.9}$$

and a quantized histogram h'_θ for a direction θ', which is an angle when θ is divided into 36 angles, is given by

$$h'_\theta(x, y) = \Sigma_x \Sigma_y w(x, y)\delta[\theta', \theta(x, y)]. \tag{17.10}$$

δ indicates the Kronecker delta given by

$$\delta[\theta', \theta(x, y)] = \begin{cases} 1 & (\theta_p = \theta') \\ 0 & (\theta_p \neq \theta') \end{cases} \tag{17.11}$$

when the quantized angle of $\theta(x, y)$ is θ_p.

17.2.2.4 Description of Quantities of Features

To describe quantities of features based on the orientation obtained in previous section, surrounding region of a feature point is rotated to the direction of the orientation. The region is divided by 4 × 4 blocks. A histogram on 8 directions is made for each block by the same method explained in the previous section. This process produces a 128(4 × 4 × 8)-dimensional feature vector. The quantity of SIFT feature is represented by this vector.

17.2.3 Object Detection

Let us suppose that the SIFT features of the working robot, hand of its arm, and target object to be manipulated are initially given to the observing robot with their registered images. In the beginning of observation, the observing robot looks around and detects corresponding points on the SIFT features of the object in captured images.

In order to find the corresponding points, the 128-dimensional feature vector, which is described in Sect. 17.2.2, are calculated for each feature point in a registered image and that in an input image. Let $\boldsymbol{v}_{Rj}$ be the vector for j-th feature point in the registered image, and $\boldsymbol{v}_{Ik}$ be the vector for k-th feature point in an input image. The Euclidean distance, d, between the vectors for a feature point of the registered image and an input image is calculated as

$$d = \sqrt{(\boldsymbol{v}_{Rj} - \boldsymbol{v}_{Ik})^{\mathrm{T}}(\boldsymbol{v}_{Rj} - \boldsymbol{v}_{Ik})} \tag{17.12}$$

where $\boldsymbol{A}^{\mathrm{T}}$ represents the transpose of $\boldsymbol{A}$. Then, two candidate points which have the smallest and the second-smallest distances of the vector, d_{I1} and d_{I2}, are then picked in the input image. If these distances satisfy $d_{I1} < d_{I2} \times 0.5$, d_{I1} is picked as the corresponding point.

The correspondences for all feature points of a registered image on an object to be detected are searched in an input image. As the result, if the ratio of the number of the corresponding point to the registered image is larger than a threshold, it is judged the object exists at the region that covers the feature points in the input image.

17.2.4 Calculation of Object Position

The observing robot can detect the 3-D positions of the working robot and target object by parallax of corresponding points between two or more images observed from different views. The observing robot can extract corresponding points on the working robot or target object in a new input image in the same way as object detection described in Sect. 17.2.3.

Figure 17.3 shows the model for the detection. Suppose that a point, P_1, on the hand is detected by the camera at corresponding points $\boldsymbol{m_1}$ and $\boldsymbol{m'_1}$ on two image planes at different locations C_1 and C_2.

The position of P_1 is obtained as follows. Let $\boldsymbol{R}$ and $\boldsymbol{T}$ be the rotation matrix and translation vector between C_1 and C_2 of the camera. Then we get

$$\boldsymbol{M_1} = \boldsymbol{R}\boldsymbol{M'_1} + \boldsymbol{T} \tag{17.13}$$

where $\boldsymbol{M_1}$ and $\boldsymbol{M'_1}$ are vectors of P_1 in the camera coordinate C_1 and C_2 respectively. Let $\boldsymbol{A}$ be an intrinsic matrix [3] of the camera. Equation (17.13) then becomes

$$w_1\boldsymbol{A}^{-1}\boldsymbol{m_1} = w'_1\boldsymbol{R}\boldsymbol{A}^{-1}\boldsymbol{m'_1} + \boldsymbol{T} \tag{17.14}$$

where w_1 and w'_1 are scale factors for $\boldsymbol{m_1}$ and $\boldsymbol{m'_1}$ respectively. We can obtain $\boldsymbol{A}$ by the camera calibration. The robot is able to know $\boldsymbol{R}$ and $\boldsymbol{T}$ by odometry. Thus, the robot can calculate w_1 and w'_1, then obtain 3D position of P_1 from the corresponding points.

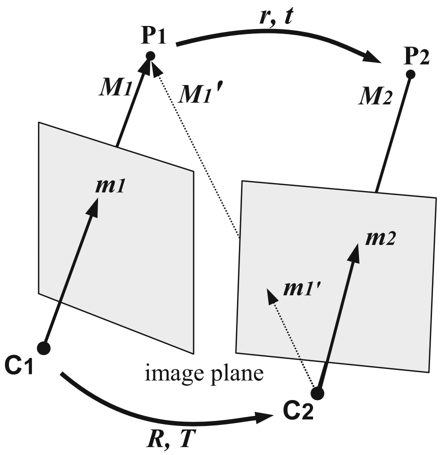


Fig. 17.3 Model of object position and movement detection

17.2.5 Hand Movement Detection

In the same way to the computation of object position, the motion of the hand of the working robot can also be calculated from two or more different images. In this case, the observing robot may observe at a same position because the hand moves. The observing robot can detect the hand of the working robot from the SIFT features and calculate its 3-D motion, which is relative rotation and translation of the hand of the working robot to the target object, from the feature points and their correspondences in two images.

In the model of Fig. 17.3, let us suppose the hand moves from P_1 to P_2 and the camera detects P_2 at point $\boldsymbol{m_2}$ on the image plane of C_2. The rotation and translation from P_1 to P_2 are computed as follows.

Let $\boldsymbol{r}$ and $\boldsymbol{t}$ be the rotation matrix and the translation vector of the moving point. We then get

$$\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M_2} = \boldsymbol{R}^{\mathrm{T}}(\boldsymbol{M_1} - \boldsymbol{T}) + \boldsymbol{t} \tag{17.15}$$

where $\boldsymbol{M_2}$ is the vector of P_2 in the camera coordinate C_2. This indicates the vectors $\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M_2}$, $\boldsymbol{R}^{\mathrm{T}}(\boldsymbol{M_1} - \boldsymbol{T})$, and $\boldsymbol{t}$ are on one plane. Consequently, it leads to

$$(\boldsymbol{M_1^{\mathrm{T}}} - \boldsymbol{T}^{\mathrm{T}})\boldsymbol{R}[\boldsymbol{t}]_{\times}\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M_2} = 0 \tag{17.16}$$

where $[\boldsymbol{t}]_{\times}$ is a skew symmetric matrix which represents operation of vector product with $\boldsymbol{t}$. Let $\boldsymbol{Q} = [\boldsymbol{t}]_{\times}\boldsymbol{r}^{\mathrm{T}}$. Using $\boldsymbol{m_1}$ and $\boldsymbol{m_2}$, (17.16) becomes

$$(w_1\boldsymbol{m_1^{\mathrm{T}}}(\boldsymbol{A}^{-1})^{\mathrm{T}} - \boldsymbol{T}^{\mathrm{T}})\boldsymbol{RQA}^{-1}\boldsymbol{m_2} = 0. \tag{17.17}$$

Because w_1 is already obtained in computing the position of P_1, $\boldsymbol{Q}$ can be computed from 8 or more corresponding points [4, 5] with decomposition, then $\boldsymbol{r}$ and $\boldsymbol{t}$ are obtained [6].

First, the computation of $\boldsymbol{Q}$ is described below.

Supposing that $\boldsymbol{Q}$ consists of three column vectors, $\boldsymbol{Q} = (\boldsymbol{q_1}\,\boldsymbol{q_2}\,\boldsymbol{q_3})$, we can define a column vector

$$\boldsymbol{q} = (\boldsymbol{q_1}^{\mathrm{T}}\,\boldsymbol{q_2}^{\mathrm{T}}\,\boldsymbol{q_3}^{\mathrm{T}})^{\mathrm{T}}. \tag{17.18}$$

Also, Let's suppose

$$(w_{i1}\boldsymbol{m_{i1}}^{\mathrm{T}}(\boldsymbol{A}^{-1})^{\mathrm{T}}\boldsymbol{R}^{\mathrm{T}} - T^{\mathrm{T}}) = (p_{ix}\ p_{iy}\ p_{iz}), \tag{17.19}$$

and

$$\boldsymbol{A}^{-1}\boldsymbol{m_{i1}} = (s_{ix}\ s_{iy}\ 1)^{\mathrm{T}} \tag{17.20}$$

for i-th corresponding point. The use of

$$\boldsymbol{b_i}^{\mathrm{T}} = (s_{ix}p_{ix}\ s_{ix}p_{iy}\ s_{ix}p_{iz}\ s_{iy}p_{ix}\ s_{iy}p_{iy}\ s_{iy}p_{iz}\ p_{ix}\ p_{iy}\ p_{iz}) \tag{17.21}$$

leads to

$$\boldsymbol{b_i}^{\mathrm{T}}\boldsymbol{q} = \boldsymbol{0} \tag{17.22}$$

from (17.17). For $i = 1, \ldots, N$,

$$\boldsymbol{Bq} = \boldsymbol{0} \tag{17.23}$$

is obtained where

$$\boldsymbol{B} = (\boldsymbol{b_1}\,\boldsymbol{b_2}\ \ldots\ \boldsymbol{b_N})^{\mathrm{T}}. \tag{17.24}$$

Equation (17.23) would not be $\boldsymbol{0}$ practically because of image noise etc. $\boldsymbol{Q}$ is therefore obtained by computing $\boldsymbol{q}$ that minimizes $\|\boldsymbol{Bq}\|$. Such a $\boldsymbol{q}$ is given as a unit eigen-vector corresponding to the minimum eigen value of $\boldsymbol{BB}^{\mathrm{T}}$ with a condition of

$$\|\boldsymbol{q}\|^2 = \mathrm{trace}(\boldsymbol{QQ}^{\mathrm{T}}) = \mathrm{trace}([\boldsymbol{t}]_\times[\boldsymbol{t}]_\times^{\mathrm{T}}) = 2\|\boldsymbol{t}\|^2. \tag{17.25}$$

Next, we describe how to calculate $\boldsymbol{t}$ and $\boldsymbol{r}$ from $\boldsymbol{Q}$.

Although $\boldsymbol{Q}$ should meet

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{t} = -\boldsymbol{r}[\boldsymbol{t}]_{\times}\boldsymbol{t} = \boldsymbol{0}, \tag{17.26}$$

this equation will not become 0 because of an influence of noise etc. Therefore, $\boldsymbol{t}$ can be computed so that $\|\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{t}\|$ is minimized. Such a $\boldsymbol{t}$ is a unit eigen-vector corresponding to the minimum eigen value of $\boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}}$. A singular value decomposition of $\boldsymbol{Q}$, $\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$, gives three column vectors for $\boldsymbol{U}$. Then, $\boldsymbol{t}$ is obtained as the column vector that corresponds to the minimum singular value.

Furthermore, let $\boldsymbol{R}_z$ rotation matrix of $(\pi/2)$ on z axis, then $\boldsymbol{Q}$ can be represented as

$$\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{\Sigma}(\boldsymbol{U}\boldsymbol{R}_z)^{\mathrm{T}}(\boldsymbol{U}\boldsymbol{R}_z)\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{T}'R' \tag{17.27}$$

where $\boldsymbol{T}' = \boldsymbol{U}\boldsymbol{\Sigma}(\boldsymbol{U}\boldsymbol{R}_z)^{\mathrm{T}}$ and $\boldsymbol{R}' = \boldsymbol{U}\boldsymbol{R}_z\boldsymbol{V}^{\mathrm{T}}$. $\boldsymbol{T}'$ represents $[\boldsymbol{t}]_{\times}$ because $\boldsymbol{T}'^{\mathrm{T}} = -\boldsymbol{T}'$, and $\boldsymbol{r}'$ is a rotation matrix because $\boldsymbol{R}'\boldsymbol{R}'^{\mathrm{T}} = \boldsymbol{I}$. Consequently

$$[\boldsymbol{t}]_{\times} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{R}_z^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}, \tag{17.28}$$

$$\boldsymbol{r} = \boldsymbol{V}\boldsymbol{R}_z^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}. \tag{17.29}$$

We can obtain hand movement information from these computations. In this computation, image positions are normalized based on the method given by [7] to minimize computational error.

17.2.6 Hand Motion Control by Working Robot

The working robot can plan a trajectory for hand motion depending on the position and orientation of the target object. This study assumes that the target object is a cylinder and the robot knows the shape of object beforehand for simplicity. Given the position and orientation of the object and hand from the observing robot, the working robot plans the trajectory of its hand to grasp the object.

This study considered simple trajectory of the hand: the path from top of the robot to the object consists of lines and circular arcs. When the object is in reachable area for the hand of the working robot, the robot moves the hand outside at middle height of the object, then the hand approaches the object with keeping its height.

The joint angles at each time in the arm motion are calculated from infinitesimal differences of the hand position and orientation in a sampling cycle time using Jacobian. The working robot checks the trajectory whenever it receives new information of the position and orientation of the target object and hand from the observing robot. If the trajectory is not appropriate to approach the object due to some errors, the robot performs the path planning again and updates the hand trajectory in the same manner.

17.3 Experiments

17.3.1 Experimental Setup

The method described above has been implemented to two wheeled-mobile robots, Pioneer P3-DX [8], which is 393 mm in width, 445 mm in length, and 237 mm in height. Figure 17.4 shows those robots.

One robot shown in Fig. 17.4a has a camera, Canon VC-C50i, which is able to rotate in pan and tilt directions so that it is qualified as the observing robot. A board computer, Interface PCI-B02PA16W, was also mounted in order to process images from the camera in observation as well as control the movement of the robot. We utilized OpenCV for developing software for the image processing in observation.

Another robot shown in Fig. 17.4b has a 6-DOF manipulator to be the working robot. A 1-DOF hand is attached at the end of the manipulator. The hand is 140 mm in width, 160 mm in length, and 100 mm in height. This robot doesn't have any sensor to detect an object.

Figure 17.4c shows the target object used in the experiment. It was a cylindrical bottle, which was 65 mm in diameter and 390 mm in height. Texture patterns were attached on its surface so that the observing robot can detect feature points easily.

Figure 17.5 shows an overview of this experiment. The working robot stayed at one position, denoted as W. The observing robot looked for the working robot, its arm, and the target object at the point O_1 and detected them. It then moved to the point O_2 and O_3 and calculated 3-D positions of the objects from the views at the points. These detected position information was sent to the working robot. The working robot then started moving its arm to handle the target object. The observing robot kept tracking the hand motion and calculated 3-D motion of the hand of the working robot continuously.

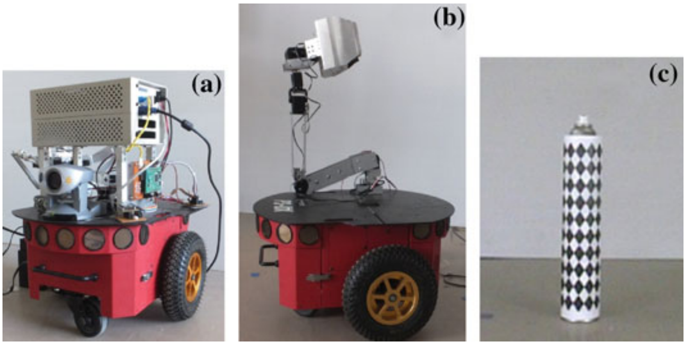


Fig. 17.4 Robots and target object used in the experiment: **a** observing robot, **b** working robot, and **c** target object

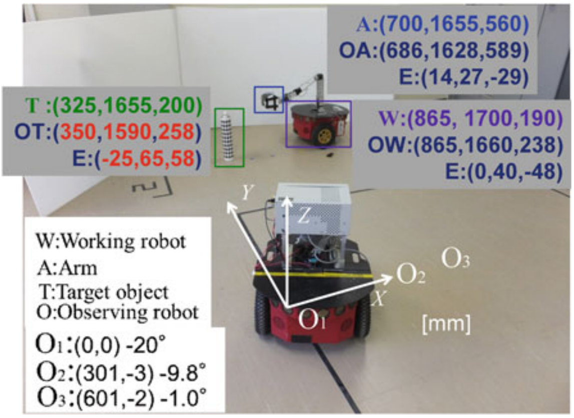


Fig. 17.5 Overview of the experiment with the result of detection by the observing robot. O_1, O_2, O_3: observation locations (*view angles* are also denoted), W real position of the working robot, OW detected position of the robot, A position of hand of the working robot, OA detected hand position, T position of the target object, OT detected position of the object. Their position values are denoted for each; E shows error between real and detected values

17.3.2 Detection of Working Robot and Target Object

Figure 17.6 shows images registered in the experiment: (a) hand of the working robot, (b) the working robot, and (c) the target object. Rectangle regions were registered for them as shown in the figures. SIFT features in each region were extracted by the method described in Sect. 17.2.2. These features were used for definition and detection of them by the observing robot. The observing robot extracted SIFT features from an input image and obtained correspondences for the working robot, its arm, and the target objects respectively to detect them. A region in which the ratio of the number of corresponding points between the registered image and an input image is larger than 0.4 was extracted, as the detection of each target. Figure 17.7 shows detected regions for the working robot, its hand, and the target object. These regions are indicated by green, red, and blue rectangles respectively.

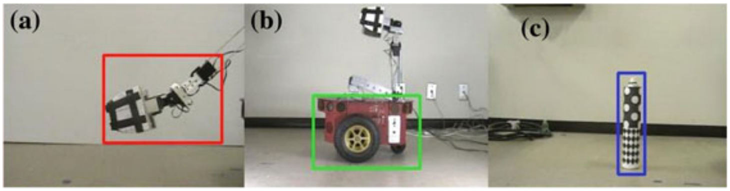


Fig. 17.6 Registered images: **a** hand of the working robot, **b** the working robot, and **c** target object

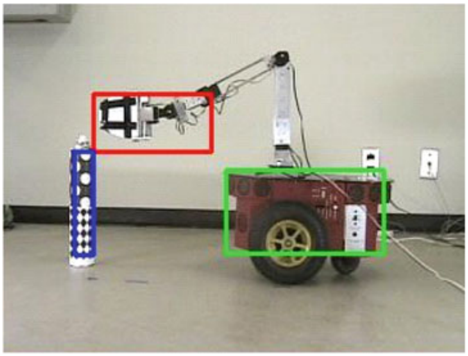


Fig. 17.7 Detected regions for the working robot (*green rectangle*), its hand (*red*), and target object (*blue*)

17.3.3 Detection of Positions

The observing robot changed the position from O_1 to O_2 and O_3 in Fig. 17.5 to observe the working robot with its hand and the target object in different visual angles.

Figure 17.8 shows the images which were taken by the observing robot at those view points. SIFT features were extracted from these images, and the corresponding points on each region between them were computed. The correspondences between two images at adjacent view points are shown by the connections of pink line in Fig. 17.8. The 3-D positions for those targets were calculated from parallax information of these corresponding points based on stereo vision theory.

The obtained positions of them are described in Fig. 17.5: *OW* for the working robot, *OA* for the hand, and *OT* for the target object. The error values to real positions, which are denoted as W, A, and T respectively, are also described below the detected positions as E. The maximum error was 65 mm for the position in the Y direction for the target object. The authors consider this error is in the acceptable range for object handling in consideration of the sizes of the hand and the target object.

17.3.4 Detection of Hand Movement

Figure 17.9 shows an overview of the hand movement detection. The observing robot stayed at O_3 after detecting hand position of the working robot; then tracked the hand movement during the working robot moved its hand from the position A to A'. The observing robot obtained corresponding points of SIFT features in the region of hand in the same way as position detection. Figure 17.10 shows detected corresponding points of SIFT features from three images in the sequence of hand movement for the working robot. Each correspondence is connected by pink line each other.

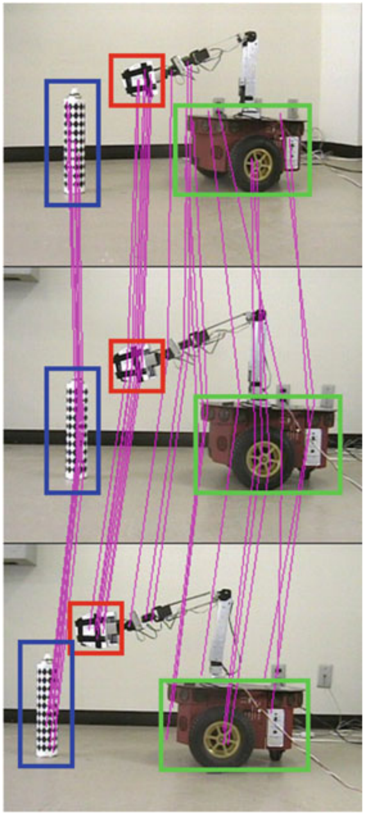


Fig. 17.8 Detected correspondences of SIFT features (connected by *pink lines*) for the images at the view points O_1 (*upper row*), O_2 (*middle row*), and O_3 (*lower row*). The *rectangles* indicate detected regions for the working robot (*green*), its hand (*red*), and the target object (*blue*)

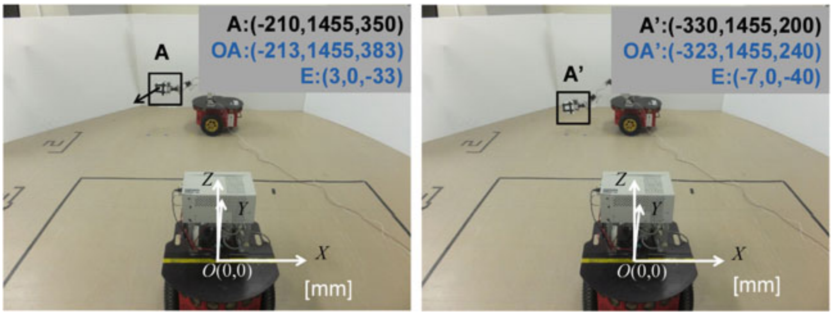


Fig. 17.9 Overview of the detection of hand movement. The working robot moved its hand from the position A (*left panel*) to the position A' (*right panel*). OA and OA' show detected positions by the observing robot. The error value to the real position, E, are also denoted

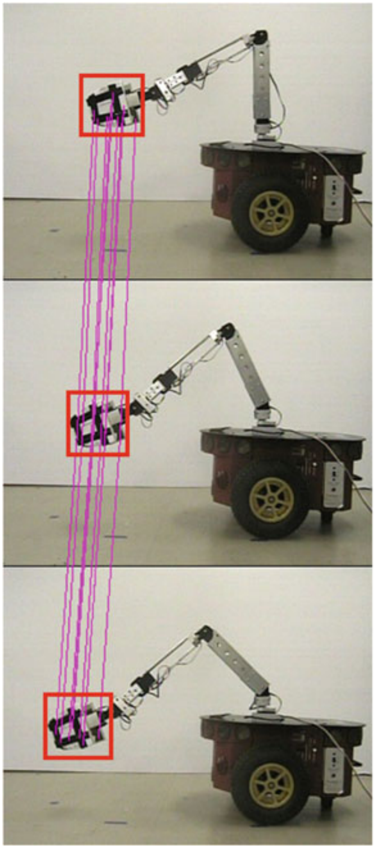


Fig. 17.10 Detected corresponding points of SIFT features on hand of the working robot in hand movement

From positions of corresponding points, a hand position was computed for each image, and a translation vector was obtained. In this experiment, the depth information of the hand and target object are very short to the distance between the observing robot and them, so they can be neglected. We therefore supposed weak perspective projection for images in the observation of hand movement for simplicity. The detected vector was $(-110, 0, -143)^{\mathrm{T}}$ to the real vector $(-120, 0, -150)^{\mathrm{T}}$. The Y direction of the hand movement did not change because weak perspective projection. Figure 17.11 shows extracted translation vector of the hand. The result shows that appropriate motion vector was detected.

In this experiment, the time taken for computing SIFT features and finding the corresponding points was from 339 to 616 ms per one image; it depended on the number of the feature points. The author s consider that the time enables the observing robot to detect the hand motion of the working robot in real time. Moreover, if it is possible to reduce picked feature points more effectively, the corresponding points would be found faster.

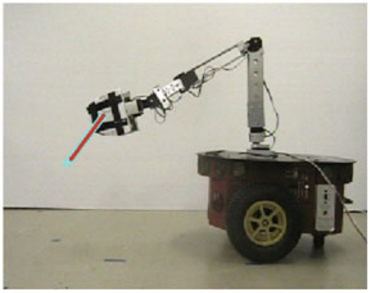


Fig. 17.11 Detected translation vector of the hand (*red arrow*) in hand movement of the working robot. The *blue arrow* is the real movement vector

17.4 Conclusions

This study described a method for assisting a working robot's object handling task based on an observation by another cooperative partner robot in the situation that the working robot can not perceive the object. The fundamental experiments confirmed processes in the proposed method: detection of the working robot, its hand, and the target object based on correspondences SIFT features, computation of their positions, and detection of hand movement of the working robot. Our future work will proceed with consideration orientations of the robot and objects, and expand this method to practical case toward real-time cooperation.

References

1. Kuniyoshi, Y., Rickki, J., Ishii, M., Rougeaux, S., Kita, N., Sakane, S., Kakikura, M.: Vision based behaviors for multi-robot cooperation. In: Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems '94, vol. 2, pp. 925–932 (1994)
2. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceeding of IEEE International Conference on Computer Vision (ICCV), pp. 1150–1157 (1999)
3. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)
4. Luong, Q.-T., Faugeras, O.D.: Self-calibration of a moving camera from point correspondences and fundamental matrices. Int. J. Comput. Vis. **23**(3), 261–289 (1997)
5. Luong, Q.T., Faugeras O.D.: The fundamental matrix: theory, algorithms, and stability analysis. Int. J. Comput. Vis. **17**(1), 43–75(33) (1996)
6. Weng, J., Ahuja, N., Huang, T.S.: Motion and structure from two perspective views: algorithms, error analysis, and error estimation. IEEE Trans. Pattern Anal. Machine Intell. **11**(5), 451–476 (1989)
7. Hartley, R.I.: In defense of the eight-point algorithm. IEEE Trans. Pattern Anal. Mach. Intell. **19**(6), 580–593 (1997)
8. Mobile Robots Pioneer P3-DX: http://www.mobilerobots.com/ResearchRobots/PioneerP3DX.aspx

Part III
Signal Processing, Sensors, Systems Modeling and Control

Chapter 18
Passive Parametric Macromodeling by Using Sylvester State-Space Realizations

Elizabeth Rita Samuel, Luc Knockaert and Tom Dhaene

Abstract A judicious choice of the state-space realization is required in order to account for the assumed smoothness of the state-space matrices with respect to the design parameters. The direct parameterization of poles and residues may be not appropriate, due to their possible non-smooth behavior with respect to design parameters. This is avoided in the proposed technique, by converting the pole-residue description to a Sylvester description which is computed for each *root macromodel*. This technique is used in combination with suitable parameterizing schemes for interpolating a set of state-space matrices, and hence the poles and residues *indirectly*, in order to build accurate parametric macromodels. The key features of the present approach are first the choice of a proper pivot matrix and second, finding a well-conditioned solution of a Sylvester equation. Stability and passivity are guaranteed by construction over the design space of interest. Pertinent numerical examples validate the proposed Sylvester technique for parametric macromodeling.

Keywords Sylvester equation · Parametric macromodel · State-space matrices · Interpolation

18.1 Introduction

Multiple simulations are often required during a typical design process of electromagnetic (EM) systems, design space exploration, design optimization, and sensitivity analysis for different design parameter values (e.g., layout features). Parametric

E.R. Samuel (✉) · L. Knockaert · T. Dhaene
Ghent University—IMinds, Gaston Crommenlaan 8 Bus 201, B9050 Gent, Belgium
e-mail: elizabeth.ritasamuel@ugent.be
lizita3@gmail.com
http://www.sumo.intec.ugent.be

L. Knockaert
e-mail: luc.knockaert@ugent.be

T. Dhaene
e-mail: tom.dhaene@ugent.be

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_18

macromodels are valuable tools for efficiently and accurately performing these design activities, while avoiding new measurements or simulations at each new parameter configuration. Parameterized macromodels are multivariate models that describe the complex behavior of EM systems, typically characterized by frequency (or time) and several geometrical and physical design parameters, such as layout or substrate features. Recently, parametric macromodeling techniques able to guarantee overall stability and passivity have been proposed in [1–4]. Unfortunately, these methods are sensitive to issues related to the interpolation of state-space matrices [5], such as the smoothness of the state-space matrices as a function of the parameters.

The direct parameterization of poles and residues is in general not appropriate, due to their possible non-smooth behavior with respect to the design parameters. This is avoided in the proposed technique, where a suitable set of state-space matrices is parametric and, hence, the poles and residues *indirectly*. A conversion from the pole-residue description obtained by means of vector fitting (VF) to a Sylvester description is computed for each *root macromodel*. This avoids the direct parameterization of poles and residues, when interpolating a set of state-space matrices in order to build a parametric macromodel. The present technique is able to deal accurately with bifurcation effects of poles and residues [6].

The key features of the Sylvester realization technique are first the choice of a pivot or reference matrix and second, the obtention of a well-conditioned solution to the Sylvester equation. Since the same pivot matrix is used for all state-space realizations of the *root macromodels*, smooth variations of the state-space matrices with respect to the design parameters can be expected. The state-space matrices obtained from the Sylvester realization are used to obtain matrix solutions of the linear matrix inequalities (LMIs) pertaining to the positive-real or bounded-real lemma, and this information is then used to perform a passivity preserving interpolation of the state-space matrices. The computations can be carried out using the solution of LMIs or algebraic Riccati equations (AREs) to generate a descriptor state-space format that preserves positive-realness or bounded-realness. Finally, suitable interpolation schemes are used to build accurate parametric macromodels which preserve stability and passivity. Pertinent numerical examples validate the proposed Sylvester realization technique for macromodeling based on interpolation of state-space matrices.

18.2 Parametric Macromodeling

We start with a set of passive models $\mathcal{S}_k, \quad k = 1 \cdots N$ with given minimal realizations

$$\mathcal{S}_k \equiv \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} \tag{18.1}$$

state-space equations

$$\dot{x} = A_k x + B_k u \tag{18.2}$$

$$y = C_k x + D_k u \tag{18.3}$$

and transfer functions

$$H_k(s) = C_k(sI - A_k)^{-1} B_k + D_k \tag{18.4}$$

In this paper we suppose that all realizations $\mathcal{S}_k$ have the same McMillan degree n and number of ports $m \leq n$. This means that all A_k, B_k, C_k, D_k matrices have respective sizes $n \times n, n \times m, m \times n, m \times m$. We further suppose that all matrices A_k are Hurwitz stable i.e., all their poles are in the open left halfplane.
We aim at obtaining a generic parametric realization of the form

$$\mathcal{S}(\mathbf{g}) \equiv \begin{bmatrix} A(\mathbf{g}) & B(\mathbf{g}) \\ C(\mathbf{g}) & D(\mathbf{g}) \end{bmatrix} \tag{18.5}$$

with vectorial parameter $\mathbf{g}$ such that the models $\mathcal{S}_k$ can be considered as snapshots of $\mathcal{S}(\mathbf{g})$ generated by freezing the parameter $\mathbf{g}$ at the fixed values $\mathbf{g}_k$. The objective is to construct a guaranteed passive macromodel $\tilde{\mathcal{S}}(\mathbf{g})$ by means of the discrete models $\mathcal{S}_k$, such that $\tilde{\mathcal{S}}(\mathbf{g})$ is passive, smooth and close to $\mathcal{S}(\mathbf{g})$[1] in some sense.

18.3 State-Space Realizations for Parametric Macromodeling

To obtain accurate parametric macromodels by interpolation of the state-space matrices, the choice of the state-space realization is fundamental.
In this section, we will discuss the well-known Gilbert realization, the balanced realization and then the proposed novel Sylvester realization, preceded by a subsection on passive parametric interpolation.

18.3.1 Gilbert Realization

The minimal state-space realization problem for linear time invariant (LTI) systems was first stated by Gilbert [7], who gave an algorithm for transforming a transfer function into a system of differential equations. The Gilbert approach is based on partial-fraction expansions.

$$H(s) = R_{0,\mathbf{p}_k} + \sum_{n=1}^{N} \frac{R_{n,\mathbf{p}_k}}{s - z_{n,\mathbf{p}_k}} \tag{18.6}$$

[1] The exact generic realization $\mathcal{S}(\mathbf{g})$ is analytically unknown in the sense that for each new value of $\mathbf{g}$ an oracle (or black-box function) has to be consulted.

where $R_{n,\mathbf{p}_k}$ and $s - z_{n,\mathbf{p}_k}$ are respectively the model residues and poles, with $R_{0,\mathbf{p}_k}$ being the direct coupling constant. The poles and the residues are stamped directly in the $A(\mathbf{p})$ and $C(\mathbf{p})$ matrices using the Gilbert realization [7]. It is well-known that model poles and residues are very sensitive to even small variations of the design parameters, resulting in quite irregular variations of each pole in the design space. Since poles and residues may present a highly non-smooth behavior with respect to the design parameters, achieving a reasonable accuracy in parametric macromodels built by interpolation of state-space matrices becomes difficult, due to the fact that pole and residue trajectories as a function of $\mathbf{p}$ are not well defined.

18.3.2 Balanced Realization

A minimal and stable realization is called balanced, if the controllability and observability Gramians are equal and diagonal [8]. Every minimal system can be brought into balanced form. The balanced realization can be implemented using the Matlab function `balreal`. This routine uses the eigendecomposition of the product of the observability and controllability Gramians to construct the balancing transformation matrix.

The most interesting properties of balanced realizations is associated with the uniqueness property of the balancing transformation [9]. When the eigenvalues (real and nonnegative) of the product of the controllability and observability Gramians, are distinct, then the balancing transformation matrix is unique. If, on the other hand, two or more eigenvalues are repeated, then their corresponding eigenvectors can be rotated arbitrarily in the corresponding eigenspace. Thus as stated in [5, 9], uniqueness is guaranteed up to a sign and it may affect the smoothness of the state-space matrices as functions of the design parameters.

18.3.3 Passive Interpolation of LTI Systems

Passivity is an important property to satisfy because stable, but not passive macromodels can produce unstable systems when connected to other stable, even passive, loads. As a first approach we opt for straightforward passive interpolation. Since each macromodel $\mathcal{S}_k$ is passive, by the positive real lemma [10] we know that this is the case if there exists a positive definite symmetric matrix P_k such that the Linear Matrix Inequality (LMI) [11]

$$\mathcal{L}_k = \begin{bmatrix} A_k^T P_k + P_k A_k & P_k B_k - C_k^T \\ B_k^T P_k - C_k & -D_k - D_k^T \end{bmatrix} \leq 0 \tag{18.7}$$

is satisfied. Now consider a positive interpolation kernel [12] $\mathcal{K}(\mathbf{g}_k, \mathbf{g}) = \mu_k(\mathbf{g})$ satisfying

$$\mu_k(\mathbf{g}) \geq 0, \quad \mu_k(\mathbf{g}_l) = \delta_{k,l} \tag{18.8}$$

It is clear that the interpolatory parametric LMI

$$\mathcal{L}(\mathbf{g}) = \sum_{k=1}^{N} \mu_k(\mathbf{g})\mathcal{L}_k \tag{18.9}$$

is semi-negative definite, and hence if we parameterize all entries of the $P_k A_k$, $P_k B_k$, C_k, D_k, P_k matrices as

$$\begin{aligned}
P(\mathbf{g})A(\mathbf{g}) &= \sum_{k=1}^{N} \mu_k(\mathbf{g}) P_k A_k \\
P(\mathbf{g})B(\mathbf{g}) &= \sum_{k=1}^{N} \mu_k(\mathbf{g}) P_k B_k \\
C(\mathbf{g}) &= \sum_{k=1}^{N} \mu_k(\mathbf{g}) C_k \\
D(\mathbf{g}) &= \sum_{k=1}^{N} \mu_k(\mathbf{g}) D_k \\
P(\mathbf{g}) &= \sum_{k=1}^{N} \mu_k(\mathbf{g}) P_k
\end{aligned} \tag{18.10}$$

it is seen by inspection that the parametric realization

$$\tilde{\mathcal{S}}(\mathbf{g}) \equiv \begin{bmatrix} A(\mathbf{g}) & B(\mathbf{g}) \\ C(\mathbf{g}) & D(\mathbf{g}) \end{bmatrix} \tag{18.11}$$

thus obtained is passive with LMI solution matrix $P(\mathbf{g})$.

18.3.4 Uniform Approach by Sylvester Equations

The issues with the passive parametric interpolation procedure when using the Gilbert and balanced realizations are twofold. First, there are 5 interpolation Eq. (18.10) to be satisfied. Second, and most important, although the interpolation technique yields (by construction) the discrete macro-models $\mathcal{S}_k$ for $\mathbf{g} = \mathbf{g}_k$, it is not at all sure that the interpolated matrices $A(\mathbf{g})$, $B(\mathbf{g})$, $C(\mathbf{g})$, $D(\mathbf{g})$, $P(\mathbf{g})$ will behave smoothly between the nodes $\mathbf{g}_k$. The reason for this is that minimal realizations are all equivalent modulo a similarity transformation, i.e., two realizations related by

$$\begin{bmatrix} \tilde{A} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{bmatrix} = \begin{bmatrix} X^{-1}AX & X^{-1}B \\ CX & D \end{bmatrix} \tag{18.12}$$

where X is any nonsingular matrix, yield the same transfer function

$$H(s) = C(sI - A)^{-1}B + D = \tilde{C}(sI - \tilde{A})^{-1}\tilde{B} + \tilde{D} \tag{18.13}$$

In order to install uniformity we propose the state-space feedback realization

$$\dot{x} = \mathbf{A}x + \mathbf{B}_k v \tag{18.14}$$

$$y = \hat{\mathbf{C}}_k x + D_k v \tag{18.15}$$

$$v = -\mathbf{F}x + u \tag{18.16}$$

where $\mathbf{A}$ is a fixed $n \times n$ pivot matrix and $\mathbf{F}$ is a fixed $m \times n$ state-space feedback matrix. This realization can be written as

$$\mathcal{R}_k \equiv \begin{bmatrix} \mathbf{A} - \mathbf{B}_k\mathbf{F} & \mathbf{B}_k \\ \hat{\mathbf{C}}_k - D_k\mathbf{F} & D_k \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{B}_k\mathbf{F} & \mathbf{B}_k \\ \mathbf{C}_k & D_k \end{bmatrix} \tag{18.17}$$

For $\mathcal{R}_k$ and $\mathcal{S}_k$ to be equivalent, we need the existence of nonsingular matrices X_k such that

$$\mathbf{A} - \mathbf{B}_k\mathbf{F} = X_k^{-1}A_kX_k \tag{18.18}$$

$$\mathbf{B}_k = X_k^{-1}B_k \tag{18.19}$$

$$\mathbf{C}_k = C_kX_k \tag{18.20}$$

Eliminating (18.19) from (18.18) we obtain the Sylvester equation

$$A_kX_k - X_k\mathbf{A} + B_k\mathbf{F} = 0 \tag{18.21}$$

for the unknown matrix X_k. We need the following

Theorem 1 *The Sylvester equation (18.21) has a unique nonsingular solution X_k provided the pair (A_k, B_k) is controllable, the pair $(\mathbf{A}, \mathbf{F})$ is observable, and the intersection of the eigenspectra of A_k and $\mathbf{A}$ is empty.*

Proof See [13, 14].

The Sylvester equations are routinely solved by the Matlab function `lyap`.

Remark 1 The Sylvester realizations $\mathcal{R}_k$, given the pivot matrix $\mathbf{A}$ and feedback matrix $\mathbf{F}$, are all unique by construction. For the choice of $\mathbf{A}$ we can take a block-diagonal or block-Jordan matrix[14] which never shares eigenvalues with any of the

A_k matrices. This can be accomplished by choosing the eigenvalues of $\mathbf{A}$ close to the imaginary axis (see also the numerical simulations). The choice of $\mathbf{F}$ is subject to the requirement that the pair $(\mathbf{A}, \mathbf{F})$ has to be observable. In some cases such as the Gilbert [7] or Vector Fitting [15] realization, all matrices B_k are equal, and then a judicious choice for $\mathbf{F}$ is $\mathbf{F} = B_k^T$. More generally speaking, $\mathbf{F}$ can be chosen quite freely, or its choice can be imbedded in the overall Sylvester algorithm [16].Next, we parameterize the new LMI's

$$\begin{bmatrix} (\mathbf{A} - \mathbf{B}_k\mathbf{F})^T \tilde{P}_k + \tilde{P}_k(\mathbf{A} - \mathbf{B}_k\mathbf{F}) & \tilde{P}_k\mathbf{B}_k - \mathbf{C}_k^T \\ \mathbf{B}_k^T \tilde{P}_k - \mathbf{C}_k & -D_k - D_k^T \end{bmatrix} \leq 0 \tag{18.22}$$

as in Sect. 18.3.3, four last equations of (18.10), i.e.,

$$\begin{aligned} \tilde{P}(\mathbf{g})\mathbf{B}(\mathbf{g}) &= \sum_{k=1}^{N} \mu_k(\mathbf{g})\tilde{P}_k\mathbf{B}_k \\ \mathbf{C}(\mathbf{g}) &= \sum_{k=1}^{N} \mu_k(\mathbf{g})\mathbf{C}_k \\ D(\mathbf{g}) &= \sum_{k=1}^{N} \mu_k(\mathbf{g})D_k \\ \tilde{P}(\mathbf{g}) &= \sum_{k=1}^{N} \mu_k(\mathbf{g})\tilde{P}_k \end{aligned} \tag{18.23}$$

The first equation of (18.10) has no counterpart in equations (18.23), since it is easy to show that

$$\tilde{P}(\mathbf{g})[\mathbf{A} - \mathbf{B}(\mathbf{g})\mathbf{F}] = \sum_{k=1}^{N} \mu_k(\mathbf{g})\tilde{P}_k[\mathbf{A} - \mathbf{B}_k\mathbf{F}] \tag{18.24}$$

Finally, the parametric Sylvester realization is then simply

$$\mathcal{R}(\mathbf{g}) \equiv \begin{bmatrix} \mathbf{A} - \mathbf{B}(\mathbf{g})\mathbf{F} & \mathbf{B}(\mathbf{g}) \\ \mathbf{C}(\mathbf{g}) & D(\mathbf{g}) \end{bmatrix} \tag{18.25}$$

This also implies that we can track the pole trajectories of the parametric system easily as the eigenvalues of the matrix $\mathbf{A} - \mathbf{B}(\mathbf{g})\mathbf{F}$, which depends only on the parameterization of $\mathbf{B}(\mathbf{g})$, i.e.

$$\mathbf{B}(\mathbf{g}) = \left(\sum_{k=1}^{N} \mu_k(\mathbf{g})\tilde{P}_k\right)^{-1} \sum_{k=1}^{N} \mu_k(\mathbf{g})\tilde{P}_k\mathbf{B}_k \tag{18.26}$$

Remark 2 Note that, even if passivity is not required, the Sylvester realizations $\mathcal{R}_k$ can be very useful for parameterization. Suppose the interpolation kernel $\mathcal{K}(\mathbf{g}_k, \mathbf{g}) = \mu_k(\mathbf{g})$ is not necessarily positive, but satisfies partition of unity,[2] i.e.,

$$\sum_k \mu_k(\mathbf{g}) = 1, \quad \mu_k(\mathbf{g}_l) = \delta_{k,l} \tag{18.27}$$

Then it is clear that the interpolation procedure

$$\begin{aligned}
\mathbf{B}(\mathbf{g}) &= \sum_{k=1}^{N} \mu_k(\mathbf{g})\mathbf{B}_k \\
\mathbf{C}(\mathbf{g}) &= \sum_{k=1}^{N} \mu_k(\mathbf{g})\mathbf{C}_k \\
D(\mathbf{g}) &= \sum_{k=1}^{N} \mu_k(\mathbf{g})D_k
\end{aligned} \tag{18.28}$$

is a very simple way to generate a parametric macromodel.

18.4 Solving LMI's

18.4.1 Convex Programming

LMI's such as (18.7) are convex formulations and can always be solved by convex optimization [17], without needing ARE solvers and/or Hamiltonian matrices. A standard trick in convex optimization is to transform the problem that must be solved into an equivalent problem, which is in a standard form that can be solved by a generic solver. Recently developed parser-solvers, such as YALMIP [18], CVX [19], CVXMOD [20], and Pyomo [21] automate this reduction process. A general approach called disciplined convex programming [22, 23] has emerged as an effective methodology for organizing and implementing parser-solvers for convex optimization. In disciplined convex programming, the user combines built-in functions in specific, convexity-preserving ways. The constraints and objective must also follow certain rules. As long as the user conforms to these requirements, the parser can easily verify the convexity of the problem and automatically transform it to a standard form for transfer to the solver. Note that the parser-solvers CVX (which runs in Matlab) and CVXMOD (Python) use the disciplined convex programming approach. For example, in our case we solve LMI (18.7) by means of the CVX code

[2] Note that multilinear interpolation satisfies both positivity and partition of unity.

```
cvx_begin sdp
variable P(n,n) symmetric
P >= 0
[ A'*P+P*A P*B-C';
... B'*P-C -D-D'] <= 0
cvx_end
```

18.4.2 Riccati Equations

An LMI of the form (18.7), i.e.,

$$\begin{bmatrix} A^T P + PA & PB - C^T \\ B^T P - C & -D - D^T \end{bmatrix} \le 0 \tag{18.29}$$

can be solved by converting to the Lur'e equations [24]

$$\begin{aligned} A^T P + PA &= -Q^T Q \\ PB - C^T &= -Q^T W \\ W^T W &= D + D^T \end{aligned} \tag{18.30}$$

In the case $D + D^T > 0$, the matrices W and Q can be eliminated, yielding the algebraic Riccati equation

$$A^T P + PA + (PB - C^T)(D + D^T)^{-1}(B^T P - C) = 0 \tag{18.31}$$

Hence, if $D + D^T > 0$, the system is positive real if the algebraic Riccati Eq. (18.31) has a stabilizing solution P [25].
If $D + D^T$ is only semi-positive definite, i.e., $\det(D + D^T) = 0$, the situation is much more complicated and the approaches in [26, 27] may provide solutions. However, the Riccati approach may also be rescued by means of the following theorem:

Theorem 2 *Frequency inversion theorem : Let $H(s) = C(sI_n - A)^{-1}B + D$ be minimal and positive-real with A Hurwitz. Then $G(s) = \tilde{C}(sI_n - \tilde{A})^{-1}\tilde{B} + \tilde{D}$ with*

$$\begin{aligned} \tilde{A} &= A^{-1} \qquad \tilde{B} = A^{-1}B \\ \tilde{C} &= -CA^{-1} \qquad \tilde{D} = D - CA^{-1}B \end{aligned}$$

is also positive real and admits the same P matrix as $H(s)$.

Proof It is straightforward to see that when A is Hurwitz, then A^{-1} is also Hurwitz and vice versa. Also, it is simple to see by substitution that $G(s) = H(1/s)$. By positive-realness, $H(s)$ admits a factorization [24]

$$H(s) + H(-s)^T = M(-s)^T M(s) \quad \forall s \in \mathcal{C}_+ \tag{18.32}$$

Since the mapping $s \mapsto 1/s$ is one-to-one in (extended) $\mathcal{C}_+$, it follows that

$$\begin{aligned} G(s) + G(-s)^T &= H(1/s) + H(-1/s)^T \\ &= M(-1/s)^T M(1/s) \quad \forall s \in \mathcal{C}_+ \end{aligned}$$

In other words $G(s)$ is positive-real. To prove it admits the same P as $H(s)$ we write the Lur'e equations

$$\begin{aligned} A^T P + PA &= -Q^T Q \\ PB - C^T &= -Q^T W \\ D + D^T &= W^T W \end{aligned}$$

Define $\mathcal{Q} = -QA^{-1}$ and $\mathcal{W} = W - QA^{-1}B$. It is easy to see that

$$\tilde{A}^T P + P\tilde{A} = -\mathcal{Q}^T \mathcal{Q} \tag{18.33}$$

Also

$$-\mathcal{Q}^T \mathcal{W} = A^{-T}\left[Q^T W - Q^T Q A^{-1} B\right] = P\tilde{B} - \tilde{C}^T \tag{18.34}$$

and finally

$$\tilde{D} + \tilde{D}^T = \mathcal{W}^T \mathcal{W} \tag{18.35}$$

Note that $\tilde{D} = H(0)$ and hence Theorem 2 maps the positive-realness problem from $s = \infty$ to $s = 0$. Of course it could be that both $H(\infty) + H(\infty)^T$ and $H(0) + H(0)^T$ are singular, in which case the approach in [26] may provide solution.

18.5 Numerical Simulations

In the following examples, we show the importance of the realization issue, and validate the proposed Sylvester approach, by comparing them with the standard Gilbert and balanced realizations.

18.5.1 Two Coupled Microstrip with Variable Length

Two coupled microstrips can be modeled starting from per-unit-length parameters. The cross section is shown in Fig. 18.1.

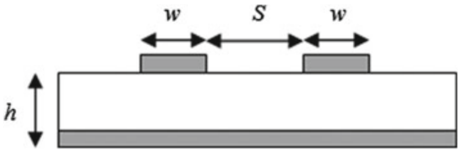


Fig. 18.1 Two coupled microstrip line

Table 18.1 Parameters of the coupled microstrips.

Parameter	Min	Max
Frequency ($freq$)	20 MHz	8 GHz
Length (L)	2.5 cm	3 cm

Figure 18.1 shows its cross section. The length L are considered as variable parameters in addition to frequency. Their corresponding ranges are shown in Table 18.1. The scattering parameters were obtained over a validation grid of 200 × 11 samples, for frequency and length respectively. We have built *root macromodels* for 6 values of the spacing by means of VF, each with an order 11.

As described in Sect. 18.3.4, a pivot matrix and a feedback matrix is chosen such that a well-conditioned solution is obtained for the Sylvester Eq. (18.21).

Also, since the eigenvalues of the pivot matrix and those of the *root macromodels* obtained from Gilbert realization must not be the same, we choose the poles very close to the imaginary axis as shown in Fig. 18.2.

The feedback matrix is chosen as column vectors of 1's, 2's and 0's similar to VF technique. A similarity transformation is then performed using the Sylvester solution to obtain the state-space matrices of the Sylvester realization.

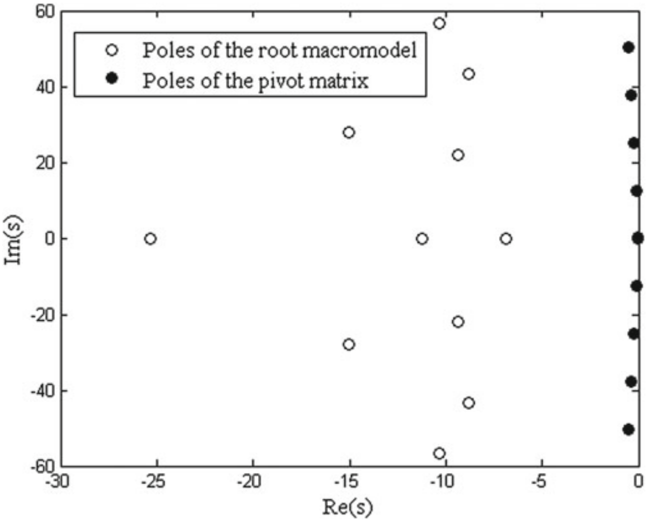


Fig. 18.2 Eigenvalues of the pivot matrix and the *root macromodels* obtained from Gilbert realization

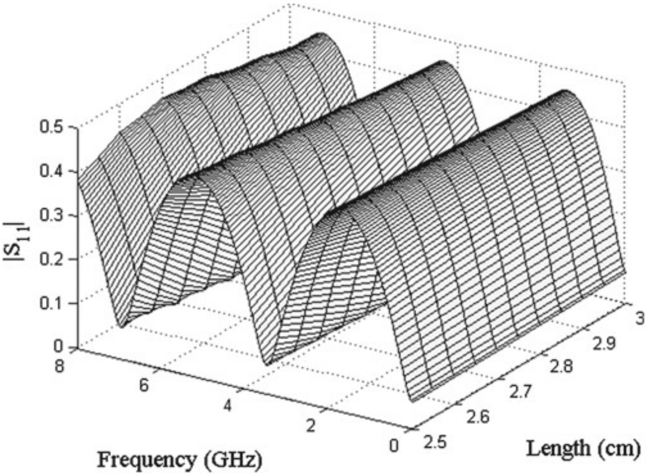


Fig. 18.3 Magnitude of the bivariate macromodel $S_{11}(s, L)$ (Sylvester realization for each *root macromodel*)

Next the realizations are converted to a passive descriptor state-space form using LMI (18.7) as described in Sect. 18.4 with the help of CVX. Finally, a bivariate macromodel is obtained by linear interpolation of the corresponding state-space matrices using the Sylvester realization as shown in Fig. 18.3.

The maximum absolute error over the validation grid for the parametric macromodel of the scattering matrix is bounded by −56 dB. Note that a very good agreement is obtained between the original data and the proposed parametric macromodeling technique. The parametric macromodel captures the behavior of the system very accurately over the entire range of the length.

Figure 18.4 shows that direct parameterization of the poles should be avoided due to potentially non-smooth behavior with respect to the design parameters with Gilbert realization.

In Fig. 18.5 it is shown that the maximum absolute error is very small for the Sylvester but it is not satisfactory for the Gilbert and balanced real realization.

18.5.2 Hairpin Bandpass Microwave Filter

In this example, a hairpin bandpass filter with the layout shown in Fig. 18.6 is modeled. The relative permittivity of the substrate is 9.9, and its thickness is equal to 0.635 mm.

The spacing S_1 and the length L of the stub are chosen as design variables in addition to frequency. Their corresponding ranges are shown in Table 18.2.

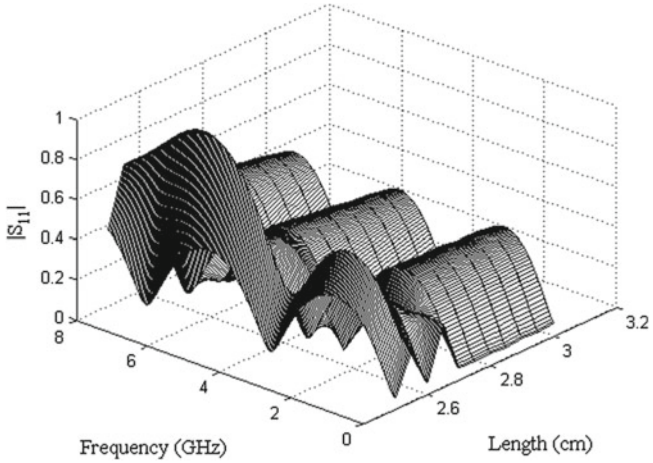


Fig. 18.4 Magnitude of the bivariate macromodel $S_{11}(s, L)$ (Gilbert realization form for each *root macromodel*)

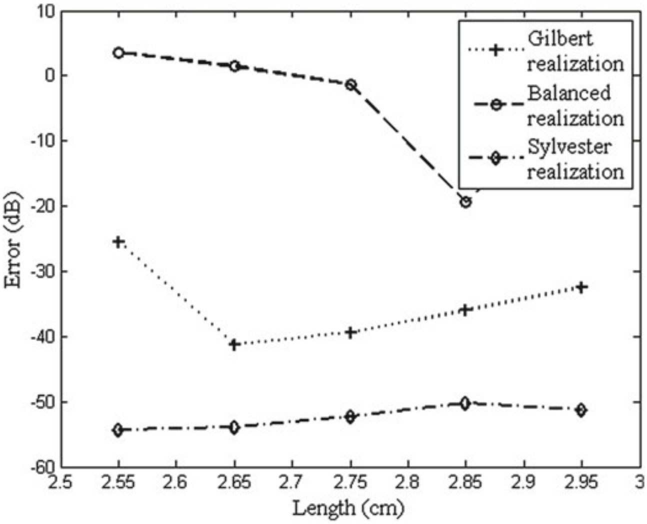


Fig. 18.5 CM: Absolute error comparison for the different realizations

The scattering parameters have been computed by means of the advanced design system (ADS) over a grid of 11 × 7 samples, for length and spacing respectively. We have built root *macromodels* for 6 × 4 values of the length and spacing respectively by means of VF, each with an order 13. Next the realization approaches as described in Sect. 18.3.3 is used to obtain Sylvester realized state-space form for each *root macromodel*. Then the realizations are converted to a passive descriptor state-space form using LMI (18.7) as described in Sect. 18.4 with the help of CVX.

Fig. 18.6 Layout of the folded stub notch filter

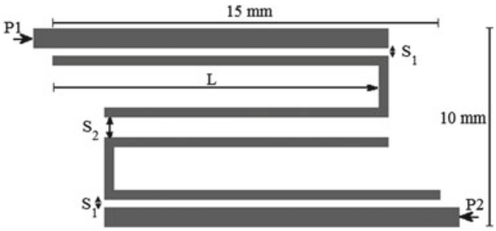


Table 18.2 Parameters of the hairpin bandpass microwave filter.

Parameter	Min	Max
Frequency $(freq)$	1.5 GHz	3.5 GHz
Length (L)	12 mm	12.5 mm
Spacing (S_1)	0.27 mm	0.32 mm

Finally, a trivariate macromodel is obtained by multilinear interpolation of the corresponding state-space matrices as shown in Fig. 18.7.
The maximum absolute error over the validation grid for the parametric macromodel of the scattering matrix is bounded by −58 dB. It can be noted that a very good agreement is obtained between the original data and the proposed parametric macromodeling technique. The parametric macromodel captures the behavior of the system very accurately over the entire design space.

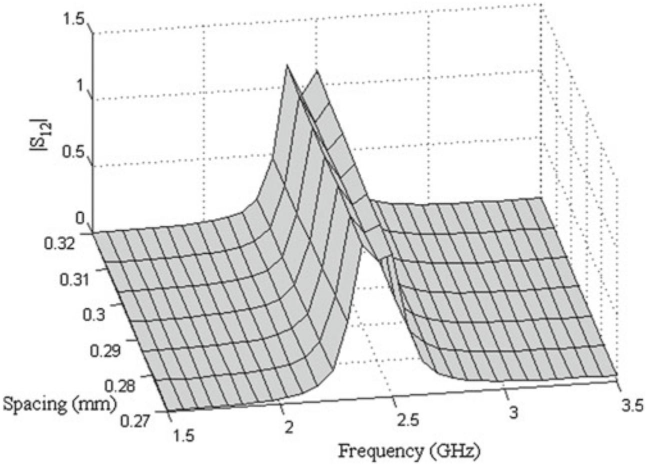


Fig. 18.7 Magnitude of the trivariate macromodel $S_{12}(s, L, S)$ for $L = 12.05$ mm (Sylvester realization for each *root macromodel*)

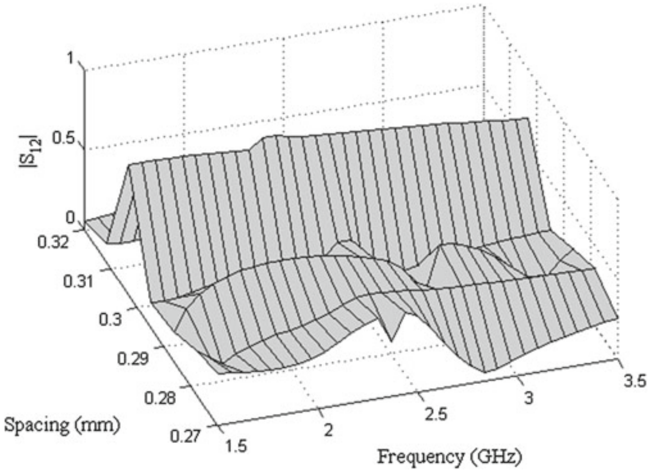


Fig. 18.8 Magnitude of the trivariate macromodel $S_{12}(s, L, S)$ for $L = 12.05$ mm (Balanced realization for each *root macromodel*)

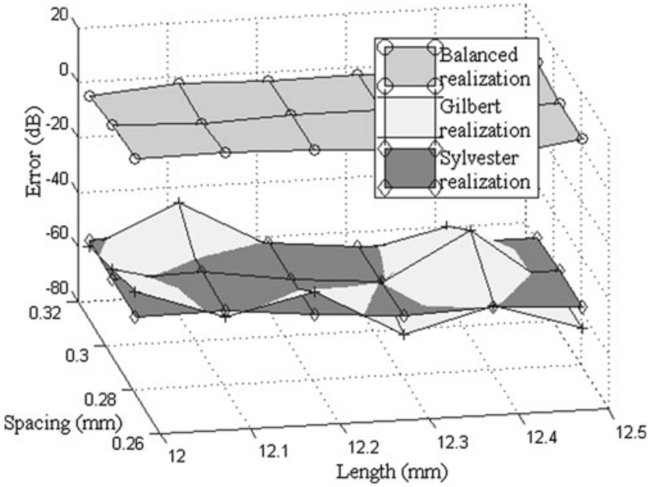


Fig. 18.9 Absolute error comparison for the different realizations

Figure 18.8 shows the parametric macromodel using balanced real realization. It is seen by comparing with Fig. 18.7 that the behavior is very erratic.

For the hairpin filter it can be also noted from the Fig. 18.9 that the maximum absolute error is very small for the Sylvester realization but it is not satisfactory for Gilbert realization and balanced real realization.

18.6 Conclusion

This paper proposes a novel state-space realization for parametric macromodeling based on interpolation of state-space matrices. A good choice of the state-space realization is required to account for the generally assumed smoothness of the state-space matrices with respect to the parameters. Suitable interpolation schemes along with Sylvester realization are used to interpolate a set of root state-space matrices in order to build accurate parametric macromodels. There are two essential aspects for this novel realization: (1) to find a proper pivot matrix and (2) to obtain a well-conditioned solution for a Sylvester equation. The numerical examples and related comparison results show that the proposed Sylvester realization provides very accurate parametric macromodel with a low computational cost. The properties of the system like stability and passivity can be preserved with the help of LMIs and by the use of proper interpolation schemes.

Acknowledgments This research has been funded by the Research Foundation Flanders (FWO) and the Interuniversity Attraction Poles Programme BESTCOM initiated by the Belgian Science Policy Office.

References

1. Ferranti, F., Knockaert, L,. Dhaene, T.: Guaranteed passive parameterized admittance-based macromodeling, IEEE Trans. Adv. Pack., **33**(3), 623–629 (2010)
2. Ferranti, F., Knockaert, L., Dhaene, T.: Parameterized S-parameter based macromodeling with guaranteed passivity. IEEE Microw. Wireless Compon Lett, **19**(10), 608610 (2009)
3. Ferranti, F., Knockaert, L., Dhaene, T., Antonini, G.: Passivity preserving parametric macromodeling for highly dynamic tabulated data based on Lure equations. IEEE Trans. Micro. Theor. Tech. **58**(12), 3688–3696 (2010)
4. Triverio, P., Nakhla, M., Grivet-Talocia, S.: Passive parametric modeling of interconnects and packaging components from sampled impedance, admittance or scattering data. Electron. Syst. Integr. Technol. Conf. pp. 16, Sept. (2010)
5. De Caigny, J., Camino, J.F., Swevers, J.: Interpolating model identication for siso linear parameter-varying systems. Mech. Syst. Signal Proc. **23**(8), 23952417 (2009)
6. Samuel, E.R., Knockaert, L., Ferranti, F., Dhaene, T.: Guaranteed passive parameterized macromodeling by using sylvester state-space realizations. IEEE Trans. Microwave Theor. Tech. **61**(4), 1444–1454 (April 2013)
7. Gilbert, E.G.: Controllability and observability in multi-variable control systems. SIAM J. Control **1**(2), 128151 (1963)
8. Moore, B.: Principal component analysis in linear systems: controllability, observability, and model reduction. IEEE Trans. Autom. Control **26**(1), 1731 (Feb. 1981)
9. Lovera, M., Mercere, G.: Identification for gain-scheduling: a balanced subspace approach. Am. Control Conf. 2007, pp. 858–863, (July 2007)
10. Anderson, B.D.O., Vongpanitlerd, S.: Network Analysis and Synthesis. NJ, Prentice-Hall, Englewood Cliffs (1973)
11. Ferranti, F., Knockaert, L., Dhaene, T., Antonini, G.: Parametric macromodeling for S-parameter data based on internal nonexpansivity, Int. J. Num. Model. Electron. Netw. Devices Fields **26**(1), 1527 (2013)

12. Ferranti, F., Knockaert, L., Dhaene, T.: Passivity-preserving parametric macromodeling by means of scaled and shifted state-space systems, IEEE Trans. Microwave Theor. Tech. **59**(10), 2394–2403, (Oct 2011)
13. De Souza, E., Bhattacharyya, S.P.: Controllability, observability and the solution of AX-XB = C. Lin. Alg. Appl. **39**, 167–188 (1981)
14. Varga, A.: Robust pole assignment via sylvester equation based state feedback parametrization. Proceedings IEEE International Symposium on Computer-Aided Control System Design, pp. 13–18, 2000
15. Gustavsen, B., Semlyen, A.: Rational approximation of frequency domain responses by vector fitting. IEEE Trans. Power Delivery **14**(3), 1052–1061 (July 1999)
16. Carvalho, J., Datta, K., Hong, Y.: A new block algorithm for full-rank solution of the Sylvester-observer equation. IEEE Trans. Autom. Control **48**(12), 2223–2228 (Dec. 2003)
17. Boyd, S., Vandenberghe, L.: Convex Optimization, Cambridge University Press, Cambridge, U.K., 2004. Available at http://www.math.nus.edu.sg/ mattohkc/sdpt3.html
18. Löfberg, J.: " YALMIP: a toolbox for modeling and optimization in MATLAB. Proceedings of CACSD Conference, Taipei, Taiwan, 2004. Available: http://control.ee.ethz.ch/joloef/yalmip.php
19. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming (web page and software), July 2008. Available: http://www.stanford.edu/boyd/cvx/
20. Mattingley, J.E., Boyd, S.: "CVXMOD: convex optimization software in Python (web page and software), Aug. 2008. Available: http://cvxmod.net/
21. Benavides, N.L., Carr, R.D., Hart, W.E.: Python optimization modeling objects (Pyomo). In: Proceedings of INFORMS Computing Society Conference, 2009. Available: https://software.sandia.gov/trac/pyutilib/export/30/trunk/doc/ pyomo.pdf
22. Grant, M., Boyd, S., Ye, Y.: Disciplined convex programming. In: Global optimization: from theory to implementation (Nonconvex Optimization and Its Applications), L. Liberti and N. Maculan, Eds. New York: Springer Science and Business Media, pp. 155–210, 2006
23. Mattingley, J., Boyd, S.: Real-time convex optimization in signal processing. IEEE Signal Proc. Mag. **27**(3), 50–61 (2010)
24. Curtain, R.F.: Old and new perspectives on the positive-real lemma in systems and control theory. Z. Angew. Math. Mech. **79**(9), 579–590 (1999)
25. Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: Linear matrix inequalities in system and control theory. SIAM Studies in Applied Mathematics, 15. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994
26. Weiss, H., Wang, Q., Speyer, J.L.: System characterization of positive real conditions. IEEE. Trans. Autom. Contr. **39**(3), 540–544 (1994)
27. Knockaert, L., Dhaene, T., Ferranti, F., De Zutter, D.: Model order reduction with preservation of passivity, non-expansivity and Markov moments. Syst Control Lett. **60**(1), 53–61 (Jan. 2011)

Chapter 19
Filtering for Stochastic Volatility by Using Exact Sampling and Application to Term Structure Modeling

ShinIchi Aihara, Arunabha Bagchi and Saikat Saha

Abstract The Bates stochastic volatility model is widely used in the finance problem and the sequential parameter estimation problem becomes important. By using the exact simulation technique, a particle filter for estimating stochastic volatility is constructed. The system parameters are sequentially estimated with the aid of parallel filtering algorithm with the new resampling procedure. The proposed filtering procedure is also applied to the modeling of the term structure dynamics. Simulation studies for checking the feasibility of the developed scheme are demonstrated.

Keywords Particle filter, Stochastic volatility, Parameter identification, Adaptive filter

19.1 Introduction

In the early 1960s, the linear filtering theory was formulated by Kalman and Bucy [12] and nonlinear filtering has already been well developed by many researchers, see Bensoussan [6] and the bibliography therein. The realization problem for the nonlinear filter is still not easy. The recent development of particle filtering theory [9] enables us to realize the nonlinear filtering in an easy way with the aid of the digital computer.

S. Aihara (✉)
Department of Computer Media, Tokyo University of Science Suwa, 5000-1 Toyohira,
Chino, Nagano, Japan
e-mail: aihara@rs.suwa.tus.ac.jp

A. Bagchi
Department of Applied Mathematics, University of Twente, P.O. Box 217, Ensched 7500 AE,
The Netherlands
e-mail: a.bagchi@utwente.nl

S. Saha
Department of Electrical Engineering, Linköping University, Link öping SE-58183, Sweden
e-mail: saha@isy.liu.se

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_19

In this paper we consider the Bates model that is used in the fiance problem. In this model, we observe the tick value of stock price and need to estimate the movement of the volatility process for trading the stock and/or options. It is not possible to apply the nonlinear filtering theory to this volatility estimation problem, because this is not covered by the usual filtering problem in the continuous stochastic systems [1]. To circumvent this difficulty, the particle filter theory is usually applied in [3, 8, 10]. The Bates model is given by

$$dS_t = \mu_S S_t dt + \sqrt{v_t} S_t dB_t + S_t dZ_t^J - \lambda m^J S_t dt, \tag{19.1}$$

$$dv_t = \kappa(\theta - v_t)dt + \xi\sqrt{v_t} dZ_t \tag{19.2}$$

where B_t and Z_t are standard Brownian motion processes with correlation ρ and Z_t^J denotes the pure-jump process. Noting that the process S_t denotes the stock value, the observation data $y_t = \log S_t/S_0$ is given by

$$dy_t = (\mu_S - \lambda m^J - \frac{1}{2}v_t)dt + \sqrt{v_t} dB_t + dq_t^J, \tag{19.3}$$

where q_t^J is a compound Poisson process with intensity λ and Gaussian distribution of jump size, i.e., $N(\mu_J, \sigma_J^2)$, and the mean relative jump size is given by $m^J = E(\exp(U^s) - 1) = \exp(\mu_J + \sigma_J^2/2) - 1$ and where the $\lambda m^J S_t$ term in (19.1) compensates for the instantaneous change in expected stock introduced by the pure-jump process Z_t^J. The particular properties of this model are

1. The observation mechanism (19.3) contains the signal dependent noise.
2. The observation noise has a correlation with the system noise.

From the first property, the estimation of stochastic volatility becomes a non-standard filtering problem. To circumvent this difficulty, all systems are discretized and the particle filter is applied in [3, 11]. However the usual discretization method transforms the original continuous non-Gaussian system into the conditional Gaussian. Recently, Brodie and Kaya [7], van Haastrecht and Pelsser [14] and Smith [13] proposed the exact simulation method from the fact that the original system has a non-central chi-square distribution and we use this technique to the particle filtering [4]. Introducing the new Brownian motion

$$\tilde{Z}_t = \frac{1}{\sqrt{1-\rho^2}}(Z_t - \rho B_t), \tag{19.4}$$

from the second property, (19.2) becomes

$$\begin{aligned} dv_t &= \kappa(\theta - v_t)dt + \xi\sqrt{v_t}\sqrt{1-\rho^2} d\tilde{Z}_t \\ &\quad + \xi\rho(dy_t - (\mu_S - \lambda m^J - \frac{1}{2}v_t)dt - dq_t^J). \end{aligned} \tag{19.5}$$

Although in [4] the exact particle filtering procedure has been applied to the systems (19.5) and (19.3), the priority property of the exact simulation method can not be guaranteed and the final parameter estimation results are not satisfactory. In this paper, we introduce the rejection and acceptance method for utilizing the exact simulation method and construct the parallel filtering algorithm with a new resampling procedure for parameter identification.

In Sect. 19.2, we review the exact particle filtering with the new use of rejection and resampling procedure. For the real application to the finance problem, the market price of risk terms are included in the original dynamics in Sect. 19.3. The new parallel filter algorithm is developed for estimating the systems unknown parameters sequentially in Sect. 19.4. The proposed method is also applied to the term structure modeling in Sect. 19.5. Finally some simulation studies for parameter estimation are presented in Sect. 19.6.

19.2 Exact Particle Filtering

19.2.1 Exact Sampling

In order to perform the particle filter, the original system is usually approximated to the discrete-time one by using the Euler method. This approximation easily causes bias from the original continuous system. For example, the discrete-time volatility process v_k often has negative values. To avoid this bias, we propose the exact sampling method which is developed by Broadie and Kaya [7], Smith [13] and van Haastrecht and Pelsser [14] for simulating the Heston process. In this paper, from (19.5) we can obtain the optimal importance function $p(v_{t_2}|v_{t_1}, y_{t_2}, y_{t_1})$. Hence we generate samples from this optimal importance function. Now we shall present the exact sampling procedure. For simplicity we consider the time interval $t_1 < t_2$ and set the following assumption: At most one jump occurs in this time interval and we observe y_{t_2} and y_{t_1}.

19.2.1.1 Exact Sampling from $p(v_{t_2}|v_{t_1}, y_{t_2}, y_{t_1})$

From (19.1), the volatility process v_{t_2} is represented by

$$v_{t_2} = \tilde{v}_{t_1} + \int_{t_1}^{t_2} \tilde{\kappa}(\tilde{\theta} - v_s)ds + \int_{t_1}^{t_2} \xi\sqrt{v_s}\sqrt{1-\rho^2}d\tilde{Z}_s, \tag{19.6}$$

where

$$\tilde{v}_{t_1} = v_{t_1} + \rho\xi\{y_{t_2} - y_{t_1} - (\mu_S - \lambda m^J)(t_2 - t_1) - \Delta q^i_{t_1}\}$$
$$\tilde{\kappa} = \kappa - \frac{\rho\xi}{2}$$

$$\begin{aligned}\tilde{\theta} &= \frac{\kappa\theta}{\tilde{\kappa}} \\ \Delta q^i_{t_1} &= \text{jump sample from } q^J_t \text{ for } t_1 < t < t_2.\end{aligned}$$

Now assuming that $\tilde{v}_{t_1} \geq 0$, we find that the transition law of v_{t_2} given by v_{t_1}, y_{t_1} and y_{t_2} is expressed as the non-central chi-square random variable $\chi^2_d(\lambda_\chi)$ with d degrees of freedom and non-centrality parameter λ_χ,

$$\frac{\xi^2(1-\rho^2)(1-e^{-\tilde{\kappa}(t_2-t_1)})}{4\tilde{\kappa}}\chi^2_d(\lambda_\chi), \tag{19.7}$$

where

$$d = \frac{4\tilde{\theta}\tilde{\kappa}}{\xi^2(1-\rho^2)}$$

and

$$\lambda_\chi = \frac{4\tilde{\kappa}e^{-\tilde{\kappa}(t_2-t_1)}}{\xi^2(1-\rho^2)(1-e^{-\tilde{\kappa}(t_2-t_1)})}\tilde{v}_{t_1}.$$

Hence by using MATLAB code “ncx2rnd.m”, we can get a sample v_{t_2} .

For the case that $\tilde{v}_{t_1} < 0$, this event may occur when v_{t_1} is very small in generating particles by using the data $y_{t_2} - y_{t_1}$. In the real world, we have already obtained the value $y_{t_2} - y_{t_1}$. Hence v_{t_1} should satisfy

$$v_{t_1} + \rho\xi\{y_{t_2} - y_{t_1} - (\mu_S - \lambda m^J)(t_2 - t_1) - \Delta q^i_{t_1}\} + \tilde{\kappa}\tilde{\theta}(t_2 - t_1) \geq 0. \tag{19.8}$$

19.2.1.2 $\tilde{v}_{t_1} < 0$ Case

We use the rejection and resampling procedure. At the time t_1, we already get many particles say $v^{(i)}_{t_1}$. Hence we check the above inequality (19.8) for each $v^{(i)}_{t_1}$. If the particles which do not satisfy (19.8) are found, we ignore these and perform a resampling procedure.

19.2.2 Construction of Probability Density Function

If we use the Euler scheme for discretization, the generated sample becomes the conditionally Gaussian. However for the exact sampling scheme, the processes generated are governed by the non-central chi-square distribution. Although the explicit function form of this distribution is not possible, we can numerically evaluate the pdf by using the MATLAB code, “ncx2pdf.m”.

- $p(v_{t_2}|v_{t_1}, y_{t_2}, y_{t_1})$ form
  Noting that the jump occurs at most one time during the time interval $[t_1, t_2]$, i.e., the probability that the jump occurs is $\lambda e^{-\lambda(t_2-t_1)}$ and no jump becomes $1-\lambda e^{-\lambda(t_2-t_1)}$, and the jump size $U^s_{\cdot}$ is Gaussian with mean μ_J and variance σ_J^2, we have

$$
\begin{aligned}
&p(v_{t_2}|v_{t_1}, y_{t_2}, y_{t_1})\\
&\quad = (1-e^{-\lambda(t_2-t_1)}\lambda(t_2-t_1))p(v_{t_2}|v_{t_1}, y_{t_2}, y_{t_1}, \Delta q^j_{t_1}=0)\\
&\qquad + e^{-\lambda(t_2-t_1)}\lambda(t_2-t_1)\int_{-\infty}^{\infty} p(v_{t_2}|v_{t_1}, y_{t_2}, y_{t_1}, U^s)\frac{1}{\sqrt{2\pi\sigma_J^2}}\\
&\qquad\quad \exp(-\frac{(U^s-\mu_J)^2}{2\sigma_J^2})dU^s\\
&\quad = (1-e^{-\lambda(t_2-t_1)}\lambda(t_2-t_1))\,\mathrm{pdf\ of}\left\{\frac{\xi^2(1-\rho^2)(1-e^{-\tilde{\kappa}(t_2-t_1)})}{4\tilde{\kappa}}\chi_d^2(\tilde{\lambda}_\chi)\right\}\\
&\qquad + e^{-\lambda(t_2-t_1)}\lambda(t_2-t_1)\times\int_{-\infty}^{\infty}\mathrm{pdf\ of}\left\{\frac{\xi^2(1-\rho^2)(1-e^{-\tilde{\kappa}(t_2-t_1)})}{4\tilde{\kappa}}\right.\\
&\qquad \left.\times\ \chi_d^2(\tilde{\lambda}_\chi-\frac{4\tilde{\kappa}e^{-\tilde{\kappa}(t_2-t_1)}\rho}{\xi(1-\rho^2)(1-e^{-\tilde{\kappa}(t_2-t_1)})}U^s)\right\}\frac{1}{\sqrt{2\pi\sigma_J^2}}\\
&\qquad\quad \exp(-\frac{(U^s-\mu_J)^2}{2\sigma_J^2})dU^s
\end{aligned}
\tag{19.9}
$$

  where

$$
\tilde{\lambda}_\chi = \frac{4\tilde{\kappa}e^{-\tilde{\kappa}(t_2-t_1)}}{\xi^2(1-\rho^2)(1-e^{-\tilde{\kappa}(t_2-t_1)})}\times\{v_{t_1}+\rho\xi\{y_{t_2}-y_{t_1}-(\mu_S-\lambda m^J)(t_2-t_1)\}\}
$$

  In (19.9), the first term implies that we have no jump and the second term is caused due to the jump size $U^s \in N(\mu^J, \sigma_J^2)$.
- $p(v_{t_2}|v_{t_1}, y_{t_1})$ form
  It follows from (19.2) that

$$
p(v_{t_2}|v_{t_1}, y_{t_1}) = \mathrm{pdf\ of}\ \frac{\xi^2(1-e^{-\kappa(t_2-t_1)})}{4\kappa}\chi_{\tilde{d}}^2(\lambda_\chi^v),
$$

  where

$$
\tilde{d} = \frac{4\theta\kappa}{\xi^2},
$$

and

$$\lambda_\chi^v = \frac{4\kappa e^{-\kappa(t_2-t_1)}}{\xi^2(1-e^{-\kappa(t_2-t_1)})} v_{t_1}.$$

In this case, from

- $p(y_{t_2}|y_{t_1}, \int\limits_{t_1}^{t_2} v_s ds)$ form

$$dy_t = (\mu_S - \lambda m^J - \frac{1}{2}v_t)dt + \frac{\rho}{\xi}(dv_t - \kappa(\theta - v_t)dt)$$
$$+ \sqrt{1-\rho^2}\sqrt{v_t}d\tilde{Z}_t + dq_t^J$$

we easily get

$$\begin{aligned}
&p(y_{t_2}|y_{t_1}, \int\limits_{t_1}^{t_2} v_s ds) \\
&= \frac{1 - e^{-\lambda(t_2-t_1)}\lambda(t_2-t_1)}{\sqrt{2\pi(1-\rho^2)\int_{t_1}^{t_2} v_s ds}} \\
&\quad \times \exp[-\frac{1}{2(1-\rho^2)\int_{t_1}^{t_2} v_s ds}\{y_{t_2} - [y_{t_1} + (\mu_S - \lambda m^J - \frac{\kappa\rho\theta}{\xi})(t_2-t_1) \\
&\quad - (\frac{1}{2} - \frac{\kappa\rho}{\xi})\int\limits_{t_1}^{t_2} v_s ds + \frac{\rho}{\xi}(v_{t_2} - v_{t_1})]\}^2] \\
&\quad + \frac{e^{-\lambda(t_2-t_1)}\lambda(t_2-t_1)}{\sqrt{2\pi((1-\rho^2)\int_{t_1}^{t_2} v_s ds + \sigma_J^2)}} \exp[-\frac{1}{2((1-\rho^2)\int_{t_1}^{t_2} v_s ds + \sigma_J^2)} \\
&\quad \{y_{t_2} - [y_{t_1} + (\mu_S - \lambda m^J - \frac{\kappa\rho\theta}{\xi})(t_2-t_1) - (\frac{1}{2} - \frac{\kappa\rho}{\xi})\int\limits_{t_1}^{t_2} v_s ds \\
&\quad + \mu_J + \frac{\rho}{\xi}(v_{t_2} - v_{t_1})]\}^2].
\end{aligned} \tag{19.10}$$

19.2.3 Exact Particle Filter Algorithm

Now we can perform the exact particle filter. The weight $w_{\cdot}^{(i)}$ is given by the following recursive form: for $i = 1, \ldots, N$ and $k = 1, \ldots, m$

$$w_{t_k}^{(i)} = w_{t_{k-1}}^{(i)} \frac{p(y_{t_k}|y_{t_{k-1}}, \int_{t_{k-1}}^{t_k} v_s^{(i)} ds) p(v_{t_k}^{(i)}|v_{t_{k-1}}^{(i)})}{p(v_{t_k}^{(i)}|v_{t_{k-1}}^{(i)}, y_{t_k}, y_{t_{k-1}})}. \tag{19.11}$$

Of course we need to perform the resampling scheme in the above filtering algorithm.

It is also possible to construct the smoothing algorithm by using forward filtering-backward sampling scheme of Doucet et. al. [9]

Algorithm. (Sample Realization.)

- Run the particle filter to obtain $(v_{t_k}^{(i)}, \omega_{t_k}^{(i)})_{1 \le k \le m, 1 \le i \le N}$.
- By using the systematic resampling method, we generate new index J_m from $\{\omega_{t_m}^{(i)}\}$ for $i = 1, 2, \dots, N$.
- Set $\tilde{v}_{t_m}^{(i)}$ as $v_{t_m}^{J_m}$.
- For $k = m - 1$ to 1;
  - Resample to get J_k from $\{\omega_{t_k}^{(i)} p(\tilde{v}_{t_k+1}^{(i)}|v_{t_k}^{(i)})\}_{1 \le i \le N}$.
  - Set $\tilde{v}_{t_k}^{(i)}$ as $v_k^{J_k}$.
- $\tilde{v}^{(i)} = [\tilde{v}_{t_1}^{(i)}, \tilde{v}_{t_2}^{(i)}, \dots, \tilde{v}_{t_m}^{(i)}]$ is the realized particles for smoothing with $1/N$ probability.

19.3 Market Price of Risk

For estimation we also need the dynamics of the state S_t and v_t under the actual probability measure $\mathcal{P}$. We specify the market price of risk for B_t and Z_t as

$$d\begin{pmatrix} B_t^{\mathcal{P}} \\ Z_t^{\mathcal{P}} \end{pmatrix} = d\begin{pmatrix} B_t \\ Z_t \end{pmatrix} - \begin{pmatrix} \lambda_S\sqrt{v_t} & 0 \\ 0 & \lambda_v\sqrt{v_t} \end{pmatrix} dt.$$

We ignore the jump-timing risk premium and the jump-size risk is assumed to be included in μ_J. Now the dynamics of y_t and v_t under $\mathcal{P}$ is given by

$$\begin{aligned} dy_t &= (\mu_S - \lambda m^J + (\lambda_S - \frac{1}{2})v_t)dt + \sqrt{v_t} dB_t^{\mathcal{P}} + dq_t^J \\ dv_t &= (\kappa\theta - (\kappa - \lambda_v \xi)v_t)dt + \xi\sqrt{v_t}\sqrt{1-\rho^2} d\tilde{Z}_t^{\mathcal{P}} \\ &\quad + \xi\rho(dy_t - (\mu_S - \lambda m^J - (\frac{1}{2} - \lambda_S)v_t)dt - dq_t^J). \end{aligned}$$

Hence it is possible to apply our particle filter algorithm to this world measure dynamics. The corresponding dynamics is transformed to

$$v_{t_2} = \tilde{v}_{t_1} + \int_{t_1}^{t_2} \tilde{\kappa}(\tilde{\theta} - v_s) ds + \int_{t_1}^{t_2} \xi\sqrt{v_s}\sqrt{1-\rho^2} d\tilde{Z}_s,$$

where

$$\begin{aligned}\tilde{v}_{t_1} &= v_{t_1} + \rho\xi\{y_{t_2} - y_{t_1} - (\mu_S - \lambda m^J)(t_2 - t_1) - \Delta q^i_{t_1}\}\\ \tilde{\kappa} &= \kappa - \frac{\rho\xi}{2} + \xi(\rho\lambda_S - \lambda_v)\\ \tilde{\theta} &= \frac{\kappa\theta}{\tilde{\kappa}}.\end{aligned}$$

$p(v_{t_2}|v_{t_1})$ becomes

$$p(v_{t_2}|v_{t_1}) = \text{pdf of } \frac{\xi^2(1 - e^{-(\kappa-\xi\lambda_v)(t_2-t_1)})}{4(\kappa - \xi\lambda_v)}\chi^2_d(\lambda^v_\chi), \tag{19.12}$$

where

$$d = \frac{4\theta\kappa}{\xi^2},$$

and

$$\lambda^v_\chi = \frac{4(\kappa - \xi\lambda_v)e^{-(\kappa-\xi\lambda_v)(t_2-t_1)}}{\xi^2(1 - e^{-(\kappa-\xi\lambda_v)(t_2-t_1)})}v_{t_1}.$$

Noting that

$$\begin{aligned}dy_t = (\mu_S - \lambda m^J + (\lambda_S - \frac{1}{2})v_t)dt + \frac{\rho}{\xi}(dv_t - \kappa\theta dt + (\kappa - \lambda_v\xi)v_t)dt)\\ + \sqrt{1-\rho^2}\sqrt{v_t}d\tilde{Z}_t + dq^J_t,\end{aligned}$$

we also have

$$\begin{aligned}p(y_{t_2}|y_{t_1}, \int_{t_1}^{t_2} v_s ds) = &\frac{1 - e^{-\lambda(t_2-t_1)}\lambda(t_2 - t_1)}{\sqrt{2\pi(1-\rho^2)\int_{t_1}^{t_2} v_s ds}}\exp[-\frac{1}{2(1-\rho^2)\int_{t_1}^{t_2} v_s ds}\{y_{t_2}\\ &- [y_{t_1} + (\mu_S - \lambda m^J - \frac{\kappa\rho\theta}{\xi})(t_2 - t_1)\\ &- (\frac{1}{2} - \frac{\kappa\rho}{\xi} + \rho\lambda_v - \lambda_S)\int_{t_1}^{t_2} v_s ds\\ &+ \frac{\rho}{\xi}(v_{t_2} - v_{t_1})]\}^2] + \frac{e^{-\lambda(t_2-t_1)}\lambda(t_2 - t_1)}{\sqrt{2\pi((1-\rho^2)\int_{t_1}^{t_2} v_s ds + \sigma^2_J)}}\\ &\times \exp[-\frac{1}{2((1-\rho^2)\int_{t_1}^{t_2} v_s ds + \sigma^2_J)}\end{aligned}$$

$$\{y_{t_2} - [y_{t_1} + (\mu_S - \lambda m^J - \frac{\kappa\rho\theta}{\xi})(t_2 - t_1) - (\frac{1}{2} - \frac{\kappa\rho}{\xi} + \rho\lambda_v - \lambda_S)\int_{t_1}^{t_2} v_s ds + \mu^J + \frac{\rho}{\xi}(v_{t_2} - v_{t_1})]\}^2].$$

19.4 Parallel Filtering Algorithm

In a market, traders buy or sell stocks from their feeling of the volatility movement of the traded stock. Form this fact, we need to estimate the volatility itself rather than the parameters in the model. The estimate of the volatility should be online. Hence, in this section, we construct the recursive online estimate for the volatility. Of course to obtain the estimate of volatility, we also get the estimate of systems parameters at the same time. Here the unknown parameters are denoted by

$$\alpha = [\kappa, \theta, \nu_S, \rho, \xi, \lambda, \nu^J, \sigma^J, \lambda_v, \lambda_S].$$

Now we set candidates of unknown parameter α such that

$$\alpha^{(j)} \in \text{uniformly random vectors in } \Theta,\ j = 1, \dots, M_p$$

If we know an a-priori information for Θ, we may set the pdf $p_o(\alpha)$. For each $\alpha^{(j)}$, we solve the particle filter $\hat{v}_{t_k}(\alpha^{(j)})$ from Sect. 19.2.3. Hence from [5], we get the posteriori density given by

$$p(\alpha^{(j)}|y_{t_0:t_k}) = \frac{\{\Sigma_{i=1}^N w_{t_{k-1}}^{(i)}(\alpha^{(j)}) LF_{k,i}\} p(\alpha^{(j)}|y_{t_0:t_{k-1}})}{\Sigma_{j=1}^{M_p}\left\{\{\Sigma_{i=1}^N w_{t_{k-1}}^{(i)}(\alpha^{(j)}) LF_{k,i}\} p(\alpha^{(j)}|y_{t_0:t_{k-1}})\right\}}$$

where

$$LF_{k,i} = p(y_k | y_{k-1}, \int_{t_{k-1}}^{t_k} v_s^{(i)}(\alpha^{(j)}) ds).$$

The estimates of volatility and parameters are given by

$$\hat{v}_k = \sum_{j=1}^{M_p} \hat{v}_{t_k}(\alpha^{(j)}) p(\alpha^{(j)}|y_{t_0:t_k}) \tag{19.13}$$

$$\hat{\alpha}_k = \sum_{j=1}^{M_p} \alpha^{(j)} p(\alpha^{(j)}|y_{t_0:t_k}). \tag{19.14}$$

19.4.1 New Resampling Procedure

The sample of parameter $\{\alpha^{(j)}\}_{j=1}^{M_P}$ is drawn only from the initial information (in this paper we set the uniform distribution). Hence for a long time period the estimates of parameters are sometimes stacked with some biases. This may be caused by the fact that there are so many unknown parameters while we get a scaler observation data. In order to improve this property, a resampling for the candidates for parameters $\alpha^{(j)}$ is usually performed in MCMC algorithm in [11]. In the parallel filtering algorithm, we already get the posterior probability $p(\alpha^{(j)}|y_{t_0:t_k})$ and from this distribution, we propose to get new samples for $\alpha^{(j)}$ by using the following procedure:

1. We set the resampling time t_p^r if

$$(\sum_{j=1}^{M_P} p^2(\alpha^{(j)}|y_{t_0:t_p^r}))^{-1} \leq \frac{2M_P}{3},$$

   we generate new sample $\alpha^{(j)}$ from the step (2) to (6).
2. Calclulate

$$\hat{\alpha}_{t_p^r} = \sum_{j=1}^{M_p} \alpha^{(j)} p(\alpha^{(j)}|y_{t_0:t_p^r})$$

$$\hat{\sigma}_{\alpha(i)} = \sum_{j=1}^{M_p} (\alpha^{(j)}(i))^2 p(\alpha^{(j)}(i)|y_{t_0:t_k^r}) - (\hat{\alpha}_{t_p^r}^{(j)}(i))^2$$

   for $i = 1, 2, \ldots, 10$.
3. We denote the parameter range at the resampling time point $t = t_p^r$ as

$$\text{lb}(i, t_p^r) \leq \alpha(i) \leq \text{ub}(i, t_p^r) \text{ for } i = 1, 2, \ldots, 10,$$

   where $\text{lb}(i, t) = \text{lb}(i)$ and $\text{ub}(i, t) = \text{ub}(i)$ for $t <$ the first resampling time.
4. From the calculated $\hat{\alpha}_{t_p^r}$ and $\hat{\sigma}_\alpha$, we reset the parameter range from t_{p-1}^r as

$$\text{lb}(i, t_p^r) = \max(\text{lb}(i, t_{p-1}^r), \hat{\alpha}_{t_k^r}(i) - 3\hat{\sigma}_{\alpha(i)})$$

   and

$$\text{ub}(i, t_p^r) = \min(\text{ub}(i, t_{p-1}^r), \hat{\alpha}_{t_p^k} + 3\hat{\sigma}_{\alpha(i)}).$$

5. Construct the candidates of parameter k;

$$\alpha_k(i) = \text{lb}(i, t_p^r) + \frac{\text{ub}(t_p^r) - \text{ub}(t_p^r)}{M_p - 1}(i - 1).$$

   for $k = 1, 2, \ldots, M_p$.

6. Construct the posterior distribution for each parameter $\alpha(i)$ by using the Gaussian approximation:

$$P(\alpha(i)|y_{t_0:t_p^r}) \sim \mathcal{N}(\alpha(i); \hat{\alpha}_{t_p^r}, \epsilon_i \hat{\sigma}_{\alpha(i)}^{1/2}),$$

where ϵ_i is a user defined parameter to increase diversity.

7. Allocate n_i copies of the particle $\alpha_k(i)$ from

$$n_i = \text{the number of } \frac{(k-1)+\tilde{u}}{M_p} \in (F_G(\alpha_{k-1}(i)), F_G(\alpha_k(i))]$$

for $\tilde{u}$= uniform random number, where F_G is an approximated Gaussian distribution (step 6):

$$F_G(\alpha_k(i)) = \frac{\int_{-\infty}^{\alpha_k(i)} \frac{1}{\sqrt{2\pi}\epsilon_i \hat{\sigma}_{\alpha(i)}} \exp[-\frac{1}{2(\epsilon_i \hat{\sigma}_\alpha(i))^2}(\alpha(i) - \hat{\alpha}_{t_p^r})^2] d\alpha(i)}{F_G(\text{lb}(i, t_p^r))}.$$

8. Construct new candidate; for $j = 1, 2, \ldots, M_P$

$$\alpha^{(j)} = [\alpha^j(1), \alpha^j(2), \ldots, \alpha^j(10)].$$

9. Reset $p(\alpha^{(j)}|y_{y_0:t_p^r}) = 1/M_P$.

19.5 Application to the Term Structure Modeling

We choose the short rate model as $r_t = \log S_t$. For simplicity we neglect the jump term. According to the discussions in [2], the bond price is well modeled by

$$P(t,T) = \exp\{-\int_0^{T-t} f(t,x)\},$$

where T denotes the maturity and

$$df(t,x) = \frac{\partial f(t,x)}{\partial x}dt + \{v_t e^{\mu_S x} \int_0^x e^{\mu_S z} dz + \sigma \int_0^x q(x,z)dz\}dt$$
$$+ \lambda_S v_t e^{\mu_S x} dt + \sqrt{v_t} e^{\mu_S x} dB_t + \sigma dw(t,x), \qquad (19.15)$$

where $w(t,x)$ is a Brownian motion process and denotes a small perturbation from r_t and $E\{w(t,x)w(t,z)\} = q(x,z)t$. We only consider the market price of risk from

B_t as λ_S. In this situation, the yield curve data are observed;

$$Y(t, T-t) = -\frac{1}{T-t}\log P(t, T).$$

Setting the time-to-maturity $\tau = T - t$ as constant, we obtain the observation data $\tilde{Y}(t) = [Y(t, \tau_i)]_{m\times 1}$ for $\tau_i, i = 1, 2, \dots, m$. From the market, we observe the spot price $y_t = \log S_t$ but this is not exactly a spot price, e.g., a day ahead implicit auction market for the electricity market. Usually this spot data is not used. However regarding this data as the exact spot data, we can directly apply the algorithm in Sects. 19.3 and 19.4 to estimate the volatility v_t and also get $E\{f(t,x)|\tilde{Y}(t), y_t, v_t\}$ as the output of the Kalman filter derived in [2]. This is a so-called "Rao-Blackwellised particle filter". We are interested in estimating

$$p(f(t,x), v_t|\mathcal{Y}_t) = p(v_t|\mathcal{Y}_t) \times p(f(t,x)|\mathcal{Y}_t, v_t),$$

where $\mathcal{Y}_t = \sigma\{\tilde{Y}(s), 0 \le s \le t\}$. Noting that

$$p(f(t,x)|\mathcal{Y}_t, v_t) = \text{Gaussian distribution } \mathcal{N}(E\{f(t,x)|\mathcal{Y}_t, v_t\}, \mathbf{P})$$

where $E\{f(t,x)|\mathcal{Y}_t, v_t\}$ and its covariance $\mathbf{P}$ are outputs of the Kalman filter in [2]. Hence we only need to use particle filter method to realize $p(v_t|\mathcal{Y}_t)$. To do this the key is the fact that the likelihood $p(\tilde{Y}(t)|v_s, 0 \le s \le t)$ is also a Gaussian distribution which is constructed by the Kalman filter output. Hence the exact sampling algorithm developed here is applicable to realize $p(v_t|\mathcal{Y}_t)$. The key point is to choose the importance function for generating $v_t^{(i)}$. We use the generating method stated in Sects. 19.2 and 19.3. This implies that the importance function $p(v_{t_2}|v_{t_1}, y_{t_1}, y_{t_2})$ given by (19.12) is not optimal and the data y_t is only used fro exact sampling. The estimate is performed under $\mathcal{Y}_t = \sigma\{\tilde{Y}_s, 0 \le s \le t\}$. Hence

$$p(v_{t_2}|\mathcal{Y}_{t_2}) \propto \frac{p(\tilde{Y}_{t_2}|\tilde{Y}_{t_1}, v_{t_1}, v_{t_2})p(v_{t_2}|v_{t_1})}{p(v_{t_2}|v_{t_1}, y_{t_1}, y_{t_2})}, \tag{19.16}$$

where $p(v_{t_2}|v_{t_1})$ is given by (19.12). In practice, we need not store $v_{t_j}^{(i)}$ for all j but only $v_{t_j}^{(i)}, v_{t_{j-1}}^{(i)}$ and the Kalman filter statistics associated with $v_{t_j}^{(i)}$.

19.6 Simulation Studies

We set the following parameters in Table 19.1.

The lower and upper bounds for parameters are set as Here we set $dt = 0.001$, $T = 1, M = 100, M_P = 60$ and $t_r = 20dt, \epsilon_1 = 1.1, \epsilon_2 = 1.1, \epsilon_3 = 1.15$, $\epsilon_4 = 1.01, \epsilon_5 = 1.01, \epsilon_6 = 1.15, \epsilon_7 = 1.15, \epsilon_8 = 1.15, \epsilon_9 = 1.15$, and $\epsilon_{10} = 1.15$.

In Fig. 19.1, we show the true volatility state and compound Poisson process. The observed log price is also shown in Fig. 19.2. The estimated volatility is shown in Fig. 19.3 with the square error in Fig. 19.4.

Table 19.1 Model parameters

	κ	θ	μ_S	ρ	ξ	λ	μ^J	σ^J	λ_v	λ_S
True	0.8638	1.1000	0.6000	−0.1500	2.1017	5.4000	−0.3000	0.2500	0.1882	0.1723

Table 19.2 Lower and upper bounds of model parameters

	κ	θ	μ_S	ρ	ξ	λ	μ^J	σ^J	λ_v	λ_S
Upper	1.6412	2.0900	1.140	−0.001	3.9932	10.260	−0.0030	0.4570	0.35758	0.62174
Lower	0.0086	0.0011	0.006	−0.300	0.0210	0.0540	−0.5700	0.0025	0.0018	0.0032

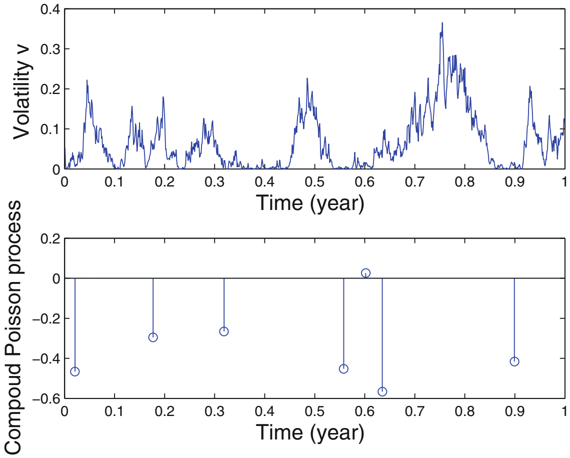


Fig. 19.1 The true volatility state and compound Poisson process

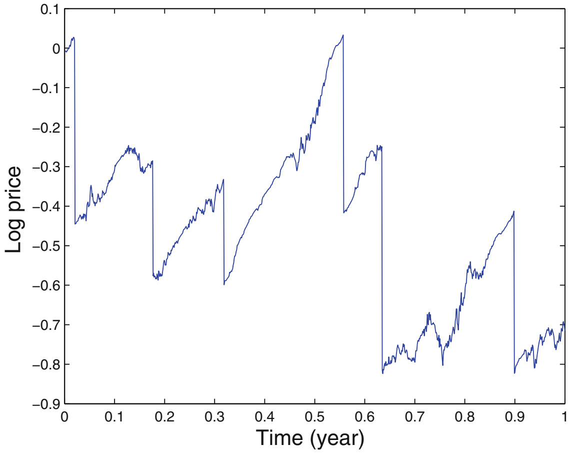


Fig. 19.2 Observed log price

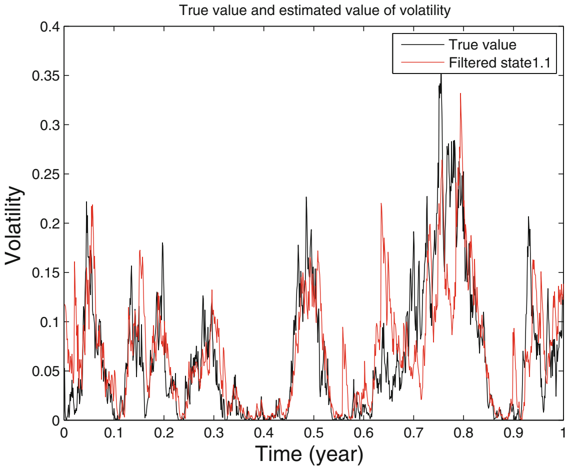


Fig. 19.3 True and estimated volatility

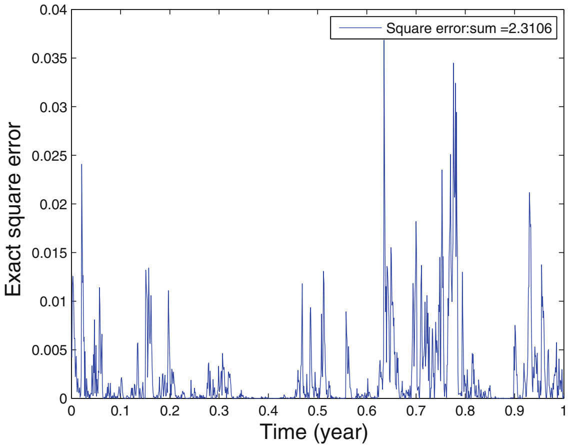


Fig. 19.4 Square error of estimated volatility

The estimates of unknown parameters are demonstrated from Figs. 19.5, 19.6, 19.7, 19.8, 19.9, 19.10, 19.11, 19.12, 19.13 and 19.14 with the corresponding histogram for $0 \leq t \leq 1$.

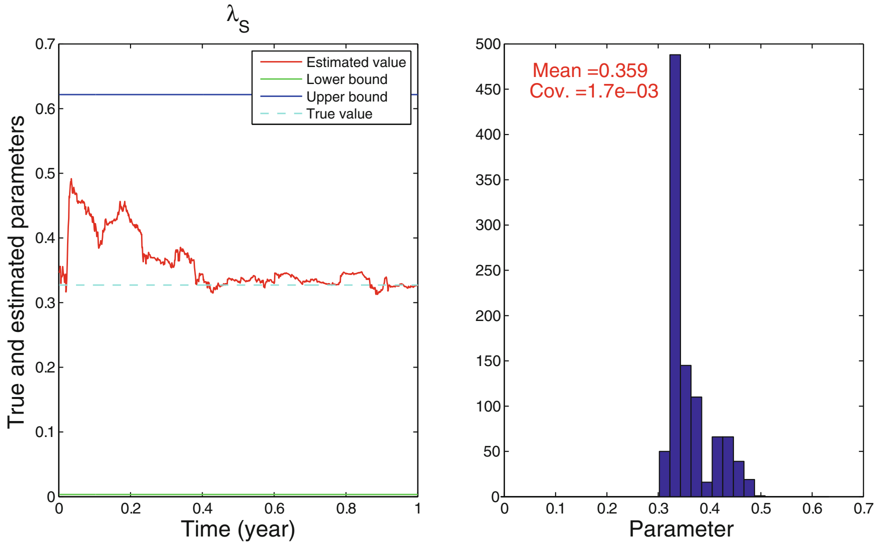


Fig. 19.5 Estimated λ_S and histogram for $0 \leq t \leq 1$

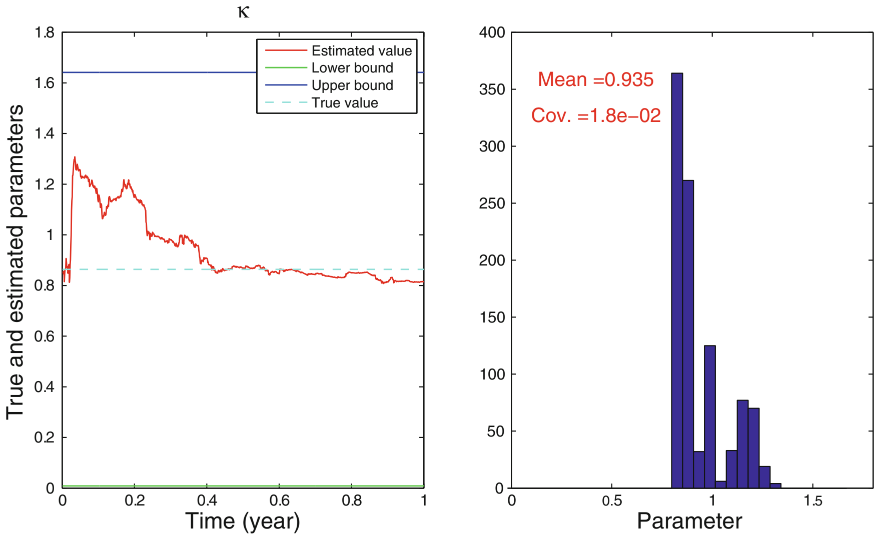


Fig. 19.6 Estimated κ and histogram for $0 \leq t \leq 1$

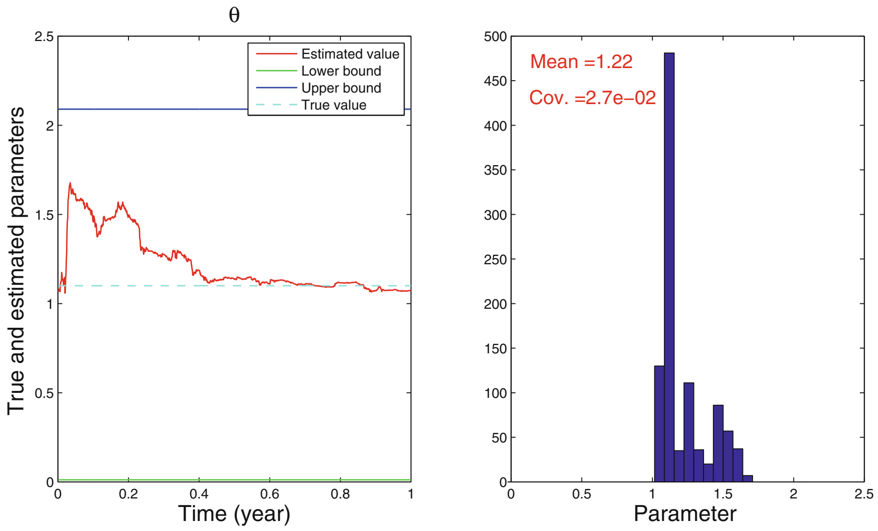


Fig. 19.7 Estimated θ and histogram for $0 \leq t \leq 1$

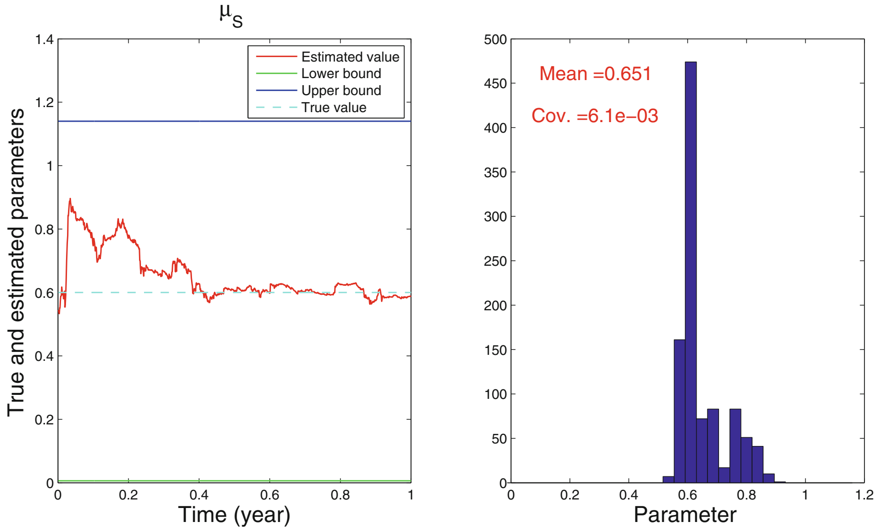


Fig. 19.8 Estimated μ and histogram for $0 \leq t \leq 1$

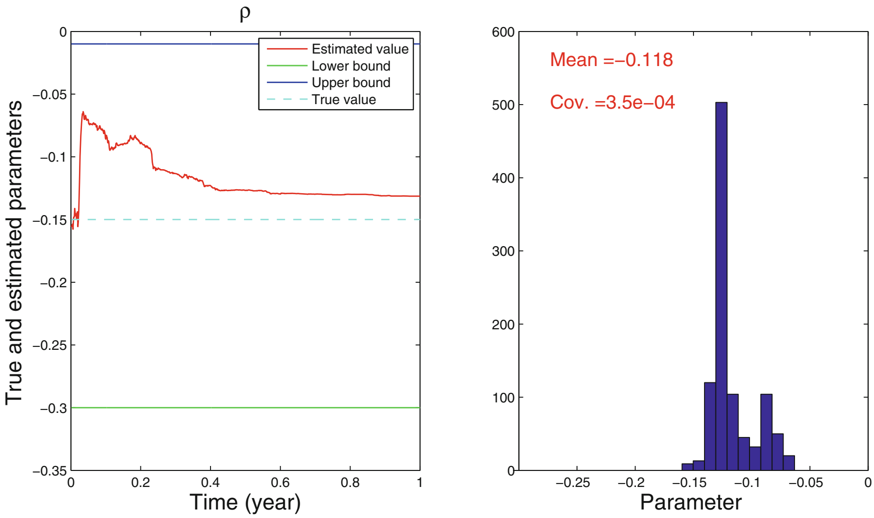


Fig. 19.9 Estimated ρ and histogram for $0 \leq t \leq 1$

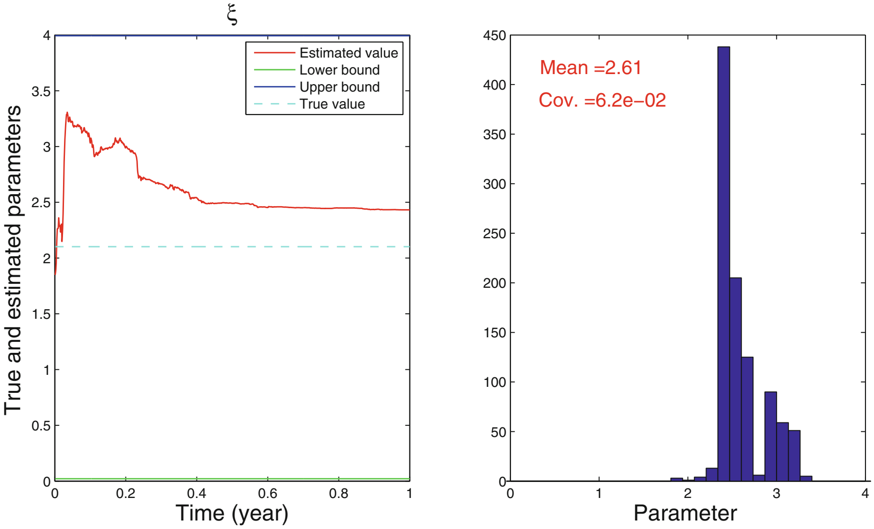


Fig. 19.10 Estimated ξ and histogram for $0 \leq t \leq 1$

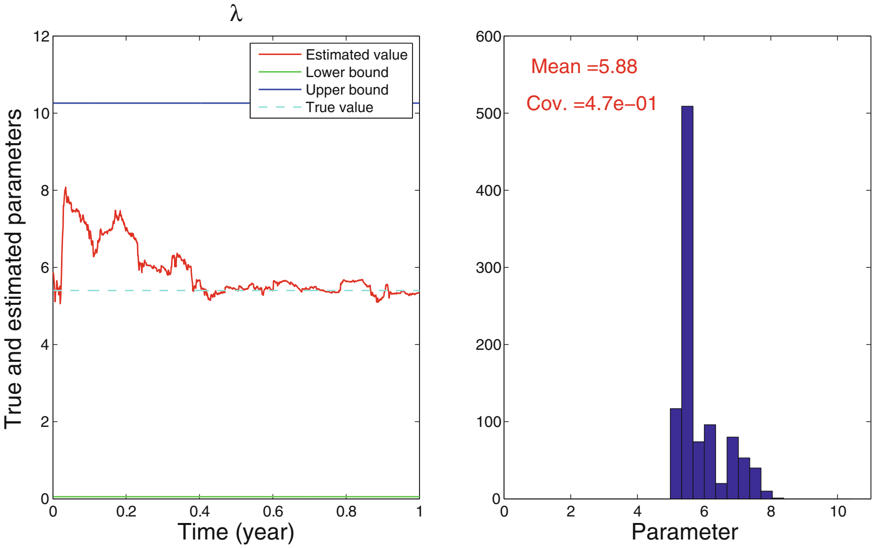


Fig. 19.11 Estimated λ and histogram for $0 \le t \le 1$

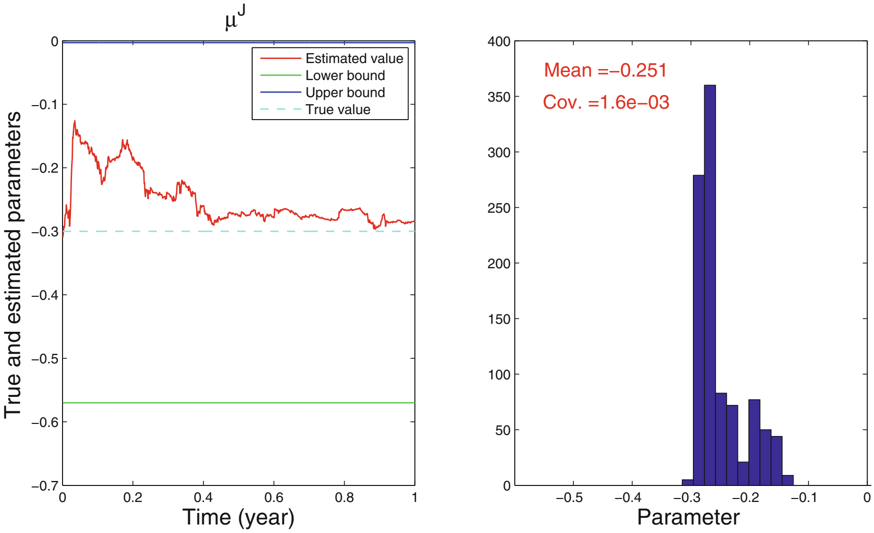


Fig. 19.12 Estimated μ^J and histogram for $0 \le t \le 1$

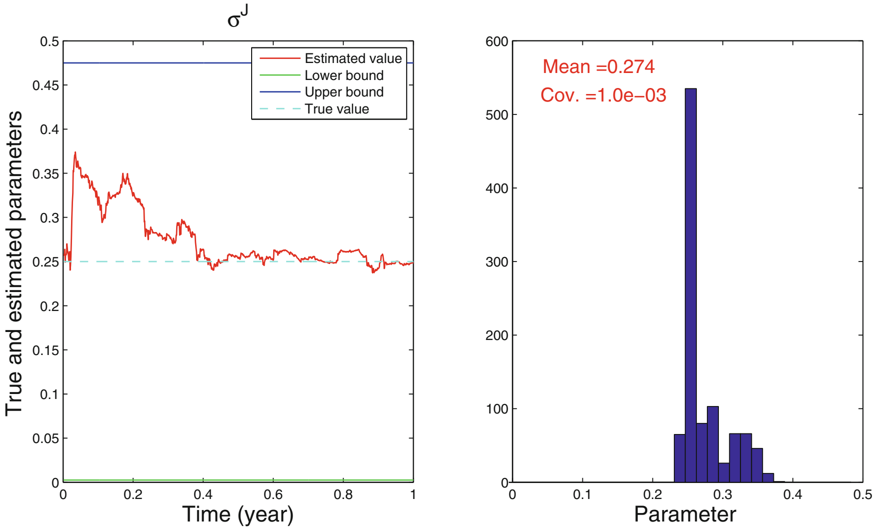


Fig. 19.13 Estimated σ^J and histogram for $0 \leq t \leq 1$

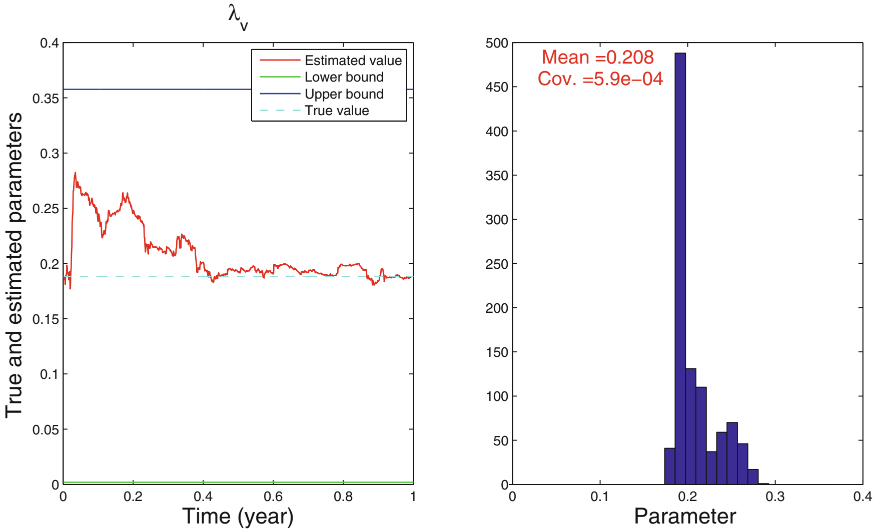


Fig. 19.14 Estimated λ_v and histogram for $0 \leq t \leq 1$

19.7 Conclusions

By using the non-central chi-square random generation method, we developed the particle filter for estimating the stochastic volatility process. The sequential estimation for the systems unknown parameters are performed with the aid of the new resampling procedure. In this procedure, we need to choose the resampling time t_r and the user defined parameter ϵ_i to obtain the good numerical results. This tuning problem is still an open problem.

References

1. Aihara, S., Bagchi, A.: Filtering and identification of Heston's stochastic volatility and its market risk. J. Econ. Dyn. Control **30**, 2363–2388 (2006)
2. Aihara, S., Bagchi, A.: Filtering and identification of affine term structures from yield curve data. IJTAF **13**, 259–283 (2010)
3. Aihara, S., Bagchi, A., Saha, S.: Estimating volatility and model parameters of stochastic volatility models with jumps using particle filter. In: Proceedings of 17th IFAC World Congress, vol. 15/1, pp. 6490–6495 (2008)
4. Aihara, S., Bagchi, A., Saha, S.: Identification of bates stochastic volatility model by using non-central chi-square random generation method. In: Proceedings of IEEE ICASSP 2012, pp. 3905–3908 (2012)
5. Anderson, B.D.O., Moore, J.B.: Optimal Filtering. Prentice-Hall Inc, Englewood Cliffs (1979)
6. Bensoussan, A.: Stochastic Control of Partially Observable Systems. Cambridge University Press, Cambridge (1992)
7. Broadie, M., Kaya, O.: Exact simulation of stochastic volatility and other affine jump diffusion processes. Oper. Res. **54**(2), 217–231 (2006)
8. Cappé, O., Moulines, E., Rydén, T.: Inference in Hidden Markov Models. Springer Science+Business Media Inc, New York (2005)
9. Doucet, A., Godsil, S., Andrieu, C.: On sequential monte carlo sampling methods for bayesian filtering. Stat. Comput. **10**, 197–208 (2000)
10. Javaheri, A.: Inside Volatility Arbitrage. Wiley, Hoboken (2005)
11. Johannes, M., Polson, N.: MCMC method for financial econometrics. In: Ait-Sahalia, Y., Hansen, L. (eds.) Handbook of Financial Econometrics. Elsevier, Amsterdam (2006)
12. Kalman, R., Bucy, R.: New results in linear filtering and prediction theory. Trans. ASME J. Basic Eng. **83**(Series D), 95–108 (1961)
13. Smith, R.: An almost exact simulation method for the heston model. J. Comput. Finance **11**(1), 115–125 (2008)
14. van Haastrecht, A., Pelsser, A.: Efficient, almost exact simulation of the heston stochastic volatility model. IJTAF **13**, 1–43 (2010)

Chapter 20
Modeling, Analysis and Control of Mechanoreceptors with Adaptive Features

Carsten Behn

Abstract This work—the development of new control strategies and sensor models—is motivated by the open question which occurred during analysis of the functional morphology of vibrissal sensor systems. The reception of vibrations is a special sense of touch, important for many insects and vertebrates. The latter realize this reception by means of hair-shaped vibrissae, to acquire tactile information about their environments. The vibrissa receptors are in a permanent state of adaption to filter the perception of tactile stimuli. This behavior now may be mimicked by an artificial sensor system. The sensor system is modeled as a spring-mass-damper system with relative degree two and the system parameters are supposed to be unknown, due to the complexity of biological systems. Using a simple linear model of a sensory system adaptive controllers are considered which compensate unknown permanent ground excitations. The working principle of each controller (feedback law including adaptor) is shown in numerical simulations which prove that these controllers in fact work successfully and effectively. Moreover, practical implementation of these controllers to a demonstrator in form of an electrical oscillating circuit results in various successful experiments which confirm the theoretical results.

Keywords Adaptive control · Bio-inspired sensor system · Modeling · Uncertain system

20.1 Introduction

In nature there are various senses that allow animals to perceive their environment. Depending on the distance of objects to be sensed from the system boundaries of the animal (usually the skin), sensors are distinguished between "far field" (e.g., vision)

C. Behn (✉)
Faculty of Mechanical Engineering, Ilmenau University of Technology,
Max-Planck-Ring 12, 98693 Ilmenau, Germany
e-mail: carsten.behn@tu-ilmenau.de
http://www.tu-ilmenau.de/tm/mitarbeiter/carsten-behn/

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_20

and “near field”, of which the sense of vibrations is a special case. Here, we focus on the sensing of vibrations for purposes of exteroception (outside the body), not ignoring the phylogenetic relation to interoception mechanisms like proprioception. Vibrations are an important piece of environmental information that insects rely on, especially arachnids, such as spiders and scorpions. To perceive vibrations, they have different types of sensilla (or tactile hairs, [1]). Vertebrates, such as cats, rats and sea lions, also possess the sense of vibration. They can perceive vibrations with the help of their vibrissae (whiskers).

Although these biological vibration receptors have a different physiology—[11] or [15] for classifications, they share common properties: When in touch with an (oscillating) object, they are moved and stimulate various (pressure-sensitive) receptors which have to analyze the stimulus and to transduce their gained adequate information to the central nervous system (CNS).

Mechanoreceptors of this kind are present throughout the integument of insects (cuticula) and mammals (fur on skin).

20.2 Bionic Aspects

Principally, the tenor of our investigations is from biomimetics (or bionics, term coined by J. E. Steele in the beginning 1960s). Bionics is a common term which includes “biological inspiration” as well. The main focus is not on “copying” the solution from biology/animality, rather on detecting the main features, functionality and algorithms of the considered biological systems to implement them in (mechanical) models and to develop ideas for prototypes. For this, a well-founded analysis of the biological paradigm is important. The global goal is not to construct prototypes with one-to-one properties of, e.g., a mechanoreceptor rather the models shall exhibit its main features.

We have to proceed in several steps, where step 1 to step 4 is usually of iterative manner:

1. analyzing live biological systems, e.g. here: receptor cells,
2. quantifying the mechanical and environmental behavior: identifying and quantifying mechanosensitive responses (e.g., pressure, vibrations) and their mechanisms as adaptation,
3. modeling live paradigms with those basic features developed before,
4. exploiting corresponding mathematical models in order to understand details of internal processes and,
5. coming to artificial prototypes (e.g., sensors in robotics), which exhibit features of the real paradigms.

20.3 Mechanoreceptors—Hair Follicle Receptors

Let us focus on mammalian receptors. The vibrissae serve mainly as levers for force transmission [18]. If the hair is deflected due to some excitations, e.g., wind, this mechanical (oscillation) energy is then transmitted to the various receptors, which respond to any movement of the hairs, see Fig. 20.1.

A receptor has only one function: to transduce a (mechanical) stimulus to neural impulses [16]. However, a receptor never continues to respond to a non-changing stimulus in transducing information to the CNS as long as the stimulus is present. It rather depends on the type of stimulus. If some impulses stimulate a receptor then there is a rapid and brief response of the receptor to it. This response declines if the stimulus is unchanging. Due to permanently changing environments the receptors have to be in a **permanent state of adaptation** to adjust their behavior. The rate or time needed to adapt or stop responding to an unchanging stimulus is the main characteristic to distinguish two different types of tactile receptors, see Fig. 20.2(*left*). The classification is, [15, 16]:

- *fast adapting (FA) receptors* encompass hair follicle sense endings: as mentioned previously, FA receptors react to applied movements or pressures with a fast (rapid) response of activity, which is succeeded by a decrease of it even though the stimulus

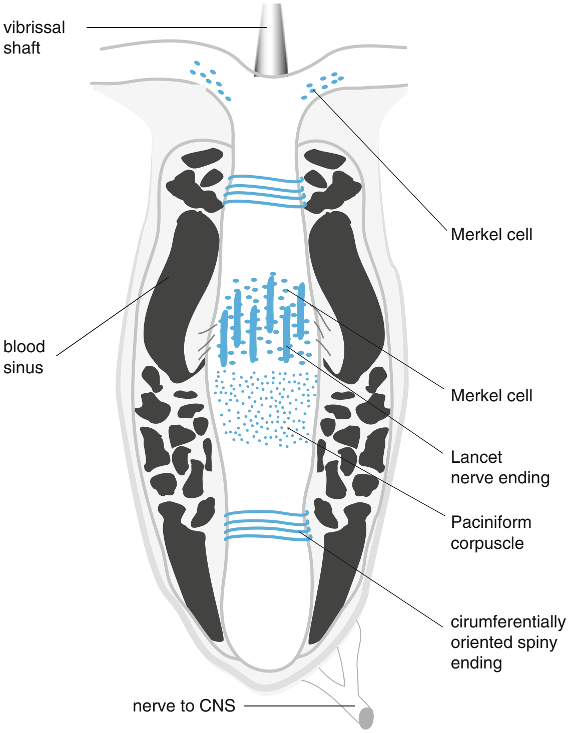


Fig. 20.1 Follicle-sinus complex (FSC) of a vibrissa with various types of receptors (*blue*); adapted from [8, 14]

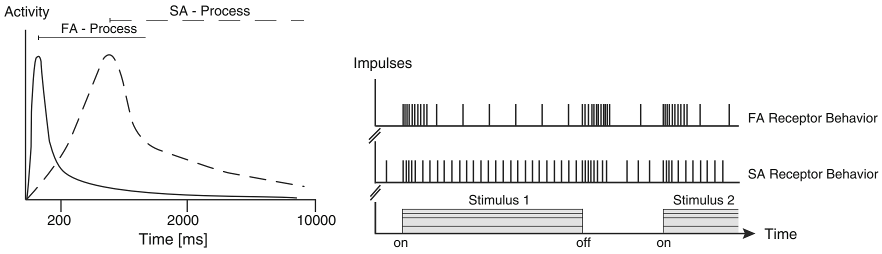


Fig. 20.2 Adaptation processes (*left*) and activity behavior (*right*) of FA and SA receptors, modified from [16]

is still present. This means, that, if a mechanical pressure via an unchanging force is applied the FA receptor responds quickly with a steep increase of activity and then decreases this activity and waits for a further stimulus, it adapts its activity in order to notice changes in the stimulus;

- the counterpart are *slow adapting (SA) receptors*, e.g., Merkel cells — these receptors work in a similar way as FA receptors but offer two ways of operation: first, as usual to receptors, a rapid response is followed by a decrease of activity. But, there is also a long duration of time of activity of these receptor cells from the beginning of the stimulus. This is in contrast to FA receptors.

The described behavior is shown in Fig. 20.2(*right*).

Here, we want to focus on the **fast adapting receptors**. The sensibility of these cells is continuously adjusted so that the receptor system converges to the rest position despite the continued excitation [7]. Hence, the perception of the continuous unchanging excitation is damped. Therefore, the excitation is considered irrelevant, once it has been perceived. If however a different excitation, such as a sudden deviation of the vibrissa sensor, occurs, this information is relevant and the sensor has to be sensitive to perceive it. If, for example, a cat is exposed to wind, the recognition of the resulting excitation of the whiskers will be damped and ignored. If the cat encounters an obstacle, the receptors should still be sensitive enough to perceive the sudden deviation of the whiskers while the wind excitation persists. Therefore, the adaption process has to ensure enduring sensitivity.

In the following, we set up a simple mechanical model to map all important features of fast adapting receptors via adaptive control strategies applied to the mechanical system.

20.4 Mechanical Modeling

Motivated by the biological observations in the foregoing section we consider a simple model of a receptor in form of a spring-mass-damper-system within a rigid frame, which is forced by an unknown time-dependent displacement $a(\cdot)$. Moreover,

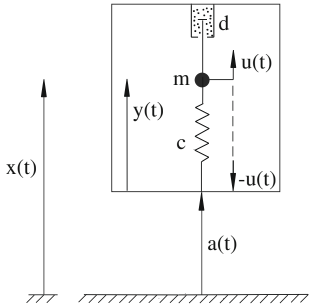


Fig. 20.3 Mechanical model of a sensor system (receptor model), [5]

the mass is under the action of an internal control force $u(\cdot)$ to compensate the unknown ground excitations, see Fig. 20.3, where x is the absolute coordinate.

The parameters of this sensory system are m (the forced seismic point mass), the damping factor d and the spring stiffness c.

We derive the differential equation of motion by using Newton's second law:

$$\left.\begin{array}{l} m\,\ddot{x}(t) = -d\,\big(\dot{x}(t) - \dot{a}(t)\big) - c\,\big(x(t) - a(t)\big) + u(t)\,, \\ x(0) = x_0\,, \quad \dot{x}(0) = x_1\,. \end{array}\right\} \tag{20.1}$$

With $y = x - a$ as the relative coordinate of the point mass, we arrive at the following differential equation of the relative motion with respect to the frame

$$\left.\begin{array}{l} m\,\ddot{y}(t) + d\,\dot{y}(t) + c\,y(t) = -m\,\ddot{a}(t) + u(t)\,, \\ y(0) = x_0 - a(0)\,, \quad \dot{y}(0) = x_1 - \dot{a}(0)\,. \end{array}\right\} \tag{20.2}$$

If $y(\cdot)$ is the measured output of the system, (20.2) is presented in normalized form

$$\left.\begin{array}{l} \begin{pmatrix} y(t) \\ \dot{y}(t) \end{pmatrix}^{\bullet} = \begin{bmatrix} 0 & 1 \\ -\frac{c}{m} & -\frac{d}{m} \end{bmatrix} \begin{pmatrix} y(t) \\ \dot{y}(t) \end{pmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ -\ddot{a}(t) \end{bmatrix} \\ y(0) = x_0 - a(0)\,, \quad \dot{y}(0) = x_1 - \dot{a}(0)\,. \end{array}\right\} \tag{20.3}$$

20.5 Scope, Problem and Goal

Scope: The goal is to achieve a predefined movement of the receptor mass m of the sensor system in Fig. 20.3 such as stabilization of the sensor system or tracking of a reference trajectory. It is obvious that the sole possibility of influencing this system lies in the (control) force $u(\cdot)$. Hence, we have to design and implement a controller which ensures a desired system output behavior. Therefore, the scope/object is to find a suitable control strategy that reproduces the specialities of the biological system receptor. This system is similar to seismic sensor systems to detect (unknown) ground excitations due to the principle of passive oscillation perception.

Problem: In general, one cannot expect to have complete information about a mechanical or biological system, but instead only structural properties are known. It is important to point out that all system parameters are supposed to be unknown because of the sophisticated nature of the biological system. The external excitation $a(\cdot)$ (biological disturbance to the receptor) is unknown and the mass, spring and damping factors are uncertain, e.g., vary in time due to thermic influences. Here, uncertainty of the factors mean that they have a positive value, but they are not known exactly, only a valid range, e.g., $c \in [\underline{c}, \overline{c}]$. Summarizing, we have to deal with a **highly uncertain (control) system** of known structure. This is why traditional control methods fail, as they rely on the knowledge of those parameters. The consideration of uncertain systems leads us to the use of adaptive control.

Task: By the above mentioned adjustment of the receptor we are given the task to adaptively compensate the unknown ground excitation: we have to design an adaptive controller, which learns from the behavior of the system, so automatically adjusts its parameters in such a way that the (seismic) point mass tends to the rest position in spite of the continuing excitation.

Goal: We choose the λ-tracking control objective [6] due to the high-gain property of the sensor system presented in (20.3). λ-tracking allows for simple feedback laws and does not focus on exact tracking since we deal with an uncertain system. Therefore, the goal is to act on the system in such a way that the system output $y(\cdot)$ is λ-tracked, i.e., the system output is forced into an error neighborhood λ around a set point trajectory $y_{\text{ref}}(\cdot)$. If $y_{\text{ref}}(\cdot) \equiv 0$, the problem is known as λ-stabilization. In this case, the receptor is supposed to remain in its equilibrium state (rest position).

Requirements: The design of controllers depends tremendously on the system properties. The adaptive control strategies should meet the following specifications:

- ability to apply the controllers without any knowledge about system parameters;
- simple feedback structure;
- optimal control performance regarding
- short settling time;
- simple structure of controller equations;
- small level of gain parameters, level of error inside the λ-tube;
- ability to quickly adapt to parameter changes.

It is imperative to keep the sensitivity of the system high. If, for example, a recurring excitation signal $a(\cdot)$ acts on the system, it is supposed that its influence is to be damped by the controller. Once the sensor/receptor has noticed this excitation it has to fade it out to wait for further new information. It has to adaptively adjust its parameters. If however the excitation subsides or is replaced by one with a much lower amplitude, the system is supposed to remain sensitive and quickly adjust the control parameters.

20.6 System Classes

The equations of motion (20.3) fall into the category of quadratic, finite-dimensional, nonlinearly perturbed, m-input $u(\cdot)$, m-output $y(\cdot)$ systems (**MIMO**-systems) of relative degree two, for short $\mathcal{S}_{2,\boldsymbol{nonlin1}}$, of the form

$$\left.\begin{aligned} \ddot{y}(t) &= A_2\,\dot{y}(t) + f_1\big(s_1(t), y(t), z(t)\big) + G\,u(t)\,, \\ \dot{z}(t) &= A_5\,z(t) + A_0\,\dot{y}(t) + f_2\big(s_2(t), y(t)\big)\,, \\ y(t_0) &= y_0\,, \quad \dot{y}(t_0) = y_1\,, \quad z(t_0) = z_0\,, \end{aligned}\right\} \tag{20.4}$$

with $y(t)$, y_0, y_1, $u(t) \in \mathbb{R}^m$, $z(t)$, $z_0 \in \mathbb{R}^{n-2m}$, A_2, $G \in \mathbb{R}^{m\times m}$, $A_0 \in \mathbb{R}^{(n-2m)\times m}$, $A_5 \in \mathbb{R}^{(n-2m)\times(n-2m)}$, $n \geq 2m$, and, for natural number q_1 and q_2 it holds

(i) $\mathrm{spec}(G) \subset \mathbb{C}_+$, i.e., the spectrum of the "high-frequency gain" lies in the open right-half complex plane;

(ii) $s_1(\cdot) \in \mathcal{L}^\infty\big(\mathbb{R}_{\geq 0}; \mathbb{R}^{q_1}\big), s_2(\cdot) \in \mathcal{L}^\infty\big(\mathbb{R}_{\geq 0}; \mathbb{R}^{q_2}\big)$ may be thought of as (bounded) disturbance terms, where $s_i(t) = \psi_i\big(t\,, y(t)\,, \dot{y}(t)\,, z(t)\big)$ is also possible with $\psi_i(\cdot\,,\cdot\,,\cdot\,,\cdot) \in \mathcal{L}^\infty\big(\mathbb{R}_{\geq 0} \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^{n-2m}; \mathbb{R}^{q_i}\big)$;

(iii) the functions $f_1 : \mathbb{R}^{q_1} \times \mathbb{R}^m \times \mathbb{R}^{n-2m} \to \mathbb{R}^m$ and $f_2 : \mathbb{R}^{q_2} \times \mathbb{R}^m \to \mathbb{R}^{n-2m}$ are continuous functions and, for compact sets $C_1 \subset \mathbb{R}^{q_1}$ and $C_2 \subset \mathbb{R}^{q_2}$, there exist $c_1, c_2 \geq 0$ such that for all $(s, y, z) \in C_1 \times \mathbb{R}^m \times \mathbb{R}^{n-2m}$

$$\big\| f_1(s, y, z)\big\| \leq c_1\,\big[1 + \|y\| + \|z\|\big]\,,$$

and for all $(s, y) \in C_2 \times \mathbb{R}^m$: $\big\|f_2(s, y)\big\| \leq c_2\,\big[1 + \|y\|\big]$.

(iv) $\mathrm{spec}(A_5) \subset \mathbb{C}_-$, i.e., the system is minimum phase, provided $f_1 = 0$, $f_2 = 0$.

It is easy to prove that every system of this system class has strict relative degree two. Therefore, relative degree two means that the control $u(\cdot)$ directly influences the second derivative of each output component. The term $A_0\,\dot{y}$ appears in connection with under-actuated systems [6].

If we inspect system (20.3) in more detail and take the physical meaning of the parameters into account, i.e., the mass of the forced (seismic) point mass m, the damping factor d and the spring stiffness c represent positive real values, then we can simplify system class (20.4) to a very special one:

- we restrict to **single**-input $u(\cdot)$, **single**-output $y(\cdot)$ systems,
- then, we claim $A_2 < 0$,
- and we neglect the coupling term, $A_0 := 0$.

Hence, we arrive at a subclass of finite-dimensional, nonlinearly perturbed SISO-system with strict relative degree two, $\mathcal{S}_{2,\boldsymbol{nonlin2}}$ for short, of the form

$$\left.\begin{aligned} \ddot{y}(t) &= A_2\,\dot{y}(t) + f_1\big(s_1(t), y(t), z(t)\big) + Gu(t)\,, \\ \dot{z}(t) &= A_5 z(t) + f_2\big(s_2(t), y(t)\big)\,, \\ y(t_0) &= y_0\,, \quad \dot{y}(t_0) = y_1\,, \quad z(t_0) = z_0\,, \end{aligned}\right\} \tag{20.5}$$

with $y(t)\,,\, y_0\,,\, y_1\,,\, u(t)\,,\, G\,,\, A_2 \in \mathbb{R}$, $z(t)\,,\, z_0 \in \mathbb{R}^{n-2}$, $A_5 \in \mathbb{R}^{(n-2)\times(n-2)}$, $n \geq 2$, and for $q_1, q_2 \in \mathbb{N}$ it holds

(i) $G > 0$, i.e., a positive input gain ("high-frequency gain");
(ii) $s_1(\cdot) \in \mathcal{L}^\infty\big(\mathbb{R}_{\geq 0}; \mathbb{R}^{q_1}\big)$ and $s_2(\cdot) \in \mathcal{L}^\infty\big(\mathbb{R}_{\geq 0}; \mathbb{R}^{q_2}\big)$, i.e., they may be thought of as (bounded) disturbance terms;
(iii) the functions $f_1 : \mathbb{R}^{q_1} \times \mathbb{R} \times \mathbb{R}^{n-2} \to \mathbb{R}$ and $f_2 : \mathbb{R}^{q_2} \times \mathbb{R} \to \mathbb{R}^{n-2}$ are continuous ones and, for compact sets $C_1 \subset \mathbb{R}^{q_1}$ and $C_2 \subset \mathbb{R}^{q_2}$, there exist c_1, $c_2 \geq 0$ such that for all $(s, y, z) \in C_1 \times \mathbb{R} \times \mathbb{R}^{n-2}$

$$\big|f_1(s, y, z)\big| \leq\; c_1 \big[1 + |y| + \|z\|\big]\,,$$

and for all $(s, y) \in C_2 \times \mathbb{R}$: $\big\|f_2(s, y)\big\| \leq\; c_2 \big[1 + |y|\big]$;
(iv) $\mathrm{spec}(A_5) \subset \mathcal{C}_-$, i.e., the system is minimum phase, provided $f_1 = 0$, $f_2 = 0$.
(v) $A_2 < 0$, i.e., this system has a zero-center in the open left-half complex plane (a "stable zero-center"), see [13].

It is easy to check that $\mathcal{S}_{2,nonlin2} \subset \mathcal{S}_{2,nonlin1}$ holds. In order to capture more relevant SISO-systems we introduce a generalized system class of $\mathcal{S}_{2,nonlin2}$ in the following — system class $\mathcal{S}_{2,nonlin3}$:

$$\left.\begin{aligned} &\ddot{y}(t) = f_0\big(s_0(t), y(t), z(t)\big)\,\dot{y}(t) + f_1\big(s_1(t), y(t), z(t)\big) + G\,u(t)\,,\\ &\dot{z}(t) = A_5\,z(t) + f_2\big(s_2(t), y(t)\big)\,,\\ &y(t_0) = y_0\,, \quad \dot{y}(t_0) = y_1\,, \quad z(t_0) = z_0\,, \end{aligned}\right\} \tag{20.6}$$

with $y(t)\,,\, y_0\,,\, y_1\,,\, u(t)\,,\, G \in \mathbb{R}$, $z(t)\,,\, z_0 \in \mathbb{R}^{n-2}$, $A_5 \in \mathbb{R}^{(n-2)\times(n-2)}$, $n \geq 2$, and for $q_0, q_1, q_2 \in \mathbb{N}$ we have to claim that the following will hold additionally:

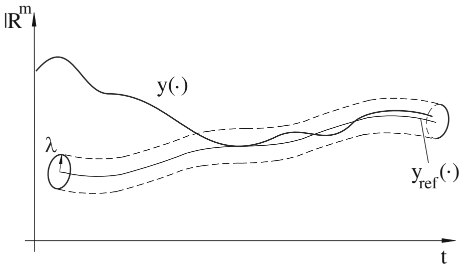
(ii) $s_0(\cdot) \in \mathcal{L}^\infty\big(\mathbb{R}_{\geq 0}; \mathbb{R}^{q_0}\big)$;
(iii) $f_0 : \mathbb{R}^{q_0} \times \mathbb{R} \times \mathbb{R}^{n-2} \to \mathbb{R}$ is a continuous function, and for a compact set $C_0 \subset \mathbb{R}^{q_0}$, there exist $c_0, \tilde{c}_0 > 0$ with $\tilde{c}_0 > c_0$, such that for all $(s, y, z) \in C_0 \times \mathbb{R} \times \mathbb{R}^{n-2}$: $-\tilde{c}_0 < f_0(s, y, z) < -c_0$.

It follows that $\mathcal{S}_{2,nonlin2} \subset \mathcal{S}_{2,nonlin3}$. For the following control systems, theorems and proofs we then focus on class $\mathcal{S}_{2,nonlin3}$ instead of $\mathcal{S}_{2,nonlin2}$.

20.7 Controllers

Since we deal with uncertain, nonlinearly perturbed (ground excitation, see the continuous functions f_i) MIMO-systems, which are not necessarily autonomous, particular attention is paid to the adaptive λ-*tracking control* objective [6]. This is to determine an *online control strategy* that achieves approximate tracking of a given, favored reference signal in the following sense:

Fig. 20.4 Reference signal and λ-tube



(i) every solution of the closed-loop system is defined and bounded for $t \geq 0$, and
(ii) the output $y(\cdot)$ tracks $y_{\text{ref}}(\cdot)$ with asymptotic accuracy $\lambda > 0$ in the sense that

$$\max\left\{0, \left\|y(t) - y_{\text{ref}}(t)\right\| - \lambda\right\} \overset{t\to+\infty}{\to} 0\,, \tag{20.7}$$

i.e., we tolerate a feasible error of prescribed size λ (accuracy). Visually, this means that the output $y(t)$ tends to a tube of radius λ around $y_{ref}(t)$, see Fig. 20.4.

Classical adaptive high-gain λ-trackers from literature are:

1. The first one is a modification of the preferred stabilizer from literature, see [10]. The modified control strategy, which is also presented in [6], is:

$$\left.\begin{aligned} e(t) &:= y(t) - y_{\text{ref}}(t)\,,\\ u(t) &= -\Big(k(t)e(t) + \tfrac{d}{dt}\big(k(t)e(t)\big)\Big)\,,\\ \dot{k}(t) &= \gamma\Big(\max\left\{0, \left\|e(t)\right\| - \lambda\right\}\Big)^2\,, \end{aligned}\right\} \tag{20.8}$$

with $k(0) = k_0 \in \mathbb{R}$, $\lambda > 0$, $y_{\text{ref}}(\cdot) \in \mathcal{R}$, $u(t), e(t) \in \mathbb{R}^{\text{m}}$, $k(t) \in \mathbb{R}$, and $\gamma \gg 1$. Due to the presented λ-tracking control objective (tolerating a tracking error of size λ, no exact tracking) this controller consists of a very simple feedback mechanism and adaptation law, and is only based on the output of the system and its time derivative—no knowledge about the system parameters is required. However, the adaptive controller (20.8) uses the derivative of the output. The following two feedback controls avoid the usage of the derivative of the system output.

2. This one includes a dynamic compensator due to a controller in [12]. This controller avoids the possible drawback of using the derivative of the output:

$$\left.\begin{aligned} e(t) &:= y(t) - y_{\text{ref}}(t)\,,\\ u(t) &= -k(t)\,\theta(t) - \tfrac{d}{dt}\big(k(t)\,\theta(t)\big)\,,\\ \dot{\theta}(t) &= -k(t)^2\,\theta(t) + k(t)^2\,e(t)\,,\\ \dot{k}(t) &= \gamma\,\max\left\{0\,, \left\|\,e(t)\right\| - \lambda\right\}^2\,, \end{aligned}\right\} \tag{20.9}$$

with $\theta(t_0) = \theta_0$, $k(t_0) = k_0 > 0$, $\lambda > 0$, $y_{\text{ref}}(\cdot) \in \mathcal{R}$, $u(t), e(t) \in \mathbb{R}^m$, $k(t) \in \mathbb{R}$, $\gamma \gg 1$ and arbitrary initial data $k_0 > 0$, $\theta_0 \in \mathbb{R}^m$.

We stress that the feedback in (20.9) does not invoke any derivatives of observables.

3. If $S_{2,nonlin1}$ is restricted to single-input, single-output systems of $\mathcal{S}_{2,nonlin3}$, then the following simple feedback control is considered, which reduces in dimension (the number of used variables calculated by internal differential equations):

$$\left.\begin{aligned} u(t) &= \; -k(t)\big[y(t) - y_{\text{ref}}(t)\big]\,, \\ \dot{k}(t) &= \gamma \; \max\big\{0, \big|y(t)\big| - \lambda\big\}^2 . \end{aligned}\right\} \tag{20.10}$$

with $k(0) = k_0 \in \mathbb{R}$. Therefore, we have a controller of order 1 whereas (20.9) is a controller of order 2. We stress that this feedback in (20.10) does not invoke any derivatives, too.

The feedback law has a P-structure, a D-term is not necessary for controlling systems of the class $\mathcal{S}_{2,nonlin3}$. Naturally we need a P- and D-term to control systems with strict relative degree two, see [17].

To summarize, these controllers are simple in their design, rely only on structural properties of the system (and not on the system's data) and do not invoke any estimation or identification mechanism. They only consist of a feedback strategy and a simple parameter adaptation law, and, moreover, do not have to depend on the derivative of the output of the system.

All three controllers achieve λ-tracking (and, of course, λ-stabilization using $y_{\text{ref}}(\cdot) \equiv 0$, as well) in applying all three controllers (20.8), (20.9) and (20.10) to the system classes:

- Theorem I: controller (20.8) applied to systems of class $\mathcal{S}_{2,nonlin1}$, in [6];
- Theorem II: controller (20.9) applied to systems of class $\mathcal{S}_{2,nonlin1}$, in [2];
- Theorem III: controller (20.10) applied to systems of class $\mathcal{S}_{2,nonlin3}$, proven and submitted.

The parameter γ strongly determines the growth of the gain parameter $k(\cdot)$. In [6] the case $\gamma = 1$ was dealt with. With small γ (e.g., $\gamma = 1$ as formerly) $k(\cdot)$ often grows too slowly as to achieve a good tracking behavior. Therefore, a sufficiently large $\gamma \gg 1$ should be used. But, if we choose γ too large, we arrive at high feedback values. Furthermore, these high values keep the sensor not really to be sensitive to extraordinary impulses, generally speaking, the receptor is "blind" if the signal is forced once into the tube, because it cannot detect the peak in observing the output. The last requirement to the controllers is not fulfilled: the closed-loop sensor system has to be sensitive to recurring excitation signals—fade it out and wait for further new information. This is not realized yet. We are able to dominate the system, but we are not able to get information on the environment in observing the output. This is addressed in the next section–design of new adaptation laws – to identify the (whole) ground excitation or only some basic characteristics of it.

20.8 Adaptors

The drawback of the **'Classical' Adaptor:**

$$\dot{k}(t) = \gamma \left(\max \left\{ 0, \, \|e(t)\| - \lambda \right\} \right)^2 , \tag{20.11}$$

is $\dot{k}(t) \geq 0, \forall \, t \geq 0$, i.e.,

$$t \mapsto k(t) = k(0) + \int_0^t \gamma \Big(\max \left\{ 0, \, \|e(\tau)\| - \lambda \right\} \Big)^2 d\tau \geq 0$$

thus implies monotonic increase of $k(\cdot)$. Typically, the classical high-gain adaptive controllers (feedback law including adaptation law) yield a non-decreasing gain, which is usual. Now we propose some new adaptation laws, which let $k(\cdot)$ decrease when e is in the tube.

A very simple modification of the adaptation law is the so-called σ-modification, $\sigma > 0$. For λ-tracking control including the gain coefficient γ [9], and revisited in [4] in simplified form we have **Adaptor 1**:

$$\dot{k}(t) = -\sigma \, k(t) + \gamma \left(\max \left\{ 0, \, \|e(t)\| - \lambda \right\} \right)^2 , \tag{20.12}$$

with $\sigma > 0, \gamma \gg 1$. The term $-\sigma \, k(t)$ decreases $k(\cdot)$ exponentially, while the second term ensures a quadratic increase of $k(\cdot)$ when $\|y(t)\|$ is outside the λ-strip. Therefore, Adaptor 1 offers two terms which are active simultaneously and counteract each other. Depending on the situation, one of the terms overcomes the effect of the other and results in a global decrease or increase of $k(\cdot)$. This law often leads to *oscillatory* behavior (maybe limit cycles) and even *chaotic* one of the system. Hence, this adaptor has to be treated carefully, because the dynamical behavior depends crucially on the parameters $\sigma > 0$. Therefore, we will not focus on this adaptor type in sequel.

The idea is now to split the part of increase and decrease of the gain as follows in **Adaptor 2** [4]:

$$\dot{k}(t) = \begin{cases} \gamma \left(\|e(t)\| - \lambda \right)^2 , & \|e(t)\| \geq \lambda \, , \\ -\sigma \, k(t), & \|e(t)\| < \lambda , \end{cases} \tag{20.13}$$

with $\sigma > 0, \gamma \gg 1$. This adaptor shows alternating increase and exponential decrease of $k(\cdot)$.

It could happen that e rapidly traverses the λ-tube. Then it would be inadequate to immediately decrease $k(\cdot)$ after e entered the tube. Rather we should distinguish three cases:

1. increasing $k(\cdot)$ while e is outside the tube,
2. constant $k(\cdot)$ after e entered the tube—no longer than a pre-specified duration t_d of stay, and
3. decreasing $k(\cdot)$ after this duration has been exceeded.

So, another adaptation law of this kind is **Adaptor 3** [5]:

$$\boxed{\dot{k}(t) = \begin{cases} \gamma \left(\left\|e(t)\right\| - \lambda\right)^2, \ \left\|e(t)\right\| \geq \lambda\,, \\ 0, \ \left(\left\|e(t)\right\| < \lambda\right) \wedge (t - t_E < t_d)\,, \\ -\sigma\, k(t), \ \left(\left\|e(t)\right\| < \lambda\right) \wedge (t - t_E \geq t_d), \end{cases} \qquad (20.14)}$$

with given $\sigma > 0$, $\gamma \gg 1$, and $t_d > 0$, whereas the entry time t_E is an internal time variable.

If the norm of the error value $\|e\|$ is close to the λ-strip, i.e., the system output y is already close to the λ-tube, and $0 < \|e\| - \lambda < 1$ holds, an exponent of $p = 2$ leads to an even smaller number. This is the main disadvantage in such a way, that, if $\|y\|$ is already close to the λ-strip around the prescribed reference signal, the adaption process, i.e., the increase of $k(\cdot)$, is very slow.
In order to make the attraction of the tube stronger, it would be advantageous to use different exponents p with better performance such as a square root. Hence, a kind of scheduling of $\dot{k}$ is introduced, different exponents for large/small distances from the tube, see [5] and **Adaptor 4:**

$$\boxed{\dot{k}(t) = \begin{cases} \gamma \left(\left\|e(t)\right\| - \lambda\right)^2, \ \left\|e(t)\right\| \geq \lambda + 1\,, \\ \gamma \left(\left\|e(t)\right\| - \lambda\right)^{0.5}, \ \lambda + 1 > \left\|e(t)\right\| \geq \lambda\,, \\ 0, \ \left(\left\|e(t)\right\| < \lambda\right) \wedge (t - t_E < t_d)\,, \\ -\sigma\, k(t), \ \left(\left\|e(t)\right\| < \lambda\right) \wedge (t - t_E \geq t_d), \end{cases} \qquad (20.15)}$$

with σ , γ , t_d , t_E as before.

Let $\lambda > 0$ be chosen in regard of certain requirements given by the context. To ensure that the system output y stays within that λ-tube along the reference signal

(e will not leave the λ-strip after entering the strip) is to track a smaller safety radius $\varepsilon\,\lambda < \lambda$, suggestions for adaptation laws are now **Adaptor 5:**

$$\dot{k}(t) = \begin{cases} \gamma\left(\|e(t)\| - \varepsilon\,\lambda\right)^2, \ \|e(t)\| \geq \varepsilon\,\lambda + 1\,, \\ \gamma\left(\|e(t)\| - \varepsilon\lambda\right)^{\frac{1}{2}}, \varepsilon\lambda + 1 > \|e(t)\| \geq \varepsilon\lambda\,, \\ 0, \left(\|e(t)\| < \varepsilon\,\lambda\right) \wedge (t - t_E < t_d)\,, \\ -\sigma\,k(t), \left(\|e(t)\| < \varepsilon\,\lambda\right) \wedge (t - t_E \geq t_d), \end{cases} \tag{20.16}$$

with σ , γ , t_d , t_E as before.

20.9 Simulation

We point out, that the adaptive nature of the controllers is expressed by the **arbitrary choice** of the system parameters. Obviously numerical simulation needs fixed (and known) system data, but the controllers **adjust** their gain parameter **to each set** of system data. The numerical simulation will demonstrate and illustrate that the adaptive controllers work successfully and effectively.

Choosing the parameters from Table 20.1 (which are arbitrarily chosen, not measured or identified from the biological paradigm, just for simulation purposes) and $\varepsilon = 0.7$ we get the results shown in Figs. 20.5(*left*) and 20.5(*right*) in applying Adaptor 5.

In former simulations, the output apparently periodically leaves the λ-tube. Then $\varepsilon\,\lambda$-tracking (we will call this kind of tracking ε-safe λ-tracking) with $\varepsilon = 0.7$ makes e not to leave the desired λ-tube, see Fig. 20.5(*left*).
The steep increase of $k(\cdot)$ at the beginning is due to the "switching on" of the controller and the small initial value of $k(0) = k_0 = 0$. This could be prevented in

Table 20.1 Global simulation parameters (dimensionless)

sensor mass m	1
damping coefficient d	5
spring stiffness c	10
initial values $\big(y(0)\,,\,\dot{y}(0)\big)$	$\big(-a(0)\,,\,-\dot{a}(0)\big)$
tolerance λ	0.2
initial gain value k_0	0
ref. signal $t \mapsto y_{\mathrm{ref}}(t)$	0 (rest position)
ground excitation	$t \mapsto a(t) = \sin(2\,\pi\,t)$

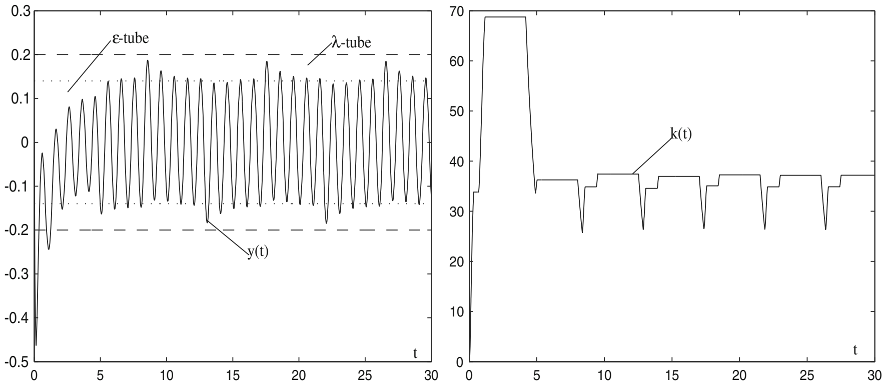


Fig. 20.5 Output $y(\cdot)$ and tubes (*left*), gain parameter $k(\cdot)$ (*right*), all versus time t

choosing a larger k_0. But, it depends on the system data which is unknown a-priori. Comparing the simulation results and the sensor behavior of the gain parameter $k(\cdot)$ in Fig. 20.5(*right*) with the impulse sequences in Fig. 20.2(*right*), one clearly recognizes that we achieved the behavior of the biological paradigm.

20.10 First Experiment

This section is devoted to the experimental verification of the successful implementation of the controller (feedback including Adaptor 5) developed above. For this purpose, we built up a demonstrator in form of an electrical oscillating circuit, the test rig is presented in Fig. 20.6(*left*). The demonstrator is shown in Fig. 20.6(*right*).

Then, the equations of motion are, using Lagrange's equations of the 2nd kind

$$L\,\ddot{q}(t) + R\,\dot{q}(t) + \frac{1}{C}\,q(t) = U(t) + u(t)\,. \tag{20.17}$$

The system output shall be the charge $q(\cdot)$. The goal is to adaptively compensate changes of $U(t)$ by means of the control input $u(t)$ to λ-track $q_{\text{ref}}(\cdot) \equiv 0$. As a rule, the charge is measured due to the voltage at the capacitor in form of

$$q(t) = C\,U_C(t)\,.$$

Due to the small system parameter values the gain $k(\cdot)$ will increase tremendously and we need high computing capacity. To avoid this we will directly control the voltage $U_C(\cdot)$ which depends linearly on the measured $q(\cdot)$, see above.

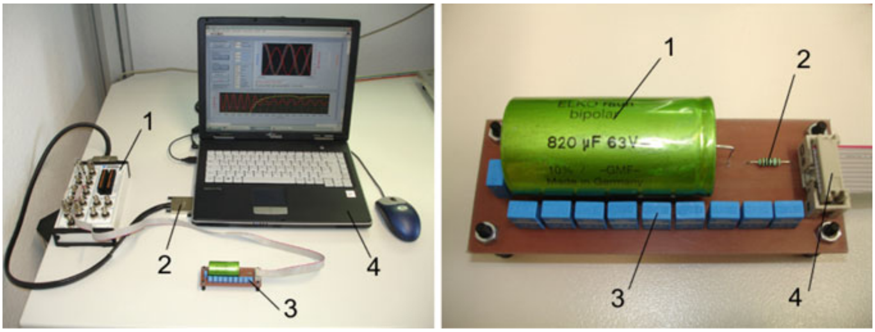
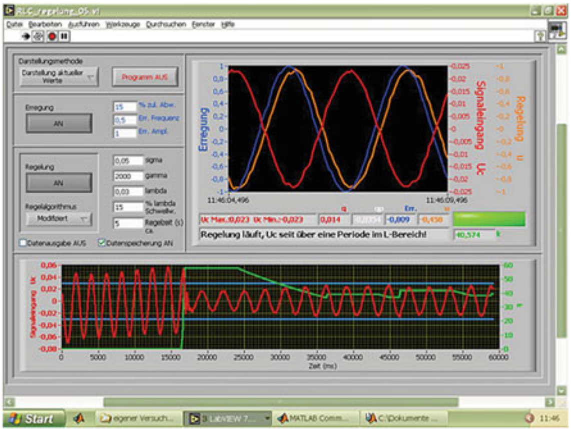


Fig. 20.6 *Left*: test rig with electrical oscillating circuit: 1 - I/O-system (BNC-2110), 2 - DAQ-6036-PCMCIA-card, 3 - demonstrator, 4 - PC with LabView; right: Circuit with 1 - capacitor ($C = 800\,\mu F$) , 2 - resistor ($R = 100\,\Omega$), 3 - one inductor (overall inductance $L_{ges} = 640\,mH$), 4 - communication to PC

We apply Adaptor 5 to guarantee that the error will not leave the λ-tube in tracking a tube of smaller radius $\varepsilon\,\lambda$. We have

- excitation: $t \mapsto U(t) = U_0 \sin(\omega\, t)$ with amplitude $U_0 = 5\,\text{V}$ and frequency $f = 0.5\,\text{Hz}$;
- Adaptor 5: $\gamma = 1000$, $\lambda = 0.03\,\text{V}$, $\varepsilon\,\lambda = 0.02\,\text{V}$ (much smaller tolerance), $\sigma = 0.05$, $t_d = 6\,\text{s}$.

We perform this experiment in using LabView to handle and to control the circuit. By means of a programmed LabView control panel, see Fig. 20.7, we are able to switch on/off the excitation $U(\cdot)$ and the control strategy $u(\cdot)$. Furthermore, several signals are displayed via this panel:



15 % zul. Abw.
0,5 Err. Frequenz
1 Err. Ampl.
0,05 sigma
2000 gamma
0,03 lambda
15 % lambda Schwellw.
5 Regelzeit (s) ca.
□ Datenausgabe AUS ☑ Datenspeicherung AN
Uc Max.:0,023 Uc Min.:-0,023 0,014 -0,009 -0,450
Regelung läuft, Uc seit über eine Periode im L-Bereich!
40,574

Fig. 20.7 LabView front panel on PC screen, using adaptor (20.16); depicted curves in *bottom* window: capacity voltage $U_C(\cdot)$, i.e., new output $y(\cdot)$, (*red line*), λ-strip (*blue lines*), and gain parameter $k(\cdot)$ (*green line*)

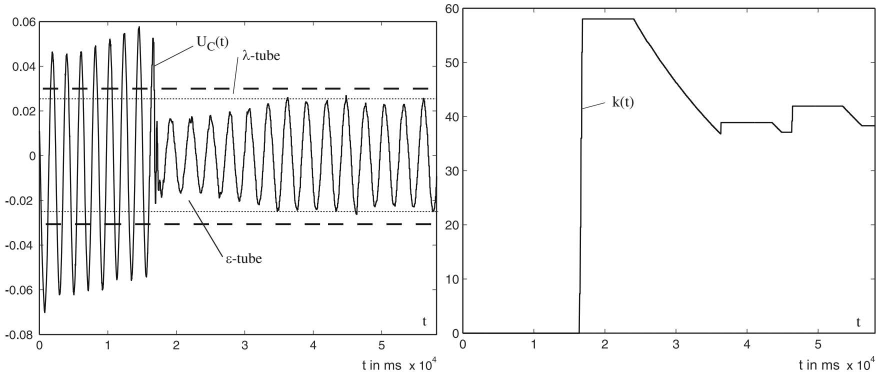


Fig. 20.8 Output $U_C(\cdot)$ and tubes (*left*), and gain parameter $k(\cdot)$ (*right*), all versus time t in using Adaptor 5

- in the top window (actual measured data): the control input $u(\cdot)$ (orange line), the system output $U_C(\cdot)$ (red line), the excitation signal $U(\cdot)$ (blue line);
- the bottom window (data on time horizon): the depicted curves are capacity voltage $U_C(\cdot)$ (i.e., the output $y(\cdot)$, red line), the λ-strip (blue lines), and gain parameter $k(\cdot)$ (green line), in only one window.

The plots of the exported data-files from LabView are shown in Fig. 20.8. The capacitor voltage (output) never leaves the λ-tube, the adaptor works effectively. Further experiments can be found in [3].

Comparing the simulation results with $\sigma = 0.05$ in Fig. 20.8(*right*) with the impulse sequences in Fig. 20.2(*right*) and the adaptation behavior in Fig. 20.2(*left*), we conclude that this technical device offers the behavior of the biological paradigm.

20.11 Conclusions

Based on the paradigm biological receptor and its fundamental feature to filter signals and transduce them, we have set up a mechanical sensor system to find hints to establish a measurement or monitoring system. These technical systems have to offer high sensitivity to signals from the environment. To mimic the complex behavior of the biological system, adaptive controllers had to be applied to a mechanical sensor system to compensate and filter unknown ground excitations (uncertainties of the system).

We considered a mechanical system in form of a spring-mass-damper-system within a rigid frame which is forced by a time-dependent ground excitation. Moreover, the mass is under the action of an internal control force. Due to a biological system we dealt with a system which is not known precisely. In detail, we assumed that the system parameters are unknown (uncertainty of system). Moreover, we allowed for uncertainties with respect to perturbations from the environment (ground excitations).

Such systems need adaptive controllers which achieve a predefined control objective. We exposed the need of improvements of already existing strategies from literature—almost all known controllers offer the same drawback: (possibly unboundedly) monotonically increasing gain parameters. Classical adaptors suffer from this monotonic increase of the control gain parameter, thereby possibly paralyzing the sensor's capability to detect future extraordinary excitations. The existing adaptive controllers from literature were improved with respect to performance, sensitivity and capabilities. Various modifications of existing controllers were made and new controller designs were discussed:

- tuning parameters γ and gain exponent p increase the growth rate of the gain parameter k;
- new adaptors allow for gain parameter decrease that improves the sensor system's sensitivity to further ground excitations;
- a smaller $\varepsilon\,\lambda$-tube is introduced to prevent the output y from leaving the λ-neighborhood.

These proposed and modified adaptors avoid the drawbacks from literature and do not invoke any estimation or identification techniques. The working principle of the new controller (feedback law including Adaptor 5) is shown in numerical simulations which proved that this controller in fact works successfully and effectively. This controller is simple in its design: its adaptation law is not complex as current adaptive control strategies in literature. Moreover, a practical implementation of this controller to a demonstrator in form of an electrical oscillating circuit results in a successful experiment which confirms the theoretical results.
However, the preceding simulation results shed some light upon the behavior of the sensor system under the governance of various adaptive controllers. Simulation and experiment show that the developed adaptive control strategy applied to the mechanical sensor system achieve the fast adapting behavior of the biological receptor.

There are various aspects of adaptive controllers that can be improved and modified in current and future work on the subject:

- tuning of the chosen controller data (optimize controller data)—clear that this is a delicate task;
- avoidance of error derivative;
- constrained control input;
- intelligent control;
- finite-time-control;
- doing further experiments.

Acknowledgments The author thanks Joachim Steigenberger (TU Ilmenau) for his critical remarks, improving suggestions and his continued interest in my work. Thanks go also to Johannes Zeh, a former diploma student of the author, who made up a fairly long list of adaptors from literature. The successfully accomplished experiments with an electrical device are done with the assistance of him, Valter Böhm and Siegfried Oberthür (all from TU Ilmenau).

References

1. Barth, F.G.: Spider mechanoreceptors. Curr Opin Neurobiol **14**, 415–422 (2004)
2. Behn, C.: Adaptive control of straight worms without derivative measurement. Multibody Sys Dyn **26**(3), 213–243 (2011)
3. Behn, C.: Mathematical modeling and control of biologically inspired uncertain motion systems with adaptive features. Habilitation thesis, Ilmenau University of Ilmenau, Germany (2013)
4. Behn, C., Steigenberger, J.: Improved adaptive controllers for sensory systems - First Attempts. In: Modeling, Simulation and Control of Nonlinear Engineering Dynamical Systems, (Ed.) J. Awrejcewicz, pp. 161–178. Springer, Amsterdam (2009)
5. Behn, C., Steigenberger, J.: Experiments in adaptive control of uncertain mechanical systems. Int Rev Mech Eng **4**(7), 886–898 (2010)
6. Behn, C., Zimmermann, K.: Adaptive λ - tracking For locomotion systems. Robot Auton Syst **54**, 529–545 (2006)
7. Dudel, J., Menzel, R., Schmidt, R.F.: Neurowissenschaft. Springer, Berlin (1996)
8. Ebara, S., Kumamoto, K., Matsuura, T., Mazurkiewicz, J.E., Rice, F.L.: Similarities and differences in the innervation of mystacial vibrissal follicle-sinus complexes in the rat and cat: A confocal microscopic study. J Comp Neurol **449**, 103–119 (2002)
9. Georgieva, P., Ilchmann, A.: Adaptive λ-tracking control of activated sludge processes. Int J Control **74**(12), 1247–1259 (2001)
10. Ilchmann, A.: Non-identifier-based adaptive control of dynamical systems: a survey. IMA J Math Control Inform **8**, 321–366 (1991)
11. Iwasaki, M., Itoh, T., Tominaga, Y.: VI Mechano- and Phonoreceptors. In: Atlas of arthropod sensory receptors - Dynamic Morphology in Relation to Function, (Eds.): E. Eguchi and Y. Tominaga, pp. 177–190. Springer, Berlin (1999)
12. Miller, D.E., Davison, E.J.: An adaptive controller which provides an arbitrarily good transient and steady-state response. IEEE Trans Autom Control **36**, 68–81 (1991)
13. Ogata, K.: Modern control engineering. 3rd Edition, Prentice Hall (1997)
14. Rice, F.L., Mance, A., Munher, B.L.: A comparative light microscopic analysis of the sensory innervation of the mystacial pad. I. Innervation of vibrissal follicle-sinus complexes. J Comp Neurol **252**, 154–174 (1986)
15. Smith, C.U.M.: Biology of Sensory Systems. 2nd Edition. Wiley, New York (2008)
16. Soderquist, D.R.: Sensory processes. Sage Publications, Thousand oaks (2002)
17. Sontag, E.D.: Mathematical control theory. 2nd Edition, Springer, Berlin (1998)
18. Voges, D., Carl, K., Klauer, G., Uhlig, R., Behn, C., Schilling, C., Witte, H.: Structural characterisation of the whisker system of the rat. IEEE Sens J **12**(2), 332–339 (2012)

Chapter 21
Memory-Based Logic Control for Embedded Systems

Václav Dvořák and Petr Mikušek

Abstract Implementation of logic control algorithms in embedded systems is limited by space and response time. Use of a single look-up table (LUT) for multiple-output Boolean function is almost always excluded due to the LUT size. The paper deals with implementation in a form of cascade of smaller LUTs with response time given by a series of few look-ups. Cascade length and memory required to store LUTs can be varied and it is shown that an optimal trade-offs can be reached. Changes in logic control can be implemented easily by re-loading data into LUTs. The presented method is thus useful for logic control in embedded systems or in microcontroller software.

Keywords Logic control · Multiple-output logic functions · Look-up table (LUT) cascades · Embedded systems

21.1 Introduction

Micro-controller-based systems such as automobile engine control systems, implantable medical devices, remote controls, appliances, office machines etc. are ubiquitous. Reducing the size and cost of these systems is essential. One area susceptible to such reduction is customized logic that frequently uses separate devices (PLAs or FPGAs). As we will show later, logic devices can be replaced by memory modules and fast enough series of look-ups in the chain of these modules. This may be often without additional space requirements as some memory capacity may remain unused in RAM chips.

V. Dvořák (✉) · P. Mikušek
Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic
e-mail: dvorak@fit.vutbr.cz

P. Mikušek
e-mail: petr@mikusek.info

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_21

Traditional serial evaluation of Boolean functions one at a time, e.g. in programmable logic controllers, implied redundant reading of input variables. Representing logic functions by means of binary decision diagrams (BDD) helped to remove that redundancy and to implement single-output logic functions using a RAM and a sequencer [1]. However, evaluation of multiple outputs was still done serially by means of auxiliary variables. The similar partitioning of outputs was used even in special purpose processors (Decision Diagram Machines, DDMs) that evaluate decision diagrams via branching programs [2]. Parallel evaluation of Boolean functions specified by Multi-terminal BDDs (MTBDDs) was done on parallel branching machine [3], but with quaternary branching only. This evaluation can be much too slow, unless we use a number of branching machines.

In this paper we will try to avoid branching programs and MTBDDs completely. We can visualize any logic function as a look-up table (LUT). Because one LUT is usually much too large, our approach is based on the iterative decomposition of the given function, one variable at a time, producing a cascade of smaller LUTs. The next step is optimal clustering of these LUTs into larger ones implemented as RAM modules. The goal is to obtain the minimum total size of look-up tables or the minimum number of LUTs in a cascade. Whereas the first goal is loosely related to minimum space, the second one is about the time involved in the chain of memory accesses. Contrary to the traditional logic design methods such as logic minimization or FPGA synthesis oriented to networks of LUTs with up to 6 inputs, our approach is targeting larger memory modules and coarse-grain linear structures (LUT cascades).

The main contribution of the paper is the algorithm of iterative decomposition/clustering and its implementation. It is the upgraded version of a former tool [4]. An accepted format for sets of incomplete Boolean functions at the input is cube notation. The paper is structured as follows. In the following Sect. 21.2 we explain representation of logic functions in cube notation and then the concept of simple decomposition. In Sect. 21.3 we introduce iterative decomposition on a simple example. Logic functions and related LUT cascades are dealt with in Sect. 21.4, together with related software tools. The decomposition method is applied to two sorts of examples in Sect. 21.5. The results are commented on in Conclusions.

21.2 Logic Functions

To begin our discussion, we introduce the following terminology. A system of m Boolean functions of n Boolean variables,

$$f_{\mathrm{n}}^{(\mathrm{i})}:\ (Z_2)^{\mathrm{n}} \to Z,\ \ i = 1,\ 2,\ \ldots, m \tag{21.1}$$

will be simply referred to as a multiple-output Boolean function F_n. If $Z = Z_2 = \{0, 1\}$, the function $f_n^{(i)}$ is complete. If the value of function $f_n^{(i)}$ is of no concern or not defined for some input vectors, we can use ternary values $Z = \{0, 1, \sim\}$ for the outputs and take the symbol "$\sim$" as a dont'care value.

Further on we will work with incompletely specified, multiple-output functions of *n* Boolean variables

$$F_n : \ (Z_2)^n \rightarrow Z, \quad Z \subset \{0, 1, \sim\}^m. \tag{21.2}$$

There are two special cases of the *m*-element ternary output vectors. Binary output vectors *v* with no dont'cares, $v \in \{0,1\}^m$, will be alternatively coded by integer values from $Z_R = \{0, 1, 2, \ldots, R - 1\}$, $R \leq 2^m$. The output vector with all don't cares means that the function value is arbitrary or the related input vector cannot occur. Espresso input format fr is assumed for function $f_n(i)$; it means that each input vector belongs to the ON-set, to the OFF-set, or to the DC-set depending on the ternary value 1, 0, or "$\sim$" of the output. If the function is given in Espresso format f (PLA format), only the ON-set is specified and an extra step is required to generate the OFF-set.

Instead of full input vectors from $(Z_2)^n$ we prefer to use a compact cube notation [5]. $(n+m)$-tuples are called *function cubes*, in which an element of $\{0, -, 1\}^n$ is called an *input cube* and element of $\{0, \sim, 1\}^m$ is called an *output cube*. The value of symbol "–" is 0 or 1, so that one cube can cover several input vectors.

Definition 1 Compatibility relation. All pairs of ternary values are compatible except pairs [0,1] and [1,0]. Two cubes *c*,*c*' are compatible, $c \approx c$', if they are compatible element-wise.

In other words, two cubes *c* and *c*'are compatible if and only if they have a non-empty common sub-cube. The compatibility relation $\approx$ is reflexive and symmetric. Additional notation is needed for special set systems:

Definition 2 Blanket β on a set *S*, $\beta = \{b_1, b_2, \ldots, b_k\}$, is a set of distinct, non-empty, possibly overlapping subsets of *S*, called blocks, whose union is *S*.

Example: $S = \{1,2,3,4,5\}$ is covered by blanket $\beta = \{\underline{1,2,3}; \underline{3,4}; \underline{1,3,4,5}\}$. To improve the notation, three blocks in β are underlined.

Definition 3 The product of two blankets is defined as

$$\beta_1 * \beta_2 = \ \{b_\mathrm{i} \cap b_\mathrm{j} \ |b_\mathrm{i} \in \beta_1 \text{ and } b_\mathrm{j} \ \in \beta_2\} \tag{21.3}$$

with empty and redundant blocks removed.

Example: $\beta_1 = \{\underline{1,2,3}; \underline{3,4}; \underline{3,4,5}\}$, $\beta_2 = \{\underline{1,2}; \underline{3}; \underline{1,4}\}$, $\beta_1 * \beta_2 = \{\underline{1,2}; \underline{3}; \underline{1}; \underline{4}\}$.

If β_1 and β_2 are blankets on *S*, we write $\beta_1 \leq \beta_2$ if and only if for each b_i in β_1 there exists b_j in β_2 such that $b_i \subseteq b_j$. For example, $\{\underline{1,2}; \underline{2,3}\} \leq \{\underline{1,2}; \underline{1,2,3}; \underline{1,3}\}$.

21.3 Decomposition Method

Definition 4 Functional decomposition of function F_n $(x_1, x_2, \ldots, x_n) = F_n$ (X) is a serial disjunctive separation of F into two functions G and H such that

$$F_\mathrm{n}(X) \;= H_{\mathrm{k+n-h}}(U, G_\mathrm{h}(V)), \tag{21.4}$$

where U, V are disjunctive subsets of set X, $U \cap V = \emptyset$, $U \cup V = X$, see Fig. 21.1. Of course, we are interested only in non-trivial decompositions when functions G and H have strictly fewer inputs than F, i.e.

$$h < n, \; k + n - h < n \rightarrow \; k < h < n. \tag{21.5}$$

In a functional decomposition, the minimization of the value of k is important.

Decomposition can be applied iteratively to a sequence of residual functions with a decreasing number of variables. The method to decompose multiple-output logic functions by means of the BDD for characteristic function [6] required large data structures. In this section we will present a more efficient method of iterative disjunctive decomposition based on notion of blankets [5] simplified for the iterative removal of a single input variable ($|U| = 1$).

Instead of the exact formulation of a decomposition algorithm, we prefer to illustrate it on a small example, an incomplete function F_4 in Table 21.1. Let us note that a set of (n+m)-tuples may not always define a Boolean function, because it is possible to assign conflicting output values. Acceptable functions must satisfy the consistency condition, which guarantees that there are no contradictions; shortly, if two input cubes are compatible, their corresponding output cubes must also be compatible.

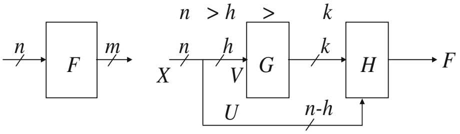


Fig. 21.1 Disjunctive decomposition of multiple output Boolean function F of n variables

Table 21.1 Cube specification of function F_4

	x1	**x2**	**x3**	**x4**	**y1**	**y2**
1	0	0	–	0	1	1
2	1	0	–	0	1	0
3	–	0	0	–	1	~
4	–	–	1	1	0	~
5	–	1	1	0	0	0
6	–	1	–	1	~	1
7	0	–	0	1	1	~

For the 1st step of iterative decomposition we will need two-block blankets, $\beta_1, \beta_2, \beta_3, \beta_4$ for input variables x_1 to x_4:

$$\begin{aligned} \beta_1 = \{\underline{1, 3, 4, 5, 6, 7;}\ \underline{2, 3, 4, 5, 6}\}\, \beta_2 &= \{\underline{1, 2, 3, 4, 7;}\ \underline{4, 5, 6, 7}\} \\ \beta_3 = \{\underline{1, 2, 3, 6, 7;}\ \underline{1, 2, 4, 5, 6}\}\ \beta_4 &= \{\underline{1, 2, 3, 5;}\ \underline{3, 4, 6, 7}\}. \end{aligned} \tag{21.6}$$

Blankets consist of subsets (blocks) of cubes denoted by line numbers in Table 21.1. The first block in each blanket includes cubes which contain "0" or "–" in place of variable x_i, cubes in the second block have value "1" or "–" in place of variable x_i.

A single variable will be removed from the function in one decomposition step. We will select randomly variable $\{U\} = x_3$ for the first step; optimization of variable ordering will be discussed later on. We first create the input blanket β_V as an intersection of two-block blankets for the subset $V = \{x_1, x_2, x_4\}$ of the remaining variables:

$$\beta_{\text{V}} = \beta_1 * \beta_2 * \beta_4 = \{\underline{1, 3;}\ \underline{3, 4, 7;}\ \underline{5;}\ \underline{4, 6, 7;}\ \underline{2, 3;}\ \underline{3, 4;}\ \underline{4, 6}\}. \tag{21.7}$$

The main task in a serial decomposition of a function F with given sets $U = \{x_3\}$ and $V = \{x_1, x_2, x_4\}$ is to find a blanket β_{G1} by merging blocks of β_V as much as possible. A condition for two blocks be mergeable is given in [5] as

$$\beta_{\text{U}} * \beta_{\text{G1}} < \beta_{\text{F}}, \tag{21.8}$$

where β_F is the blanket with blocks related to various output cubes $y_1y_2 = 00, \ldots,$ 11 and cubes with them compatible; in our example $\beta_F = \{\ \underline{4,5};\ \underline{4,6};\ \underline{2,3,7};\ \underline{1,3,6,7}\ \}$.

By merging blocks we want to obtain the minimum number of new larger blocks so that $\beta_{G1} \geq \beta_V$. In our example seven blocks in blanket β_V can be merged to four blocks of β_{G1}

$$\beta_{\text{G1}} = \{\underline{1, 3;}\ \underline{5;}\ \underline{3, 4, 6, 7;}\ \underline{2, 3}\} \tag{21.9}$$

and encoded arbitrarily with two bits—G_1 outputs, see Table 21.2. For example blocks $\underline{1,3}$ and $\underline{2,3}$ are *unmergeable* because if we merge them, β_{G1} and $\beta_3 * \beta_{G1}$ would contain block $\underline{1,2,3,}$ but this block is not covered in β_F. Thus condition (21.8) could not be satisfied. However, we can merge blocks $\underline{3,4}$; $\underline{3,4,7}$; $\underline{4,6,7}$; $\underline{4,6}$ and create block $\underline{3, 4, 6, 7}$, because

$$\begin{aligned} \beta_3 * \beta_{\text{G1}} &= \{\underline{1, 3;}\ \underline{3, 6, 7;}\ \underline{2, 3;}\ \underline{1;}\ \underline{5;}\ \underline{4, 6;}\ \underline{2}\} \leq \beta_{\text{F}} \\ &= \{\underline{4, 5;}\ \underline{4, 6;}\ \underline{2, 3, 7;}\ \underline{1, 3, 6, 7}\}. \end{aligned} \tag{21.10}$$

The minimal cardinality of blanket β_{G1} ensures that parameter k in Fig. 21.1 has the smallest possible value ($k = log_2|\beta_{G1}| = 2$). Function G_1 in our example is specified in Table 21.2 by four blocks (21.9) of blanket β_{G1}.

Construction of function H_1 follows from the blanket $\beta_3 * \beta_{G1}$.The truth table of function H_1 is given in Table 21.3.

Table 21.2 Cube specification of function G_1

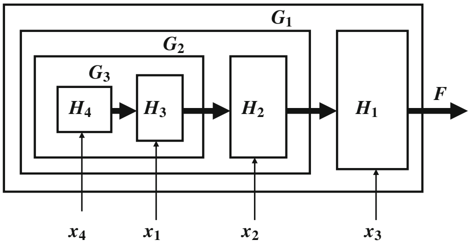
	β_v	β_{G1}	x_1	x_2	x_4	G_1
1	$\underline{1,3}$	$\underline{1,3}$	0	0	0	00
2	$\underline{3,4,7}$	$\underline{3,4,6,7}$	0	0	1	01
3	$\underline{5}$	$\underline{5}$	–	1	0	10
4	$\underline{4,6,7}$	$\underline{3,4,6,7}$	0	1	1	01
5	$\underline{2,3}$	$\underline{2,3}$	1	0	0	11
6	$\underline{3,4}$	$\underline{3,4,6,7}$	–	0	1	01
7	$\underline{4,6}$	$\underline{3,4,6,7}$	–	1	1	01

Table 21.3 The truth table of function H_1

$\beta_1 * \beta_{G1}$	x_4	G_1	H_1
$\underline{1,3}$	0	00	11
$\underline{3,6,7}$	0	01	11
$\underline{2,3}$	0	11	10
$\underline{1}$	–	00	11
$\underline{5}$	1	10	00
$\underline{4,6}$	1	01	01
$\underline{2}$	–	11	10

In the 2nd decomposition step we could apply the same procedure to function G_1, removing for example variable x_2 and looking for the input blanket β_V for the subset $V = \{x_1, x_4\}$, and so on. Fig. 21.2. shows the iterative decomposition up to the last variable x_4. The original function F can be implemented as a cascade of LUTs defined by functions H_i. To reduce the cascade length and thus the overall serial access time to LUTs, we can combine several consecutive LUTs into a single LUT. Also, it makes sense to continue decomposition only to the point where the last module H_k has just one more input than outputs, see Fig. 21.1 and (21.5). For example, function F in Table 21.1 can be implemented as two LUTs specified by functions G_1 and H_1, each with 8 words 2-bit wide.

LUT cascade composed of p-input/q-output LUTs can be stored in one memory. Cascaded LUTs are accessed one by one under a supervision of a HW- or SW-based controller. Outputs from the previous LUT and external input(s) together address



$F = H_1\,(x_3, G_1)$
$G_1 = H_2\,(x_2, G_2)$
$G_2 = H_3\,(x_1, G_3)$
$G_3 = H_4\,(x_4)$

Fig. 21.2 Disjunctive decomposition of multiple output Boolean function F of 4 variables

the next LUT, until the last LUT is reached. LUTs are stored in a RAM and can be changed at will. Even if the memory is accessed once for each LUT in the cascade, operation is approximately ten times faster than branching programs [1].

21.4 Logic Functions and LUT Cascades

The most important parameter of logic functions targeted for cascade implementation is their *profile*. It is cardinality of blanket β_{Gi} along the LUT cascade. For general, multiple-output Boolean functions the following lemma holds true [7]:

Lemma 1 *Let* $F : (Z_2)^n \rightarrow Z_R$ *be the fully specified function of binary variables* $x_1, x_2, \ldots, x_n$. *Then* $|\,\beta_F| = R$ *and*

$$\left|\beta_{G_i}\right| \leq \min(2^{n-i}, R^{2^i})\, \mathit{for}\ i = \mathit{1},\ \mathit{2},\ \ldots, n - \mathit{1}. \tag{21.11}$$

The random fully specified functions have the profile (the upper bound) in the shape of a mountain peak with slope that rises as powers of 2 at the beginning of the cascade and descends much faster as 2^i-powers of R at its end. E.g. the profile of a general function with $n = 20$ input variables and 2 output variables ($R = 4$) is illustrated in Fig. 21.3.

Example The profile of a general $R = 4$-valued function of 12 variables is according to above Lemma upper bounded by

$$2,\ 4,\ 8,\ 16,\ 32,\ 64,\ 128,\ 256,\ 512,\ 256,\ 16,\ 4. \tag{21.12}$$

The upper bounds on profile are too weak for most of real-life functions in digital engineering practice. Very often the functions are defined only in a subset of all 2^n binary input vectors and have typically low profiles. We will therefore study this important class of functions.

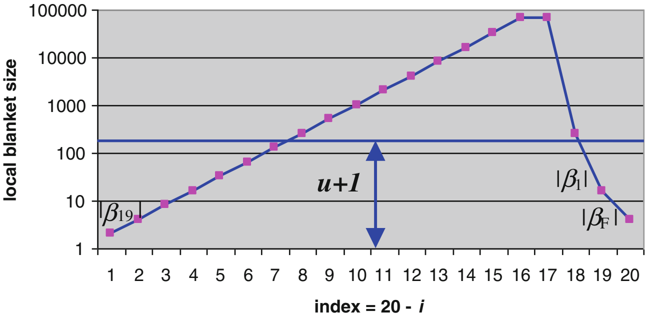


Fig. 21.3 Profile of general function and its limiting asymptote

Definition 5 Let the function F_n attains non-zero values 1, 2, ..., k in $|X| = u$ binary vectors, $X \subset (Z_2)^n$, $k \leq u << 2^n$ and is zero or not defined elsewhere (all binary outputs ~) . The *weight* of function F_n, denoted by u, is the cardinality of X, $u = |X|$.

Theorem 1 [7] *The profile of function* F_n:$(Z_2)^n \rightarrow$ {1, 2, ..., k, ~ } *specified by weight u is upper-bounded by*

$$\left(2,\ 4,\ 8,\ \ldots,\ 2^h, u+1,\ u+1,\ u+1,\ \ldots, u+1\right), \tag{21.13}$$

where $h = \lfloor log_2(u+1) \rfloor$.

As the above theorem applies to arbitrary functions, optimization of variable order is not so important; functions with the weight in the range $2^i \leq u \leq 2^{i+1}$-1 are all realizable by the same LUT cascade. Modification of such functions, such as increasing or decreasing the weight, is easy by re-loading LUTs. Moreover, on the base of Theorem 1, it is possible to estimate the size of LUTs for these functions and figure out the optimum clustering of input variables without a search for optimum variable ordering. Any order of variables leads to the profile (21.13).

Example The upper bound on profile (21.12) will change to

$$2,\ 4,\ 8,\ 16,\ 32,\ 64,\ 64,\ 64,\ 64,\ 64,\ 16,\ 4 \tag{21.14}$$

if the function is specified in not more than $u = 63$ and not less than $u = 32$ input vectors, $32 \leq u \leq 63$.

The $\log_2|\beta_{Gi}|$ gives the number of binary values transferred between neighbor LUTs. To minimize the total size of all LUTs in a cascade, it is thus necessary

1. to minimize the values in the profile, especially the value of $w = \max\{|\beta_{Gi}|\}$;
2. to combine consecutive LUTs in an optimal way.

As regards the first option, the program tool HIDET1 (Heuristic Iterative Decomposition Tool) was developed to aid LUT cascade synthesis [4]. The HIDET1 made use of iterative decomposition of multiple output fr Boolean functions specified by cubes with the restriction that input cubes must be disjoint and output cubes are only binary (elements 0 and 1). This restriction has been removed in HIDET3 used at present: don't cares are allowed in output cubes and input cubes may overlap (share one or more input vectors). It also uses a variable-ordering heuristics to order variables optimally, because the ordering of variables may sometimes influence the profile dramatically [8]. It is worth to do such optimization, especially when the value of w, the maximum profile width, is slightly above 2^i. By optimization it may drop under 2^i and ensure a huge saving of memory.

The key procedure in HIDET3 is merging blocks of blanket β_V. It is accomplished by examining all pairs of blocks in blanket β_V and testing their mergeability. We construct a conflict graph $\Gamma(N, E)$, where nodes (N) are blocks of β_V and edges (E)

connect unmergeable block pairs. By coloring this graph and by joining nodes of the same color we obtain blocks of blanket β_{Gi}. The local blanket size $|\beta_{Gi}|$ is the chromatic number of the graph $\Gamma(N, E)$. The procedure is repeated for each step of the iterative decomposition.

Another optimization tool has been developed for clustering LUTs in the cascade, which explores all possible groupings of n inputs, $n \leq 32$. The input to this tool is a profile of the given function obtained by HIDET. The output is a set of segments of ordered input variables according to one of three optional optimization criteria: searching a minimum of

- the memory area, regardless the number of LUT inputs;
- the product of memory area and cascade length;
- the memory area when the number of LUT inputs is the given value N or less.

21.5 Experimental Results

Two types of functions have been explored: functions specified by weight and the real-world function implemented in MCS 51 microcontroller as a PLA.

In the first case, experiments have been done on benchmark index-generating functions (i.e. those ones specified by weight $u = k$ in Definition 5) with $n = 10$, 16 and 20 variables. Optimum profiles of these functions have been found by HIDET tool and are given in Table 21.4. The optimum LUT cascades for random index-generating functions are listed in Table 21.5. The table gives memory requirements for the cascades with only a single cell and then the fraction of this memory M in % needed when several (#LUTs) are used in place of a single LUT: two cells (#LUT=2), generic cascades with n cells (#LUT=in) and cascades optimized for memory area or for the memory area—cascade length product. The interesting result is that the memory consumption has a local minimum for #LUTs $< n$. The generic cascades with the finest granularity (#LUTs = n) are not optimal in this respect.

The biggest drop in memory area comes from dividing a single cell into two. Further benchmark-specific subdivision of cascades beyond 4 cells produces only a small reduction in memory area, but after some point it goes up again. This typical trend is illustrated on the example of lrs6 benchmark with 21 input variables in Fig. 21.4. Optimization for area-time product leads to slightly shorter cascades (2–6 cells) and slightly larger memory area.

The second example has to do with LUT cascade replacing 13-input, 8-output PLA1 in MCS-51 chip. Here $u = 175$, but only logic equations were known, see the Appendix. Equations have been converted to PLA matrix in f format via eqntott tool and further converted to fr format by means of Espresso synthesizer. The resultant matrix of 147 cubes has been processed by HIDET3 and the following profile has been obtained:

$$2,\ 4,\ 8,\ 14,\ 25,\ 41,\ 62,\ 81,\ 88,\ 103,\ 90,\ 77,\ 256. \qquad (21.15)$$

Table 21.4 Optimum profiles of index generating functions ($n = 10, 16, 20$ and values of u as shown)

n10_$_{\mathrm{U31}}$	2	4	8	14	20	24	27	29	30	32									
n10_$_{\mathrm{U63}}$	2	4	8	15	26	37	43	51	58	64									
n10_$_{\mathrm{U127}}$	2	4	8	16	30	54	80	101	116	128									
n16_$_{\mathrm{U31}}$	2	4	7	11	18	22	26	28	29	30	31	31	31	31	31	32			
n16_$_{\mathrm{U63}}$	2	4	8	14	25	35	44	51	54	59	61	62	63	63	63	64			
n16_$_{\mathrm{U127}}$	2	4	8	16	30	50	69	90	103	111	116	120	122	124	126	128			
n20_$_{\mathrm{U31}}$	2	4	8	11	15	19	24	27	28	29	30	30	30	31	31	31	31	31	32
n20_$_{\mathrm{U63}}$	2	4	8	15	25	37	45	51	54	58	60	61	62	63	63	63	63	63	64
n20_$_{\mathrm{U127}}$	2	4	8	16	29	49	71	89	99	110	116	120	122	125	126	127	127	127	128

Table 21.5 Memory requirements of LUT cascades for benchmark functions with 1, 2, in, and optimum number of cells with respect to memory area or product memory * speed (#LUTs)

			total LUT memory in bits, % of a single LUT							
			# LUT = 1	#LUT = 2	#LUT = in		memory		memory * speed	
name	in	out	M [b]	M [%]	#L	M [%]	#L	M [%]	#L	M [%]
n10_U31	10	5	5,120	37.50	10	36.29	3	31.25	2	37.50
n10_U63	10	6	6,144	50.00	10	60.45	2	50.00	2	50.00
n10_U127	10	7	7,168	75.00	10	96.46	2	75.00	2	75.00
n16_U31	16	5	327,680	4.69	16	1.15	6	1.07	4	1.37
n16_U63	16	6	393,216	6.25	16	2.12	5	1.95	4	2.34
n16_U127	16	7	458,752	9.38	16	3.85	5	3.52	3	4.69
n20_U31	20	5	5,242,880	1.17	20	0.09	8	0.09	5	0.12
n20_U63	20	6	6,291,456	1.56	20	0.18	7	0.17	5	0.22
n20_U127	20	7	7,340,032	2.34	20	0.34	7	0.32	5	0.39

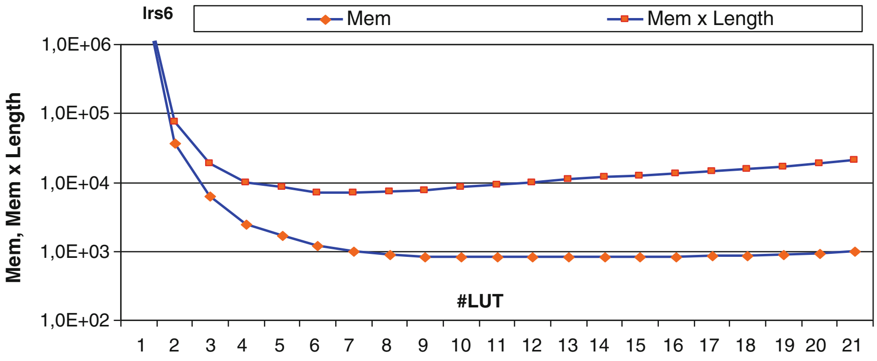


Fig. 21.4 Memory area and memory area times cascade length vs the number of LUTs

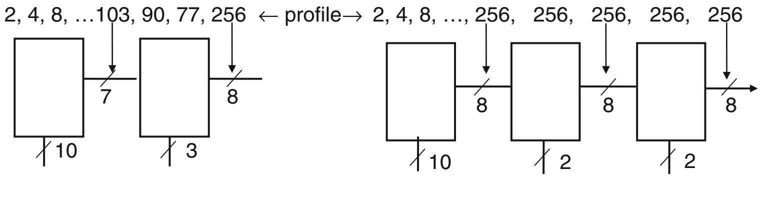

2, 4, 8, …103, 90, 77, 256 ← profile→ 2, 4, 8, …, 256, 256, 256, 256, 256
7
8
10
3
8
8
8
10
2
2

Fig. 21.5 Two decompositions of PLA1 MCS-51

By inspection, this profile suggests 2 LUTs, first one indexed by 10 variables, the second one by 7 outputs from the first LUT (given by $|\beta_{G3}| = 103$) plus 3 remaining variables, Fig. 21.5. On the other hand, had we relied on Theorem 1, we would need 3 LUTs with and 10, 2 and 1 input variable and with 8 output variables each. However, with 3 such LUTs we are able to realize any function of $n = 14$ variables with $u \leq 256$, Fig. 21.5.

21.6 Conclusions

The decomposition technique based on blankets has been found quite suitable for engineering applications, such as designing application-specific systems. It has been successfully applied to random functions specified by weight, for which the size of the LUT cascade is computable beforehand, and to PLA1 used in microcontroller MCS 51. Output vectors are computable by few accesses to LUTs.

The HIDET3 tool used for decomposition has no restrictions on input functions; scalability is at present limited to functions with around 20 input variables, but the work on extending this range is in progress. One way to reduce complexity is partitioning of binary outputs and parallel execution of resulting LUT cascades. The future research should address this issue as well as optimal LUT packing into a single memory module when the size of LUTs in the cascade is not uniform.

Acknowledgments This research has been carried out under the financial support of the research grants “Natural Computing on Unconventional Platforms”, GAČR GP103/10/1517, and “Security-Oriented Research in Information Technology”, the research plan MSM0021630528.

Appendix: Description of PLA1 in MCS-51

```
INORDER = A B C D E F G H I J K L M ;
SO = !A&!G&!I&J&M | A&!B&!I&J&M | A&F&!I&M;
CS = !A&!B&D&!E&!F&!G&!H&!I&!K&!L&M |
   A&B&!E&!F&!G&!H&!I&!J&!K&!L&!M | !A&!E&!I&M | !E&!I&J&M | !D&!I&M ;
BL = !B&E&!F&!G&!H&!I&!J&!K&!L | !B&C&!D&!H&!I&!J&M | !B&D&E&!H&!I&!J&!M |
   !D&!I&!J&K&M | !A&!G&!I&J&M | E&H&!I&!L&M |
   C&!D&G&!I&M | !A&F&!I&M | G&!I&K&M | E&G&!I&M ;
NL = !B&E&!F&!G&!H&!I&!J&!K&!L | C&!D&!H&!I&L&M | !D&!I&!J&K&M | !A&!G&!I&J&M|
   D&E&!H&!I&M | !A&F&!I&M | E&!I&!L&M | G&!I&K&M ;
V1 = !A&!G&!I&J&M | C&!D&F&!I&M
   | A&!B&!I&J&M | !A&F&!I&M | F&!I&K&M | E&F&!I&M ;
V3 = !B&!C&!D&E&!F&!G&!H&!I&!J&!K&!L | !B&!G&!I&J&K&M |
   !D&!I&!J&K&M | B&C&!I&K&M ;
V4 = !B&C&!D&E&!F&!G&!H&!I&!J&!K&!L |
   !B&D&E&!F&!G&!H&I&!J&!K&!L&M | !A&!G&!I&J&L&M |
   C&!D&!H&!I&L&M | !A&F&!I&L&M | C&!D&H&!I&M | D&E&!I&L&M ;
V5 = !B&D&E&!F&!G&!H&I&!J&!K&!L&M |
   !B&E&!F&!G&!H&!I&!J&!K&!L | C&!D&!H&!I&L&M |
   !D&!I&!J&K&M | !A&!G&!I&J&M | C&!D&H&!I&M |
   A&!B&!I&J&M | D&E&!I&L&M | !A&F&!I&M | E&!I&!L&M ;
```

References

1. Sasao, T., Matsuura, M., and Iguchi, Y.: A cascade realization of multiple-output function for reconfigurable hardware. In: International Workshop on Logic and Synthesis IWLS01, pp. 225–230. Lake Tahoe, CA (2001)
2. Nakahara, H., Sasao, T., and Matsuura, M.: A comparison of architectures for various decision diagram machines. In: International Symposium on Multiple-Valued Logic, pp. 229–234. IEEE CS Press, Barcelona (2001)
3. Nakahara, H., Sasao, T., Matsuura, M., Kawamura: A parallel branching program machine for sequential circuits: Implementation and evaluation. IEICE Transactions on Information and Systems, E93-D, pp. 2048–2058 (2010)
4. Mikušek, P., Dvořák, V.: On lookup table Cascade-based realizations of arbiters. In: Proceedings of the 11th EUROMICRO Conference on Digital System Design DSD, pp. 795–802. IEEE CS Press, Parma (2008)
5. Brzozowski, J. A., Luba, T.: Decomposition of boolean functions specified by cubes. Research report CS-97-01, University of Waterloo, Canada (1997)
6. Nakahara, H., Sasao.: A method to decompose multiple-output logic functions. In: 41st Design Automation Conference DAC, pp. 428–433. IEEE Press, San Diego (2004)
7. Dvořák, V., Mikušek, P.: On the cascade realization of sparse logic functions, In: Proceedings of the 14th EUROMICRO Conference on Digital System Design DSD, pp. 21–28. IEEE CS Press, Oulu (2011)
8. Drechsler, R., Becker, B.: Binary Decision Diagrams-Theory and Implementation. Kluwer Academic Publishers, Boston (1998)

Part IV
Industrial Engineering, Production and Management

Chapter 22
Intelligent Non-split Mixing Decisions: A Value-Engineering Tool Applied to Flour Mills

Jürg P. Keller and Mukul Agarwal

Abstract Many manufacturing processes involve using a multitude of intermediate products to make a few final products. The decision regarding which intermediate products go to make which final product involves engineering the value of the final products with respect to several properties and the amounts. Even when each property "mixes linearly," this decision is complex in cases where no mathematical function is known for the relationship between the value of a product and its properties. The complexity of a tool to support this decision making is further aggravated in cases where an intermediate product cannot contribute to more than one final product, e.g. due to mechanical limitations, process constraints, logistics restrictions, or traceability considerations. For this situation, an interactive decision-support tool is developed, and applied to the sensitive example of flour mills, where up to 80 intermediate products, 6 final products, and 6 properties are involved during continuous mixing of flours. The tool allows the head miller to flexibly specify the feasible space in the dimensions of decision variables, properties, and amounts. For any change in this specification, the tool computes and presents without prohibitive time lag a convenient overview of all relevant non-inferior solutions, to facilitate selection of a particular solution. The head miller makes specifications and selects a solution for one final product at a time, usually starting with the most valuable product, but can iterate back to any product at will. Better and more reliable mixing decisions are achieved with the support of the tool.

Keywords Decision support systems · Intelligent manufacturing · Discrete-event systems · Systems modelling and simulation · Logistics engineering · Cost and value engineering · Production planning · Quality control and management · Performance evaluation and optimization · Human-system interface

J.P. Keller (✉)
Institute for Automation, University of Applied Sciences and Arts Northwestern Switzerland,
Klosterzelgstrasse 2, Windisch 5210, Switzerland
e-mail: juerg.keller1@fhnw.ch

M. Agarwal
Corporate Technology, Bühler AG, Uzwil 9240, Switzerland
e-mail: mukul.agarwal@buhlergroup.com

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_22

22.1 Introduction

Modern flour mills usually process a mixture of different wheat grains through about a dozen processing stages connected in series and parallel. In each stage, grain particles are ground further and the resulting particles are separated into various streams, some of which go on to further processing stages while some others are withdrawn as intermediate product streams. This generates about 30–80 intermediate streams with different physical properties and yields. These intermediate product streams are mixed in continuous operation to generate a mere 4–6 final product streams, which are stored in product silos and then packed for shipping.

The decision, how to mix the many intermediate streams into a few product streams, is crucial for generation of maximal sales value while accounting for many constraints on the physical properties and yields of the final streams, as well as on the mixing decision variables. Since usually the sales value of the final streams is not known as a function of their physical properties, but can only be judged qualitatively by the production personnel, tools are needed to guide the decision process by showing overviews of the compromises involved in the various scenarios and letting decisions be made successively and iteratively for each final stream. This involves prolonged interactive use of the tool until the final decision is reached. The tool must therefore not only offer an intuitive and convenient graphic interface, but also be computationally fast in order to show scenario overviews and to allow manoeuvring through scenarios without debilitating time lag.

In a previous work [5], the authors presented a tool that solved the above challenges and was successfully tested in a real flour mill. The tool dealt with the computational speed by using multiple Linear Programming optimizations implemented with an accelerated algorithm. However, this solution is not applicable to a majority of flour mills where each intermediate stream is completely diverted to a single final stream, and no splitting of an intermediate stream to multiple final streams is possible due to constraints of the physical flaps installed in the piping. In this work, therefore, a new tool is developed for the situation involving integer mixing-decision variables.

For the mixed-integer programming problem at hand, many well-known algorithms already exist, in principle. Some of these algorithms are summarized in [2]. Employing these algorithms to the problem at hand would mean optimizing up to 480 binary variables (for upto 80 intermediate streams multiplied by up to 6 final streams). Since, for binary variables, the Linear Programming algorithm must be coupled with a Branch-and-Bound procedure, this high number of binary variables involves prohibitive computational time. To circumvent this drawback, the problem is solved here not for all final streams together, but for one final stream at a time. This decoupling is feasible and sensible for the problem at hand, because the production personnel needs to specify physical-property constraints for each final stream, and prefers to do this only by specifying them (and obtaining results) for one final stream at a time, starting with the most valuable final stream.

Solving the decoupled problem for one single final stream is similar to the problem addressed in recent literature for distributed power-network operation [1]. The latter, however, deals only with power as a single "physical property" and with a-priori known, fixed constraints for this physical property, whereas the problem at hand involves up to 6 different physical properties, and, more importantly, their constraints are not known and cannot be specified a priori. The main special feature of the problem at hand is that the user needs to see the Pareto-optimal solution for *all* possible physical-property constraints, so that he can specify (actually, select) the constraints in an informed manner, knowing the feasibility and the influence of his choice on the final solution.

Precisely this special feature makes the use of existing solutions computationally prohibitive for the current problem. Computing and displaying all binary solutions for the entire range of physical-property constraints would be extremely time consuming. To circumvent this issue, a new solution strategy is used in this work that presents to the user not the binary solutions, but instead a continuous-solution space for the entire range of constraints. Computation of the continuous-solution space involves only Linear Programming, but no Branch-and-Bound procedure, and is consequently fast. The user can then conveniently analyse the solution space and specify a (usually much narrower) range of feasible constraints that she wants to focus on. The solution procedure then generates all binary solutions in vicinity of the continuous solution.

The new tool circumvents the prohibitive computational burden for the mixed-integer programming problem by dividing the problem into several sub-problems, by exploiting the continuous-decision-variable solution as an anchor, and by using an efficient procedure that employs Linear Programming and Branch-and-Bound methods intermittently to obtain binary solutions in the vicinity of the anchor. A special graphic interface is additionally designed to present flexible overviews of resulting scenarios to the user. The new tool allows fast, interactive decision making for the considerably more challenging situation of discrete decision variables. The tool was tested successfully on real plant data.

22.2 Problem Formulation

The production personnel in modern flour mills is faced with, among others, the following crucial decision. For each of the 30–80 intermediate product streams withdrawn continuously from the mill, the personnel has to decide, which of the 4–6 final product streams it should be diverted to. In a majority of flour mills, this is an integer decision that can take up to 6 values for each intermediate stream (see Fig. 22.1).

The following facts render this decision difficult. All intermediate and final streams are characterized by identical physical quality parameters, 4–6 in number. These properties are independent of each other. Each property "mixes" linearly with respect to the weight fractions. The yields (or flow rates) and the physical properties of the intermediate streams are given, whereas those of the final streams depend on the decision and are subject to various constraints. For example, a particular final stream may represent a high-purity flour constrained by an upper limit on the physical property "ash content". Another final stream may have a low yield limit due to

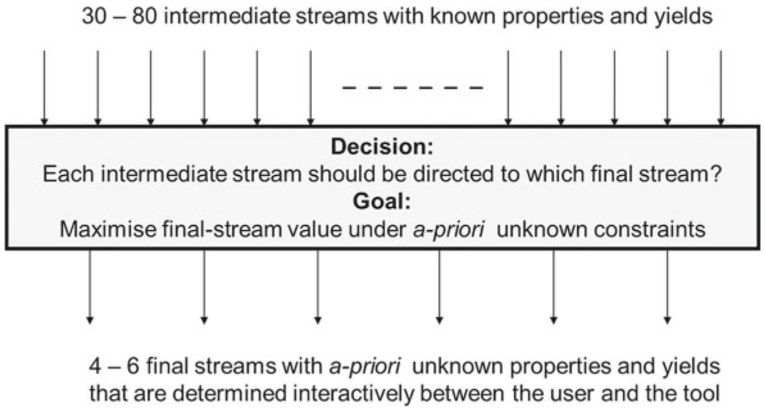

30 – 80 intermediate streams with known properties and yields
Decision:
Each intermediate stream should be directed to which final stream?
Goal:
Maximise final-stream value under *a-priori* unknown constraints
4 – 6 final streams with *a-priori* unknown properties and yields that are determined interactively between the user and the tool

Fig. 22.1 The decision problem

an already known sales to a particular customer, or a high yield limit due to physical limitations of piping, silo, or inventory. In addition, the flaps used to divert an intermediate stream to a particular final stream may have limitations in that a flap can physically output only to certain final streams or quality constraints forbid it to output to certain final streams. The decisions for all intermediate streams, all final streams, all physical properties, and all yields are interconnected in a complex way.

The decision making is complicated further by the fact that, ultimately, the final streams should have the highest possible sales value, but characterization of this value is elusive, since the function relating the sales price to the physical properties is not fixed, or at least not known. The user can only judge this value in a qualitative, comparative way.

Obviously it is not possible even for an expert personnel to make this decision in a near optimum way. The decision is therefore usually determined by on-line trial-and-error based on experience. It is far from trivial to develop an appropriate decision-support tool for this decision, since the tool must optimize possible solutions for the entire range of goal specifications, present all options and compromises to the decision-maker in facilitating overviews allowing for multiple dimensions, conserve maximum flexibility for further steps while making decisions in a particular step, and recompute and redisplay all outputs without prohibitive time lag when any user-input is changed.

A tool for supporting the above decision faces several challenges.

1. *Qualitative Judgement of Value*: Since the value of flours cannot be formulated explicitly as a function of the flour properties and yields, a decision-support tool cannot simply output a unique optimal answer. Instead, the tool must present all non-inferior outcomes and compromises.
2. *High Dimensional Space*: Since large numbers of properties, yields, intermediate streams and final streams are involved in the decision-making, the resulting high

dimensionality is challenging for the tool not only computationally, but also in view of the 2D graphic interface.

3. *Need for Quick Reaction*: The entire decision-making with the support of the tool involves scores of clicks and inputs from the user, while each time the user appraises the tool output before making the next click or input. The tool must therefore respond within 15–20 s after each click or input, so that the new result is quickly visible for appraisal. Even then, the user needs dozens of minutes of interactive use to arrive at the complete final solution.

22.3 Solution Procedure

In principle, for the given problem, the tool must calculate and display all discrete non-inferior solution points, covering all intermediate and final streams simultaneously. This is a formidable task that would require unimaginably high computation time, even using super computers.

22.3.1 Solution Strategy

A new solution strategy is developed that avails itself of several accelerating improvements, so as to achieve computation times of 15–20 s for displaying the needed results.

22.3.1.1 Stepwise Decision for Each Final Stream

The decision in the above mixed-integer problem involves up to 80 decision variables (one for each intermediate stream), each of which can take up to 6 discrete values (one for each final stream). The mixed-integer problem is broken down to up to 6 sub-problems (one for each final stream). In each sub-problem, each of the up to 80 decision variables can take only 2 discrete values (yes/no, indicating whether that intermediate stream is directed to the sub-problem final stream or not). The massively complex mixed-integer problem is thus reduced to a few much simpler, mutually decoupled binary problems. The consequence of this reduction is that the decisions for each final stream can be made only successively, and not all at once.

It turns out that, as in the previous work [5], this poses no limitation for the user, but is indeed in line with what the user would prefer anyway. Due to the complexity of the given problem, the user cannot simultaneously make decisions for all final streams, even with the help of a tool. The decision-making is therefore preferably performed in steps, one step for each final stream, usually beginning with the most valuable flour stream and ending with the least valuable. The above problem reduction allows the user to make a step decision based on the results of previous step decisions, as well as to revert to and re-adjust a previous step decision.

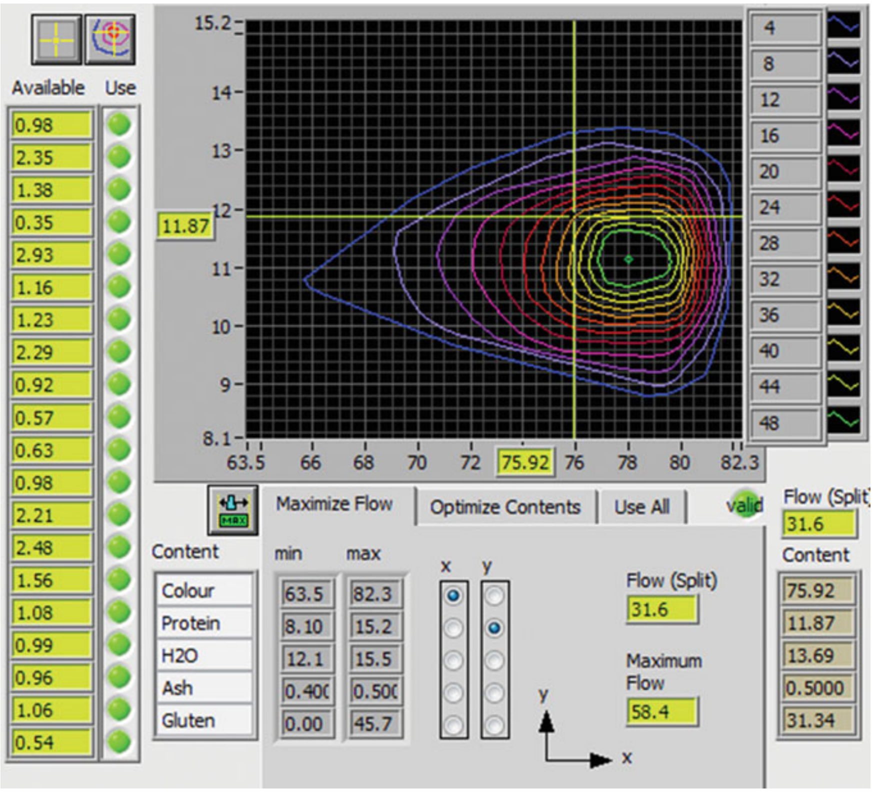
22.3.1.2 Continuous Solution as Anchor

Even for a reduced sub-problem (for a single final stream) as stated above, the number of possible combinations of all binary decision variables is formidably large. Computing all non-inferior combinations for a sub-problem is therefore not practical. A further insight into the problem leads to a considerable relief in this respect.

Even a cursory reflection of the problem reveals that the final solution of the binary-variable problem must lie close to the solution of the continuous-variable problem in the decision-variable space. This holds not only from the process point-of-view, but also from the viewpoint of the user. From the process aspect, a unique optimal solution does not even exist a priori, since the judgment of "value" is merely qualitative and comparative, but not absolute, i.e. the "optimal" solution is essentially chosen by the user as a best-judged compromise. From the user viewpoint as well, it is easier to first "choose" a continuous solution, and then choose an implementable binary solution in its vicinity. The solution strategy thus lets the user first select a sub-problem solution in the continuous space, as in the previous work [5].

Figure 22.2 shows an example selection for the most valuable final stream. Before using the tool, in this case the user measures five properties and the yield for each of 72 intermediate streams in the mill, records this information in an Excel file and with a click reads in this Excel file into the tool. The user then proceeds to make a selection for the most valuable final stream in the shown sub-window where, in an area on the left which can be scrolled, the yields of all the intermediate streams are visible and any of them can be excluded a priori by clicking the "Use" button to red. The user then selects, in a tab labelled "Maximize Flow," the two properties that should be displayed on the x- and y-axis and, if needed, narrows down the allowed range of the other three properties for the final stream under consideration. The tool displays, instantly, a contour plot showing contours with constant yield (also called "flow"), and a point in the middle for the largest possible yield (58.4 in this case). For each point on this diagram with fixed values for the two properties chosen as the axes, the tool maximizes the yield with respect to the values of the other three properties within their specified ranges, and displays this maximum yield at that point as part of a contour. The user can now move a cursor to any desired point on the plot, and sees instantly the attainable yield (31.6 on the chosen point), and all five property values for the selected point (shown on the right-hand-side). Instead of the tab "Maximize Flow", the tab "Optimize Contents" can be used, if the user is willing to compromise on the yield in order to obtain better values for the properties. In that tab (detail not shown in Fig. 22.2), the user sees the best possible property values for each possible value of yield, and can select an informed compromise. The last tab "Use All" allows the user to force all remaining available intermediate streams on the left-hand-side to be used completely for the final stream under consideration (usually the least valuable final stream, which is considered last).

The solution strategy in the current work thus can use the selected point in the continuous space as an anchor, and needs to merely compute all non-inferior binary solutions in a reasonable vicinity of this point. The computational burden is reduced considerably through this anchoring, but is still exorbitant.



Available	Use
0.98	●
2.35	●
1.38	●
0.35	●
2.93	●
1.16	●
1.23	●
2.29	●
0.92	●
0.57	●
0.63	●
0.98	●
2.21	●
2.48	●
1.56	●
1.08	●
0.99	●
0.96	●
1.06	●
0.54	●

Content	min	max	x	y
Colour	63.5	82.3	●	○
Protein	8.10	15.2	○	●
H2O	12.1	15.5	○	○
Ash	0.40(	0.50(	○	○
Gluten	0.00	45.7	○	○

Flow (Split) 31.6

Maximum Flow 58.4

Flow (Split) 31.6

Content

75.92
11.87
13.69
0.5000
31.34

Fig. 22.2 An example of selection of a solution point by the user in the continuous decision space (Figures 22.2 and 22.4 show realistically created artificial data for illustration, since the actual industrial data used for the tests is classified)

22.3.1.3 Linear Programming with Relaxation and Branch-and-Bound

The problem of finding binary solution points in the vicinity of the continuous solution anchor point is not amenable to Linear Programming (LP) solely, since the latter delivers continuous values for the decision variables. Finding binary points in the vicinity would involve setting each continuous-value decision variable to either 0 % or 100 %. This gives a set of branches, for each of which another LP problem with fewer decision variables is solved. The branches with significantly poorer result than the anchor are terminated, and in the remaining branches the decision variables with continuous values are set again to 0 and 100 %, creating the next set of combinatorial branches. This Branch-and-Bound procedure, coupled with an LP solution at each branch, and subsequent reduction in the number of decision variables in each branching stage, is repeated until no more branches are left.

The above procedure is still computationally prohibitive, since the LP solutions at each branch continue to involve a large number of decision variables with continuous values. This large number represents the number of basic variables in the LP problem and equals the number of properties involved plus one yield (i.e. up to 7). The solution strategy therefore resorts to a relaxation method that introduces auxiliary relaxation variables that can take continuous values by becoming basic variables, thereby relieving (at least) a corresponding number of decision variables, which in turn become non-basic, take a discrete 0-or-1 value, and lead to no new branches. In practice, it was noticed that, when yield is included as a relaxation variable, a significantly higher number of decision variables is "relieved" in this way, since not only the auxiliary relaxation variables, but also several other auxiliary variables in the LP formulation then become basic variables.

The relaxation can be introduced for yield and/or property variables, and allows that the value of these variables in the anchor can be violated to a certain extent. This "extent" can be specified with certain weights, which could be optimized or selected by trial-and-error. Two cases were tried in this work: relaxation of yield only, and relaxation of yield as well as all properties. In the first case, the relaxation weights could easily be set by trial-and-error, since their value is not very sensitive with respect to the final result. In the latter case, however, setting the relaxation weights for the properties is much more difficult, not only because of the higher number of coupled weights involved, but more so because the choice of these weights is extremely sensitive with respect to the final result. This solution avenue was therefore dropped in favor of relaxation only with respect to yield.

The above relaxation procedure is a variation of the relaxation methods reported in the literature [3, 7]. In these methods, the Lagrangian relaxation weights serve to penalize the constraint violation, i.e. for the problem at hand, it would penalize the deviation of the yield and property values from the anchor values. In the procedure used above, the relaxation weights serve to reduce the number of decision variables that come out to be basic variables and consequently lead to new branches.

22.3.2 Definitions

In addition, let:

$$\begin{aligned}
\mathrm{W} &= \left[\mathrm{w}_{\mathrm{k,i}}\right], (\mathrm{n}_{\mathrm{k}} \times \mathrm{n}_{\mathrm{F}}) \\
\mathrm{X} &= \left[\mathrm{x}_{\mathrm{k,p}}\right], \left(\mathrm{n}_{\mathrm{k}} \times \mathrm{n}_{\mathrm{p}}\right), \mathrm{X}_{\mathrm{P}} = \left[\mathrm{x}_{\mathrm{k,p}}\right], (\mathrm{n}_{\mathrm{k}} \times 1) \\
\mathrm{F}_{\mathrm{p}} &= \left[\mathrm{F}_{\mathrm{i,p}}\right], (\mathrm{n}_{\mathrm{F}} \times 1) \\
\underline{1}_{\mathrm{m}} &= \left[\,1\ 1 \cdots 1\,\right]^{\mathrm{T}}, (\mathrm{m} \times 1)
\end{aligned} \tag{22.1}$$

n_F	number of intermediate streams
n_P	number of final streams
n_k	number of properties
i	index for intermediate streams
p	index for final streams
k	index for properties
c_i	cost of i-th intermediate stream
r_k	weighting factor for scaling of k-th property
$F_{i,p}$	yield of i-th intermediate stream that is directed to the p-th final stream
$F_{i,max}$	maximum yield of i-th intermediate stream that is available
P_p	yield of p-th final stream
$P_{p,d}$	increment for p-th final stream
$w_{k,i}$	weight fraction of k-th property in the i-th intermediate stream
$X_{k,p}$	weight fraction of k-th property in the p-th final stream
X_d	weight fraction of properties in $P_{p,d}$
S_r	matrix to select streams optimized using LP in the mixed-integer problem
S_f	matrix to select streams with fixed given values that are not optimized
n_f	number of streams with fixed given values that are not optimized
V	vector with fixed given values for streams selected using S_f. (0 = not used, 1 = used)
z_i	fraction of a selected stream, $z_i \in [0, 1]$ with 0 = not used and 1 = used.

22.3.3 LP Formulation for Splittable Intermediate Streams

For the p-th final stream, the basic LP-problem with splittable intermediate streams can be formulated as follows:

$$\begin{aligned} WF_p &= X_p P_p \\ \underline{1}_{n_F}^{T} F_p &= P_p \\ \begin{bmatrix} \underline{1}_{n_F}^{T} \\ W \end{bmatrix} F_p &= \begin{bmatrix} P_p \\ X_p P_p \end{bmatrix} \\ 0 &\le F_{p,i} \le F_{i,max}, \forall i \\ J &= \sum_{\forall i} -c_i F_{p,i} \\ F_P &:= \arg\max J\left(F_p\right) \end{aligned} \tag{22.2}$$

Here, the objective function J is represented as a cost criterion for intermediate streams. The cost factors c_i can thereby represent either actual costs of these flours or the strength of the user's a-priori preference for wanting to use a particular intermediate stream.

$$c_i = \sum_{\forall k} \left| x_{k,p} - w_{k,i} \right| r_k \tag{22.3}$$

The above formulation (Eqs. 22.2 and 22.3) is completely defined and solvable when the yields and properties of the final streams are predefined. But as mentioned above, this is not the case here.

For the mixed-integer optimization in the present situation, it is necessary to redefine the LP formulation such that intermediate streams that are predefined to be zero or $F_{i,max}$ get removed a priori.

$$\begin{aligned} F_{p,r} &= S_r F_p \\ F_{p,f} &= S_f \left(V \circ F_p\right) \end{aligned} \tag{22.4}$$

Here S_r selects streams that can be varied in the range 0 to $F_{i,max}$, and S_f selects streams that are set to 0 or $F_{i,max}$ depending on the value 0 or 1 in the vector V. The LP formulation then becomes:

$$\begin{aligned} \begin{bmatrix} W\, S_r^T & -P_P I_{n_k} \end{bmatrix} \begin{bmatrix} S_r F_p \\ \tilde{X}_p \end{bmatrix} &= X_{p,min} P_p - W S_f^T S_f \left(V \circ F_{max}\right) \\ \begin{bmatrix} \underline{1}_{n_{F,r}}^T & 0 \end{bmatrix} \begin{bmatrix} S_r F_p \\ \tilde{X}_p \end{bmatrix} &= P_p - \underline{1}_{n_{F,f}}^T S_f \left(V \circ F_{max}\right) \\ 0 &\leq F_{p,i} \leq F_{i,max}, \forall i \\ 0 &\leq \tilde{X}_p \leq \tilde{X}_{p,max} \\ J &= \sum_{\forall i} -c_i F_{p,i} \\ \begin{bmatrix} S_r F_p & \tilde{X}_p \end{bmatrix} &:= \arg\max J \left(\begin{bmatrix} S_r F_p & \tilde{X}_p \end{bmatrix}\right) \end{aligned} \tag{22.5}$$

or in matrix form:

$$\begin{aligned} \begin{bmatrix} \underline{1}_{n_{F,r}}^T & 0 \\ W S_r^T & -P_P I_{n_k} \end{bmatrix} \begin{bmatrix} S_r F_p \\ \tilde{X}_p \end{bmatrix} &= \begin{bmatrix} P_p - \underline{1}_{n_{F,f}}^T S_f \left(V \circ F_{max}\right) \\ X_{p,min} P_p - W S_f^T S_f \left(V \circ F_{max}\right) \end{bmatrix} \\ 0 &\leq F_{p,i} \leq F_{i,max}, \forall i \\ 0 &\leq \tilde{X}_p \leq \tilde{X}_{p,max} \\ J &= \sum_{\forall i} -c_i F_{p,i} \\ \begin{bmatrix} F_p & \tilde{X}_p \end{bmatrix} &:= \arg\max J \left(\begin{bmatrix} F_p & \tilde{X}_p \end{bmatrix}\right) \end{aligned} \tag{22.6}$$

22.3.4 LP Formulation with Relaxation

The above formulation (Eqs. 22.6) now needs to be modified to include relaxation for yield, so as to force the streams to be close to 0 or 100 % of the intermediate stream yield (see Sect. 3.1.3). This is achieved by allowing certain tolerances in the constraints. From LP perspective, this creates additional basic variables, such that the streams will take only discrete values at either the upper or the lower bound. For this purpose, define:

$$F_i = z_i \cdot F_{i,max} \tag{22.7}$$

$$P_p = P_{p,soll} + P_{p,+} - P_{p,-} \tag{22.8}$$

where $P_{p,soll}$ represents a given minimal yield that must be reached and $P_{p,+}$ and $P_{p,-}$ represent relatively small corrections to it. The latter appear in the formulation as products with the property weight fractions, leading to a bilinear optimization problem. In order to avoid this bilinearity, the property weight fractions in these cross-terms should be fixed. This can be achieved by iterating the values of the property weight fractions starting from the anchor value.

The following LP formulation then results:

$$\begin{gathered}
\begin{bmatrix} WF_{max,diag}S_r^T & -P_{P,soll}I_{n_k} & -X_d & X_d \end{bmatrix} \begin{bmatrix} S_r z \\ \tilde{X}_p \\ P_{p,+} \\ P_{p,-} \end{bmatrix} = X_{p,min}P_{p,soll} - WS_f^T S_f\,(V \circ F_{max}) \\
\begin{bmatrix} F_{max}^T S_r^T & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} S_r z \\ \tilde{X}_p \\ P_{p,+} \\ P_{p,-} \end{bmatrix} = P_{p,soll} - \underline{1}_{n_{F,f}}{}^T S_f\,(V \circ F_{max}) \\
0 \le z_i \le 1, \forall i \\
0 \le \tilde{X}_{p-f} \le \tilde{X}_{p,max} \\
0 \le P_{p,+} \le P_{max}, \quad 0 \le P_{p,-} \le P_{max} \\
J = \sum_{\forall i} -c_i\,(z)\,F_{i,max} - \xi_{F,+}P_{p,+} - \xi_{F,-}P_{p,-} \\
\begin{bmatrix} S_r z & \tilde{X}_p & P_{p,+} & P_{p,-} \end{bmatrix} := \arg\max J\left(\begin{bmatrix} S_r z & \tilde{X}_p & P_{p,+} & P_{p,-} \end{bmatrix}\right)
\end{gathered} \tag{22.9}$$

or in matrix form:

$$\begin{gathered}
\begin{bmatrix} F_{max}^T S_r^T & 0 & -1 & 1 \\ WF_{max,diag}S_r^T & -P_{P,min}I_{n_k} & -X_d & X_d \end{bmatrix} \begin{bmatrix} S_r z \\ \tilde{X}_p \\ P_{p,+} \\ P_{p,-} \end{bmatrix} = \begin{bmatrix} P_{p,min} - \underline{1}_{n_{F,f}}{}^T S_f\,(V \circ F_{max}) \\ X_{p,min}P_{p,min} - WS_f^T S_f\,(V \circ F_{max}) \end{bmatrix} \\
0 \le z \le 1, \forall i \\
0 \le \tilde{X}_{p-f} \le \tilde{X}_{p,max} \\
0 \le P_{p,d} \le P_{max} \\
J = \begin{bmatrix} -\,(c \circ F_{max})^T S_r^T & 0 & -\xi_{F,+} & -\xi_{F,-} \end{bmatrix} \begin{bmatrix} S_r z \\ \tilde{X}_p \\ P_{p,+} \\ P_{p,-} \end{bmatrix} - (c \circ F_{max})^T S_m^T S_m z \\
\begin{bmatrix} S_r z & \tilde{X}_p & P_{p,+} & P_{p,-} \end{bmatrix} := \arg\max J
\end{gathered} \tag{22.10}$$

22.3.5 *Mixed-Integer Procedure for Non-splittable Intermediate Streams*

In the proposed solution procedure for the mixed-integer problem at hand, the solution to the LP problem in Eq. 22.9 is taken as the anchor starting point to search for binary variable solutions in its vicinity. The above LP problem comprises $\mathrm{n_p} + 3$ equations and $\mathrm{n_F} - \mathrm{n_f} + 2$ variables that determine the split of the intermediate streams. The chosen form for the solution of this LP problem generates additional $\mathrm{n_p} + 3$ auxiliary variables based on the number of equations. When the LP problem has been solved, out of all variables precisely $\mathrm{n_p} + 3$ variables emerge as basic variables and take values that do not lie on the constraints. If one of these $\mathrm{n_p} + 3$ basic variables belongs to the set $\mathrm{z_i}$, then the corresponding intermediate stream gets split.

Since this split is not acceptable for the mixed-integer problem at hand, a combination with the Branch-and-Bound method is used. Branch-and-Bound methods have a long history [6] for solving mixed-integer problems not only in conjunction with Linear Programming using, e.g., rules [4] or heuristics [10], but also for quadratic [9] and nonlinear [8] problems.

Using the Branch-and-Bound method for the problem at hand, when one or more of the $\mathrm{n_p} + 3$ basic variables belong to the set $\mathrm{z_i}$, then several new LP problems are generated using all possible integer-value (0 or 1) combinations as a-priori fixed values of these variables. All these LP problems are then solved, and those branches are discarded that lead to a considerably poorer objective-function value than the source-branch LP solution. Generation of further LP problems with new branching ends when all branches get discarded.

22.3.6 *Formal Solution Procedure*

The above procedure is formalized as follows.

Analogous to the definition of $\mathrm{S_r}$ and $\mathrm{S_f}$, define I_r and I_f as sets of indices for streams that are optimized and that are fixed, respectively, and I as the set of indices for all streams. Define corresponding vectors $\mathrm{z_r}$ and $\mathrm{z_f}$ indicating fractions of the intermediate stream that go into a final stream. The vector $\mathrm{z_f}$ containing fixed streams thus only comprises values 0 and 1, whereas the vector $\mathrm{z_r}$ starts out with real values and approaches the values 0 or 1 during the optimization.

Let the *k*th mixed-integer LP problem, $\mathrm{LP_{m,k}}$, determine $\mathrm{z_r}$, I_r, and the objective function value as:

$$\left(z_{r,k}, I_{r,k}, J_{m,k}\right) = \mathrm{LP}_{m,k}\left(z_{f,k}, I_{f,k}\right) \tag{22.11}$$

In the resulting vector $\mathrm{z_{r,k}}$, the indices for which the value is not 0 or 1 form an index set $I_{r,k}$. Next, all (0,1)-combinations are determined for each $\mathrm{z_{r,k,i}}$ with $i \in I_{r,k}$. This generates a set of vectors $Z_{\mathrm{p,k}}$ that must then be combined with the corresponding fixed-value vector $\mathrm{z}_{\mathrm{f,k}}$, together with its associated index set $I_{f,k}$. The result is a set L_{k+1} of pairs $(\mathrm{z}_{\mathrm{f,k+1}}, I_{f,k+1})$ for $\mathrm{z}_{\mathrm{f,k+1}} \in Z_{\mathrm{p,k}} \otimes \mathrm{z}_{\mathrm{f,k}}$ and $I_{f,k+1} = I_{r,k} \cup I_{f,k}$. Each of these pairs is then solved as an independent LP problem.

This procedure is repeated until all intermediate streams take only the values 0 or 1, i.e., until $I_{f,k+1}$ comprises indices for all intermediate streams.

A flow diagram for the procedure is shown in Fig. 22.3. During the computation, the size of the set L is shown to the user, and the user can adjust the sub-optimality parameter ξ to influence the number of solutions that would be generated.

22.4 Graphic Interface

The graphic interface must present all options and compromises to the user in facilitating overviews. The user begins with a search in the continuous-solution domain in order to specify an anchor, for a particular selected final product stream. This search is guided by well-structured graphic overviews and flexible input possibilities as shown in the previous work [5]. After a continuous-solution anchor has been selected for the final product stream, the user clicks a button to switch to the integer-solution mode, which calculates and displays the mixed-integer solutions in the vicinity of the anchor continuous solution, using the solution procedure described above.

The user interface for displaying these results is shown in Fig. 22.4. Since slightly sub-optimal solutions might get preferred by the user based on considerations not available in the above problem formulation, and since several near-optimal solutions might yield objective-function values quite close to each other, the tool displays all such results in a convenient overview, and allows the user to flexibly and comfortably select the particular solution that is best in view of the other considerations. On the left-hand side of the graphic interface (Fig. 22.4 top), the user sees, for the particular final product stream at hand, which intermediate product streams would be used (value 1, color green, shade light gray) and which would not be used (value 0, color red, shade dark gray). Each relevant solution is thereby shown in a single column. At the bottom of this part of the display, the user can provisionally select one particular solution by clicking it green (or shade light gray).

On the right-hand side of the graphic interface (Fig. 22.4 bottom), the user sees, for the various solutions, the values of the physical properties versus the yield of the final product stream at hand. Thereby, the solution selected on the left-hand side, as well as the anchor solution (as intersecting dash-dotted lines), are highlighted for easier judgment. At the bottom of this part of the display, the user can select a range of yields in order to narrow down the number of solutions that are displayed on the left-hand side.

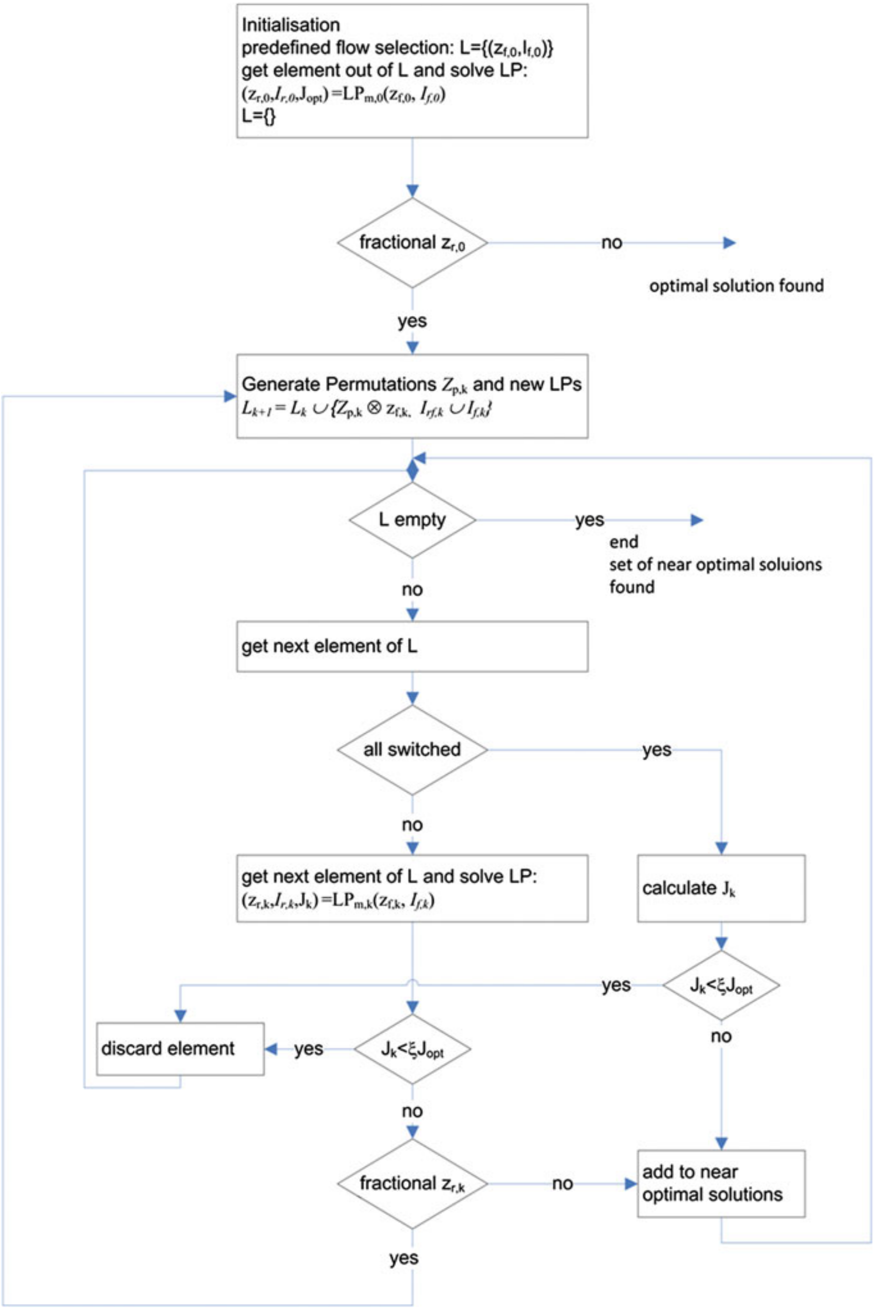
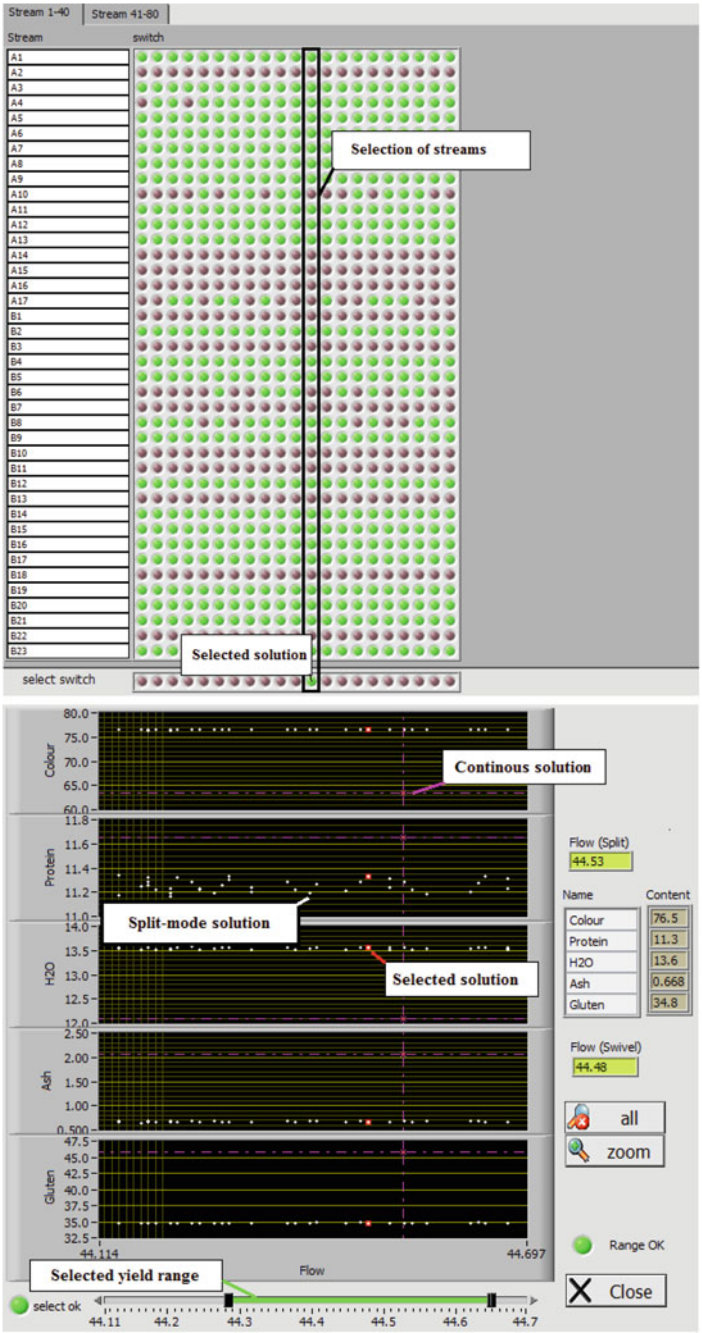

Initialisation
predefined flow selection: L={($z_{f,0}$,$I_{f,0}$)}
get element out of L and solve LP:
($z_{r,0}$,$I_{r,0}$,J_{opt})=$LP_{m,0}$($z_{f,0}$, $I_{f,0}$)
L={}
fractional $z_{r,0}$
no
optimal solution found
yes
Generate Permutations $Z_{p,k}$ and new LPs
$L_{k+1} = L_k \cup \{Z_{p,k} \otimes z_{f,k},\ I_{rf,k} \cup I_{f,k}\}$
L empty
yes
end
set of near optimal soluions
found
no
get next element of L
all switched
yes
no
get next element of L and solve LP:
($z_{r,k}$,$I_{r,k}$,J_k)=$LP_{m,k}$($z_{f,k}$, $I_{f,k}$)
calculate J_k
yes
J_k<ξJ_{opt}
no
discard element
yes
J_k<ξJ_{opt}
no
fractional $z_{r,k}$
no
add to near
optimal solutions
yes

Fig. 22.3 Schematic flow diagram for the mixed-integer search procedure



Selection of streams

Selected solution

Continous solution

Split-mode solution

Selected solution

Selected yield range

Flow (Split) 44.53

Name	Content
Colour	76.5
Protein	11.3
H2O	13.6
Ash	0.668
Gluten	34.8

Flow (Swivel) 44.48

Fig. 22.4 Graphic interface for integer solutions in the vicinity of the anchor continuous solution (*Top* left-hand-side of the screen shot. *Bottom* right-hand-side of the screen shot.)

22.5 Results

Tests in industrial cases confirmed that the tool fulfils the requirements of ease-of-use and flexibility. The results shown in the overview graphics were judged by the users as extremely useful. A decisive factor for the positive judgment by the users and their eagerness to use the tool was the relative quick reaction time of the computation,

- in the range of 10–20 s, whenever the user klicks the button for computing discrete solutions around the continuous-solution anchor, and
- without any perceptible delay, whenever the user changes any other input.

Compared to the manual decision making, the tool-supported decision showed a value increase of about 5 %, which is considerable in this industry.

22.6 Conclusions

A novel decision-support tool is presented that solves the problem of optimally allocating each of many intermediate product streams in flour mills to one of a few final product streams, while accounting for uncorrelated physical properties of the streams, a-priori unknown properties and yields of the final product streams, and the lack of explicit formulation of "value" of the final product streams with respect to their physical properties. A fast solution strategy is developed that decouples the original problem into simpler problems for each final product stream, uses a fictitious continuous-space solution as an initial facilitating anchor, and computes several integer solutions in the vicinity of the anchor by combining Branch-and-Bound with multiple Linear Programming steps complemented by a relaxation scheme. The developed optimization solution enables display of results without annoying time lag, whenever the user changes any input in the user-interface.

The tool can be used, and would be useful, for any mixing-decision application that fulfils *all* of the following conditions:

- A large number of products get mixed to a small number of products. The "input" products can be raw materials at the beginning of a process or intermediate products at any stage of a process. The "output" products can be intermediate products at the next (or at a later) stage of the process or final products at the end of the process.
- Several not-correlated properties are involved.
- Each property "mixes linearly."
- No mathematical relationship between the properties and the value of a product is available.
- The interplay and the compromises between properties, amounts, values, and constraints on these are not known a priori.
- Each input product (with its entire amount) can be allocated only to a single output product.

References

1. Borghetti, A., Paolone, M., Nucci, C.A.: A Mixed Integer Linear Programming Approach to the Optimal Configuration of Electrical Distribution Networks with Embedded Generators, 17th Power Systems Computation Conference. Stockholm, Sweden (2011)
2. Chen, D.-S., Batson, R.G., Dang, Y.: Applied Integer Programming: Modeling and Simulation. Wiley, New York (2010)
3. Guan, X., Zhai, Q., Papalexopoulos, A.: Optimization based methods for unit commitment: lagrangian relaxation versus general mixed integer programming. IEEE Power Eng. Soc. Gen. Meet. **2**(4), 2666 (2003)
4. Hansen, P., Jaumard, B., Savard, G.: New branch-and-bound rules for linear bilevel programming. SIAM J. Sci. Stat. Comput. **13**(5), 1194–1217 (1992)
5. Keller, J.P., Agarwal, M.: Fast, Flexible, Interactive Decision-Support Tool: An Industrial Application, 8th International Symposium on Intelligent and Manufacturing Systems. Adrasan, Turkey (2012)
6. Lawler, E.L., Wood, D.E.: Branch-and-bound methods: a survey. Oper. Res. **14**(4), 699–719 (1966)
7. Lee, M.L., Kim, J.-G., Kim, Y.-D.: Linear programming and lagrangian relaxation heuristics for designing a material flow network on a block layout. Int. J. Prod. Res. **47**(18), 5185–5202 (2009)
8. Leyffer, S.: Integrating SQP and branch-and-bound for mixed integer nonlinear programming. Comput. Optim. Appl. **18**, 295–309 (2001)
9. Vielma, J.P., Ahmed, S., Nemhauser, G.L.: A lifted linear programming branch-and-bound algorithm for mixed integer conic quadratic programs. INFORMS J. Comput. **20**, 438–450 (2008)
10. Wolsey, L.A.: Heuristic analysis, linear programming and branch and bound. Math. Prog. Stud. **13**, 121–134 (1980)

Chapter 23
Evaluation of Multi-axis Machining Processes Based on Macroscopic Engagement Simulation

Meysam Minoufekr, Lothar Glasmacher and Oliver Adams

Abstract Process planning and process design to identify stable process areas is nowadays characterized by time-consuming correction loops, where the number of iterations and the effort involved are mostly from the experience and knowledge of process designer. This requires on the one hand additional planning steps as deriving process parameters and secondly an evaluation of the achieved product quality. By using the macro simulation model introduced in this paper, the computational complexity to obtain significant process knowledge is decreased and thus made accessible more easily. Detailed tool-workpiece engagement is calculated through the presented model, which co-relates to mechanical and thermal stresses on the tool. Based on the calculations the process can be designed by reducing the tool load in the course of the process. This way, the tool life of the used milling cutters can be significantly increased resulting in an increase of process robustness and efficiency, thereby reducing used resources.

Keywords Geometric modelling · NC machining simulation · Tool/workpiece engagement

23.1 Introduction

The process sequence for milling of free-form surfaces can be divided into roughing, pre-finishing and finishing. When roughing, the goal is often to achieve a high performance cutting process (HPC) by driving the process at maximal feed rates and depth of cut resulting in a high material removal rate (MRR). However, in HPC processes high feed rates lead to increasing cutting forces and tool loads which have to be controlled in order to avoid a large tool wear or even tool breakage during the process.

M. Minoufekr (✉) · L. Glasmacher
CAx-Technologies, Fraunhofer-Institute for Production Technology (IPT), Aachen, Germany
e-mail: meysam.minoufekr@ipt.fraunhofer.de

L. Glasmacher
e-mail: lothar.glasmacher@ipt.fraunhofer.de

O. Adams
RWTH Aachen University, Aachen, Germany
e-mail: o.adams@wzl.rwth-aachen.de

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9_23

The workpiece resulting from the roughing operation is usually characterized by a macroscopic surface roughness on the surface contour, [1]. In pre-finishing, a uniform surface is achieved by removing the rest material generated in the rough machining. The goal of pre-finishing is a constant material distribution on the entire workpiece, so that in the following finishing operation, requirements of accuracy and surface quality can be achieved. As the last process, finishing is critical since the final surface of the part is generated. Failures in finish milling lead to expensive rework or even to scrap generation. Both, in pre-finishing and in the subsequent finishing processes, often a high speed cutting (HSC) approach is applied by using extremely high spindle speed and feed rates. In the course of HSC processes, the engagement conditions, e.g. the contact angle and the resulting chip thickness, have to be controlled since, in combination with the cutting speeds, the mechanical and thermal loads on the tool may result in low surface quality on the final part. Consequently, in HPC as well as in HSC processes, methods are needed for a careful process design to drive the processes to their limits but also to avoid critical situations along the value chain of the processes.

Especially in the transition from roughing to pre-finishing, the analysis and evaluation of the engagement conditions plays an important role, as the contact situation between cutter and workpiece becomes unpredictable. The residual material geometry on the workpiece, which results from roughing, usually cannot be determined in advance for parts with free-form geometry. Contact situations of the tool and the workpiece leading to unknown and possibly undesirable engagement conditions seem to be unavoidable during the process. Due to the continuously changing machining allowance and contact situation, it is important for the process planner to understand the interaction between the tool and the workpiece and evaluate the same against significant process parameters. In particular, the engagement conditions during simultaneous five-axis milling have to be observed in this context.

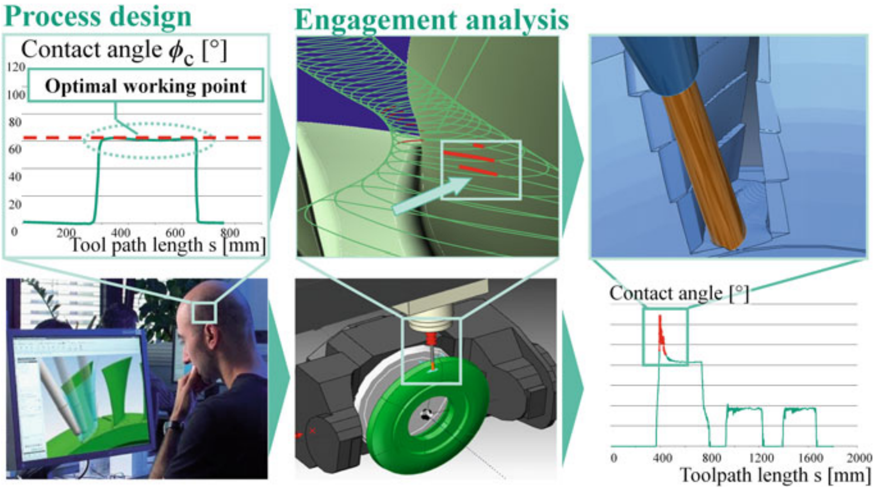


Fig. 23.1 General idea for simulation of critical sections on the toolpath (*left*). Example of a BLISK manufacturing with a critical engagement due to collision of the workpiece with the non-cutting part of the cutter (*right*)

The simulation approach introduced in this work helps to understand the geometrical engagement situation occurring in the milling process. Thus, critical regions in the process where engagement conditions exceed the tolerances can be automatically identified (Fig. 23.1, left). This is the basis for a process analysis, since process parameters directly related to the geometrical engagement conditions are evaluated and optimised in advance.

So, this work suggests the macro simulation model for multi-axis machining processes. It is structured as follows. In Sect. 23.2, problems in current multi-axis machining are described. Furthermore, the engagement and fundamental parameters defining the cutter-workpiece contact are analyzed. In Sect. 23.3, the macroscopic simulation model for machining processes is introduced. Based on this concept, the calculation models for discrete engagement conditions are explained in-depth. Then, in Sect. 23.4 the macro simulation is discussed by a roughing example, where engagement conditions are calculated. The paper is finally concluded in Sect. 23.5.

23.2 Background and Problem Definition

Almost any complex part can be produced using simultaneous multi-axis machining. Here, the simultaneous multi-axis milling is characterized by suddenly changing the tool orientation and the transient contact conditions between tool and workpiece. The manufacturing of the parts is carried out on NC controlled machine tools. Here, the process is loaded as a series of NC commands, i.e. the NC program, on the machine tool and executed. Depending on the NC command in the NC program, up to three translational and two rotational axes are controlled simultaneously during the machining. Accurate NC programs are crucial for manufacturing without interruption and potential machine breakdowns. Due to the complex kinematics, milling processes are not verifiable without supporting process design tools because it is difficult to visualise the location of the cutting tool due to the complex axis control. The consequences are unpredictable and probably critical to cutting conditions (Fig. 23.1, right).

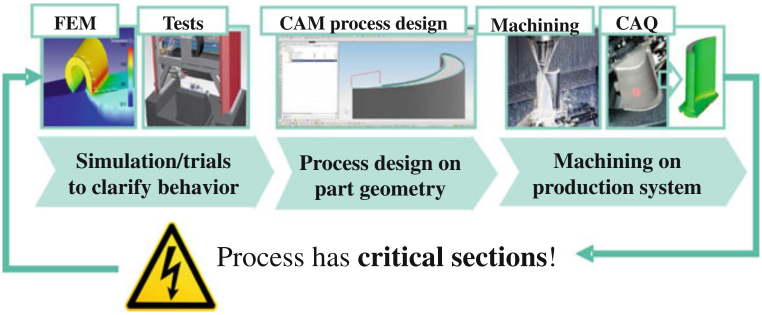

FEM
Tests
CAM process design
Machining
CAQ
Simulation/trials to clarify behavior
Process design on part geometry
Machining on production system
Process has **critical sections**!

Fig. 23.2 Sequential CAx process chain for machining

Nowadays, the development of machining processes is characterized by a sequential iterative approach, which has to be followed in a few optimization loops before reaching stable production. Therefore, computer based technologies (CAx technologies) are involved in planning and verification of the entire milling process in advance (Fig. 23.2). The parameter windows and the acquired technology knowledge gained from machining trials are used to determine the process-specific, optimal parameter combinations which are essential in the design and planning of the manufacturing.

The planning in CAM is dedicated to the design of the machining operation where the milling toolpath is parameterized by process variables such as cutting speed and depth of cut to manufacture the real part geometry. The designed process is transferred to the production system by an appropriate output data format for machining. Potential errors that are identified at the end of the process design stage, or during the process carried out on the machine tool, cause expensive iteration steps. According to [2], the costs to eliminate an error increase with the progress of the process design stages. Consequently, an identification of process parameters leading to undesired cutting conditions is considered necessary at an early stage in the process planning.

23.2.1 *Simulation-Based Design of Multi-axis Processes*

The main application of process simulation is in the area of computer-based process planning and design (Fig. 23.3). By using the process simulation, it is possible to locate errors and fix problems faster, than it is achievable through trial-and-error methods. The starting point for using the process simulation system is between the

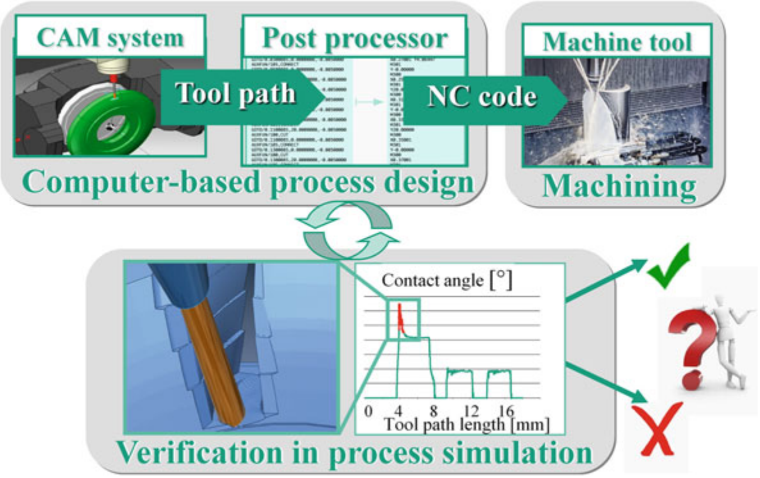


Fig. 23.3 Evaluation of the process behaviour by the presented simulation approach

NC data generation by the CAx system and the loading of NC data on the machine tool. Unlike the sequential CAx process chain, the NC data is not transferred directly to the machine tool, but passed to the simulation system. Especially requirements referring to the efficiency of the used simulation tool process play a role because the simulation increases the CAx process design chain length, [2]. In this context, the simulation model with affordable complexity of computation time and space is desirable.

23.2.2 State of the Art in the Simulation of Machining Processes

Numerous simulation approaches have been developed in the last decades with the objective of determining the important parameters of the machining process which can be classified as FE-based models, analytical models and geometrical models. The group of geometrical models can be further subdivided into approaches using constructive solid geometries (CSG) and spatial space partitioning schemes. Zabel gives a detailed overview of these models and their application in [2].

Simulation techniques based on the discretization of the workpiece vary according to the geometric design, [3–5]. Milling simulation methods based on the voxel or dexel model are presented in [6]. The consideration of physical properties is of minor importance. Systems like *Vericut* offer simulation methods for collision detection in three and five axis milling, [7]. The adjustment of the NC programs is often based on parameters, wherein the simulated residual material on the finished part is used to increase the material removal rate. However, only the feed rates are adjusted, the geometry of the toolpath remains unchanged. In these systems, assumptions in the modeling approaches are highly simplified. Hence, results obtained are insufficient for a qualitative statement about the geometrical engagement situation in the course of the milling process.

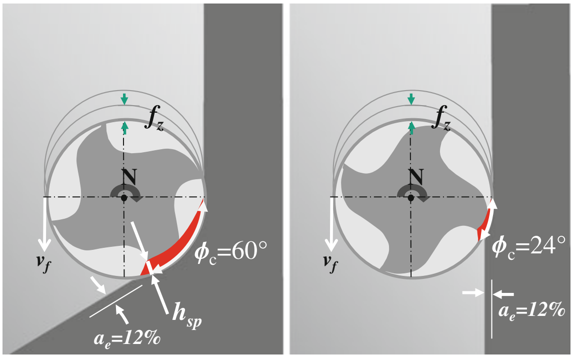


Fig. 23.4 Straight versus corner cuts result in different engagement despite equal cutting depth, (Diehl 2011)

23.2.3 Macroscopic Engagement Conditions for Process Evaluation

The mechanical and thermal tool load is primarily dependent on the combination of three factors: the chip thickness h_{sp}, the contact angle ϕ_{c} and the contact length l_{sp}, Optimal cutting conditions can be maintained easily in straight cuts by constant cutting width a_e. However, engagement situations change rapidly in a multi-axis process. In machining of free-form surfaces, the contact angle ϕ_c increases, whenever the tool enters a turn causing a longer contact length l_{sp} between the cutting edges and the workpiece (Fig. 23.4). In fact, cutting-edge temperature increases significantly as the contact length l_{sp} increases, which results in a decrease of tool life. This relation has been proven by Meinecke in [8].

The relevance of the contact angle ϕ_c can also be seen in cutting theory, as the chip thickness h_{sp} increases with increasing ϕ_c and thus leads to higher cutting forces, [10]. According to Ståhl, [11], the approximate chip thickness h_{sp} and the chip width b_{sp} are expressed by the contact angle ϕ_c and the axial depth of cut a_p, respectively:

$$h_{sp} \approx f_z \cdot \sin \phi_c \tag{23.1}$$

$$b_{sp} \approx \frac{a_p}{\sin k} \tag{23.2}$$

where κ is the major cutting edge angle of the cutting tool, f_z the feed per tooth and a_p the cutting depth. Specifically, the the machining forces in radial (F_r), tangential (F_t) and axial direction (F_a) are related to the contact angle ϕ_c. This can be seen by substituting h_{sp} and b_{sp} in the Kienzle equation, [10]:

$$\mathrm{F_i} = \frac{\mathrm{a_p}}{\sin \kappa} \cdot \mathrm{k_{i1.1}} \cdot (\mathrm{f_z} \cdot \sin \phi_c)^{(1-\mathrm{m_i})} \tag{23.3}$$

where $k_{i1.1}$ and m_i are material specific constants and $i \in \{a, r, t\}$. Hence, the contact angle and the contact length play a major role in the evaluation of machining processes. For the calculation of the contact angle ϕ_c, it is necessary to regard the cutter-workpiece engagement at each point in time in the course of the process. However, the contact angle can be determined by considering the interaction of the tool bounding geometry with the workpiece, where the angular contact area on the tool is referred as $\phi_c = \phi_{ex} - \phi_{st}$, i.e. defined as the difference of the exiting angle ϕ_{ex} and the starting angle ϕ_{st}. In accordance to [8], quantities as ϕ_{ex} and ϕ_{st} are referred to as macro conditions, since their calculation is abstracted from the exact tool geometry and the tool cutting edges are not taken into account.

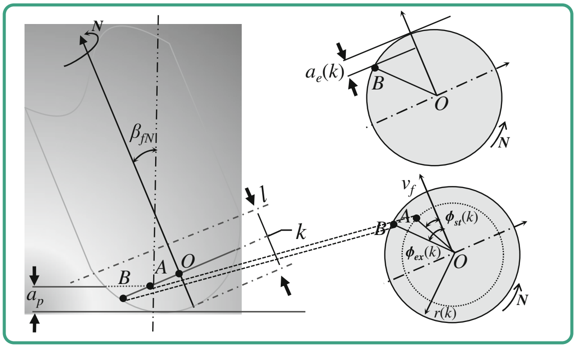


Fig. 23.5 Parameterization of the engagement conditions in reference to the tool length *l*

23.2.4 Influence of the Tool Geometry and Tool Kinematic

In addition to the suddenly changing contact situation in the course of the multi-axis milling, two further aspects complicate the calculation of macro conditions at each instant of the process; the tool geometry and tool orientation. During the machining of freeform surfaces many constellations of the engagement may arise from the material removal leading to sudden changes of engagement conditions along the tool.

To capture the engagement situation in multi-axis machining, an extended definition of process parameters and engagement conditions is needed considering the tool geometry and the tool orientation. Hence, the engagement conditions are parameterized by their axial location $k \in [0, l]$ on the tool axis at an instant in the course of the process. Thus, the starting and exiting angle are defined as $\phi_{st}(k)$ and $\phi_{ex}(k)$, respectively. Also the radial depth of cut is referred to the current location of the tool axis and is deifined as $a_e(k)$. Since the tool radius is also variable along the tool axis it is also parameterized as $r(k)$ (Fig. 23.5).

When machining with ball end mills, for instance, every cutter point undergoes a different load during the engagement. The altering contact along the tool axis has consequences for the determination of the chip length l_{sp}, since it depends on the contact angle ϕ_c and tool radius $r(k)$ which is varying according to the tool shape along the tool axis (Fig. 23.6, left).

The calculation of the engagement condition is further complicated by the orientation of the tool relative to the workpiece and the feed direction. In context of NC machining, the position of the tool is defined by a local tool coordinate system TCS which has its origin at the tool center point $\boldsymbol{O}_{TCS}$ and is determined by the current feed direction v_f, the tool orientation n and the bi-normal $b = n \times v_f$ (Fig. 23.6, right). The orientation of the tool vector n is determined by the lead and tilt angle β_{fN} and β_N, respectively. During the machining of freeform surfaces, the tool may be tilted to the surface normal, i.e. $\beta_N \neq 0$ and $\beta_{fN} \neq 0$, respectively. Since the contact situation depends on the lead angle β_N and tilt angle β_{fN}, describing the angular situation of the tool normal vector n to the surface of the workpiece, [1], the starting and exiting angles $\phi_{st}(k)$ and $\phi_{ex}(k)$ vary along the tool axis resulting from the kinematic situation of the tool and the workpiece. Hence, at each individual tool height, there is a different contact angle $\phi_c(k)$ (Fig. 23.6, right). Depending on the tool orientation to the workpiece, the contact angle $\phi_c(k)$ varies along the tool axis at a discrete instant t.

23.2.5 Problem Definition and Research Question

On one hand, the engagement situation depends on the orientation resulting in altering contact angles $\phi_c(k)$ along the tool axis. This affects also the course of h_{sp}, which is followed by Eq. (23.2). On the other hand, the tool geometry, in particular the variable tool radius along the tool axis leads to changing contact lengths during the engagement. Additionally, the contact situation changes in the course of the cutting process at each instant t. Thus, the contact angle on the tool is defined as $\phi_c(t, k)$ specifying the angular contact on point $k \in [0, l]$ along the tool axis and at an arbitrary point in time t. Furthermore, the contact length can be expressed as

$$l_{sp}(t, k) = \phi_c(t, k) \cdot r(k) \tag{23.4}$$

In this context, the following assumption can be formulated:

> A discretized simulation model allows a sufficiently accurate approximation of macroscopic engagement conditions in the course of multi-axis machining processes.

To verify this key assumption, the following research question has to be investigated:

> Can the geometrical contact situation of multi-axis machining processes be described as a sequence of discrete states, where engagement conditions vary on each discrete point k on the tool axis and at each discrete instant t?

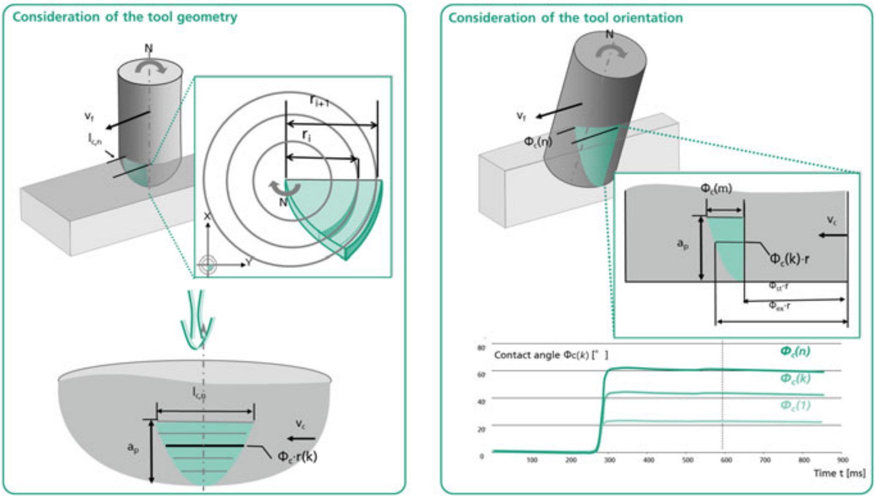


Fig. 23.6 Influence of the tool geometry on the calculation of engagement conditions (*left*). Influence of the tool orientation on the engagement conditions (*right*)

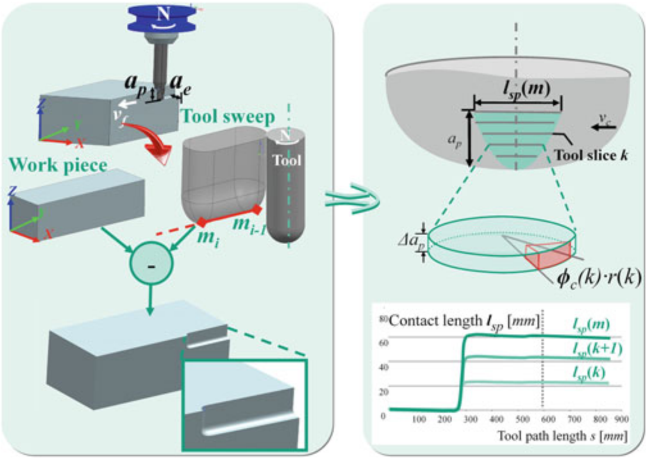
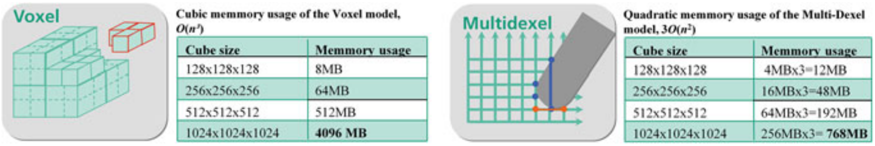


Fig. 23.7 Geometric approach for macroscopic engagement simulation of multi-axis processes

Answering this question involves the realization of a simulation model which allows a sufficiently accurate approximation of geometrical engagement conditions, in particular macroscopic engagement conditions, between the cutting tool and the workpiece in simultaneous multi-axis machining.

23.3 Solution and Method

The geometrical simulation approach introduced in this paper, determines macroscopic engagement conditions by regarding the cutter-workpiece engagement area based on a discretized geometry model. The macro simulation is based on a hierarchical simulation approach which allows the prediction of engagement conditions on the tool over a sufficiently long period of time by purely geometrical modeling (Fig. 23.7). Based on the macroscopic simulation, critical process areas can be identified or can be investigated for sub-optimal process performance. Therefore, the toolpath is divided into discrete segments. At each discrete point on the toolpath, the contact between the tool and the workpiece geometry is determined and then used for calculating the engagement conditions (Fig. 23.7). Between two points on the toolpath, the intersection of the bounding geometry of the tool and the workpiece model is determined. The set of intersecting points can be used directly for the calculation of removed material on the workpiece model. The removed material data resulting from the tool position and orientation on the current toolpath point is used to derive the macro conditions, i.e. the contact angle $\phi_c(t, k)$ and the chip length $l_{sp}(t, k)$. At a discrete instant t of the process, the profile of macroscopic quantities can be determined along the tool axis by subdividing the tool in $m + 1$ tool discs, resulting in a particular value $\phi_c(t, k)$ at each tool disc k, where $k = 0, 1, 2, \dots m$.



Cubic memmory usage of the Voxel model, $O(n^3)$

Cube size	Memmory usage
128x128x128	8MB
256x256x256	64MB
512x512x512	512MB
1024x1024x1024	**4096 MB**

Quadratic memmory usage of the Multi-Dexel model, $3O(n^2)$

Cube size	Memmory usage
128x128x128	4MBx3=12MB
256x256x256	16MBx3=48MB
512x512x512	64MBx3=192MB
1024x1024x1024	256MBx3= **768MB**

Fig. 23.8 Space complexity of the modelling schemes and their memory usage for a cube geometry (*top*)

Since the calculation of macro conditions is abstracted from the exact cutting geometry, the computational effort is drastically reduced. So, macroscopic conditions are calculable throughout the entire process by avoiding expensive calculation steps. Both aspects, namely sufficient reliability of the model as well as demands on efficiency are achieved. In this regard, efficient techniques for workpiece update with special consideration of a sufficiently precise engagement modeling are indispensable. Especially complex workpiece geometries with machining operations consisting of several 100,000 NC blocks need to be verified in the macro simulation with the reasonable response times.

23.3.1 Calculation Models for Macro Simulation

The calculation of engagement conditions is based on the geometrical contact of the cutter and the workpiece. Accurate predictions on the contact between the tool and workpiece are the basis for further calculations and therefore essential in the simulation approach. Both the tool and the workpiece can be modeled in the context of geometrical simulation in various ways. Highly dimensioned and complex workpiece geometries require a large memory usage and fast methods for the update during machining. So, the choice of appropriate "modeling schemes", i.e. how a workpiece is discretized, [12] affects the simulation accuracy and space complexity greatly (Fig. 23.8).

A further focus is on the introduction of modeling schemes with special consideration of a sufficiently accurate mapping of engagement conditions since engagement calculations are directly derived from the discretized elements of the workpiece model and hence influence the quality of the simulation results. Practical tests have shown that the dexel and the multi-dexel scheme provide a high accuracy for engagement calculation operations while allowing an efficient dynamic update of the workpiece (Fig. 23.9).

The workpiece model representation used in this work is based on the multi-dexel scheme introduced by Stautner, [6]. Furthermore, the boundary model of the milling tool, given as a CSG model, is used to determine the contact area, i.e. the cutting edges of the tool are neglected. The dexel scheme is an instance of spatial enumeration techniques with desired requirements referring accuracy and performance, [12]. The dexel model of the workpiece is determined by a set of parallel and equidistant rays intersected with the original workpiece geometry (Fig. 23.10, left). By intersecting each ray with the workpiece, two points on a line segment, which is totally inside the workpiece, are determined. A dexel is defined by these two points on the line segment. The intersection points can be calculated with a high accuracy since the dexel grid provides float precision parallel to the ray direction. However, a single dexel model leads to deviations in representing the original workpiece if the shape is sampled parallel to the projection plane of the dexel grid. Here, the determination of intersection is depending on the grid interval.

To overcome this insufficiency, the multi-dexel scheme is used which can be regarded as an extension of the dexel model. As shown in (Fig. 23.10, right), the multi-dexel model consists of three overlapping orthogonal dexel grids. Compared to the single direction dexel model, the multi-dexel scheme expresses a model more precisely since the computation of each coordinate of an intersection point can be performed with float precision. The multi-dexel model is defined as a set of $\{d_{i,j,k}\}$, by a constant grid distance δ relative to the origin point $\boldsymbol{O}$, (Ren, 2008). Each dexel $d_{i,j,k}$ has two nodes $\xi^{l}_{i,j,k}$ and $\xi^{u}_{i,j,k}$ as the lower and the upper bound of the dexel segments. The space complexity is estimated by $O(N_x \cdot N_y + N_y \cdot N_z + N_x \cdot N_z) \cdot M_d \in O(N^2)$, where N_i are the cell numbers along principal axes and M_d is the maximum number of dexel nodes. To simulate the material removal by machining, the contact area between the tool and dexels that are involved in the cutting are calculated at a given instant. This means a rearranging of the dexel data at region of tool-workpiece interaction, while the entire material removal requires a sequential update of the dexel elements. These steps are repeated at each discrete point of the machining toolpath until the machining process is completed.

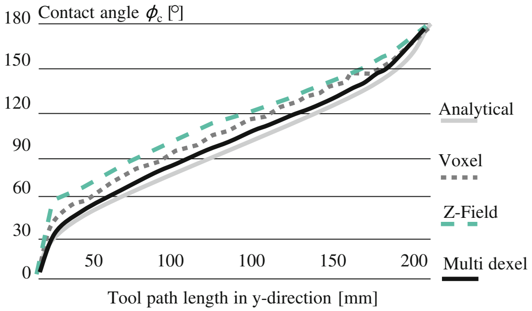


Fig. 23.9 Accuracy of the modelling schemes for engagement calculation

	Nominal	Δ Z-Fiels	Δ Voxel	Δ Multidexel
Max. Error	137.54°	10.64°	14.83°	4.23°
Ø Error	0.00°	6.77°	8.23°	2.7°

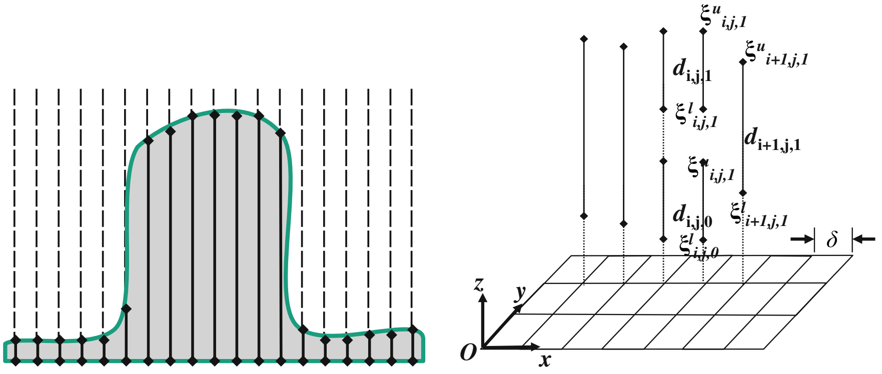


Fig. 23.10 Dexel based representation of the workpiece (*left*). Logical structure of the dexel model, Ren, 2008 (*right*)

23.3.2 Calculation of Macroscopic Engagement Conditions

The contact between the tool and the workpiece can be calculated by the intersection of the tool boundary geometry and the dexels. This requires determining the area of the tool that is in contact with each dexel and then identifying the corresponding $\xi^l_{i,j,k}$ and $\xi^u_{i,j,k}$ for each $d_{i,j,k}$ involved during a cut. Hence, let S_{Tool} and $\mathrm{S}_{Workpiece}$ be defined as the tool domain and the workpiece domain, respectively. Further, let $p = (p_x, p_y, p_z) \in S_{Tool}$ be a discrete point of S_{Tool} and $q = (q_x q_y q_z) \in S_{Workpiece}$ be a point on the workpiece. If the condition

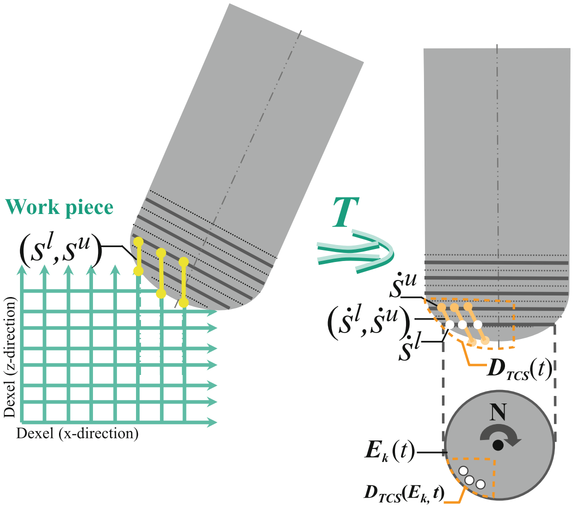


Fig. 23.11 Material removal mapped on the discrete tool disc along the tool axis

$$\exists\,\alpha,\beta,\gamma \in \{\mathrm{i},\mathrm{j},\mathrm{k}\}\ \text{with}\, \mathrm{p}_\alpha \leq\ \mathrm{q}_\alpha,\ \mathrm{p}_\beta = \mathrm{q}_\beta,\ \mathrm{p}_\gamma = \mathrm{q}_\gamma, \tag{23.5}$$

is satisfied at a discrete point in time t, the tool and the workpiece are considered to be in engagement and lead to the engagement domain $D(t)$, which is defined as follows:

$$D(t) = \{(s^l, s^u)\ |s^l \in \mathbb{R}^3, s^u \in \mathbb{R}^3\}, \tag{23.6}$$

where s^l=(s_x^l, s_y^l, s_z^l) and s^u=(s_x^u, s_y^u, s_z^u) fulfill (23.5) for a $q \in S_{Tool}$. The set $D(t)$ consists of point pairs describing the line segments which are cut by the tool at t. The corresponding points of the 2-tupels in $D(t)$ can be expressed as $s^l = (i, j,\xi_{i,j,k}^l)$ and $s^u = (i, j,\xi_{i,j,k}^u)$, respectively by regarding the intersection of the cut dexel $d_{i,j,k}$ and S_{Tool} in the nodes $\xi_{i,j,k}^l$ and $\xi_{i,j,k}^u$ (Fig. 23.11). To express the contact conditions referring tothe cutting area on the tool, each element of $D(t)$ is transformed into the coordinate system of the tool TCS, see Sect. 23.2.4. Thus, the engagement domain D_{TCS} (t) is defined as

$$\mathrm{D}_{TCS}(\mathrm{t}) = \{(\dot{\mathrm{s}}^\mathrm{l}, \dot{\mathrm{s}}^\mathrm{u})|\dot{\mathrm{s}}^\mathrm{l} = \mathrm{T}\ \cdot\ \mathrm{s}^\mathrm{l}\!-\mathrm{v}, \dot{\mathrm{s}}^\mathrm{u} = \mathrm{T}\cdot\mathrm{s}^\mathrm{u} - \mathrm{v}\}, \tag{23.7}$$

where $s^l\ \in\ D(t), s^u\ \in\ s\,D(t)$ and $T\ \in\ \mathbb{R}^{3\times 3}$ defined as the basis transformation matrix build up by the three vectors n,v_f,$b\ \in\ \mathbb{R}^3$. The vector $v\ \in\ \mathbb{R}^3$ describes the offset between $\boldsymbol{O}$ and $\boldsymbol{O}_{TCS}$. Furthermore, it can be easily seen that $D_{TCS}(t)$ is bounded by the bounding geometry of the tool, i.e.

$$\forall(\dot{s}^l, \dot{s}^u) \in D_{TCS}(t){:}\left(\dot{s}^l, \dot{s}^u\right) \subseteq S'_{Tool}, \tag{23.8}$$

where $S'_{Tool} := \{T\ \cdot\ q - v|q \in S_{Tool}\}$. The macroscopic engagement conditions are derived by using the engagement domain $D_{TCS}(t)$ at each instant t. As described in Sect. 23.2.5, the contact conditions may vary along the tool axis. In order to estimate the course of engagement conditions depending on the tool axis, the engagement area is evaluated by discretizing the tool along its length in $m + 1$ discs (Fig. 23.11).

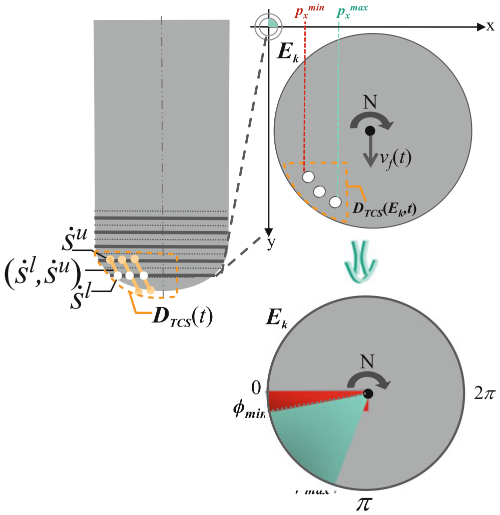
Each disc has a position and an orientation on the tool that defines the oblique cutting performed by that disc segment.For a defined tool orientation $n = (n_x, n_y, n_z)$, the tool discs can be defined as a set of planes originated in $\boldsymbol{O}_{TCS}$:

$$\mathbb{E} = \{\mathrm{E_k}\ \subset\ \mathbb{R}^3|\ \mathrm{E_k} = \{\mathrm{p} \in \mathrm{E_k}|\mathrm{n_x p_x} + \mathrm{n}_y\mathrm{p_y} + \mathrm{n_z p_z} = \mathrm{k}\cdot\Delta\mathrm{z}\}\}, \tag{23.9}$$

where k = 0, 1, 2,…, m. The contact area for each tool disc E_k is determined by finding the intersection of a line segments $(\dot{s}^l, \dot{s}^u) \in D_{TCS}(t)$ with E_k. Hence, the engagement area of E_k at a discrete instant t can be described as

$$D_{TCS}(E_k, t) = \{p|\exists(\dot{s}^l, \dot{s}^u) \in D_{TCS}(t){:}p \in E_k\ \cap\ (\dot{s}^l, \dot{s}^u)\}. \tag{23.10}$$

Fig. 23.12 Estimation of the minimal and maximal angles on a tool disc E_k



For each E_k and $(\dot{s}^l, \dot{s}^u)$, it is true that $||E_k \cap (\dot{s}^l, \dot{s}^u)|| = 1$, i.e. E_k and $(\dot{s}^l, \dot{s}^u)$ intersect only in p, except $(\dot{s}^l, \dot{s}^u)$ completely lies in E_k. By recalling (23.6), for every $p \in (\dot{s}^{l,}\dot{s}^u)$ it is also true that $p \in S'_{Tool}$ since p lies on the line segment limited by $\dot{s}^l$ and $\dot{s}^u$. It follows that $D_{TCS}(E_k, t) \subseteq S'_{Tool}$ for some $E_k \in \mathbb{E}$. Looking closer to $D_{TCS}(E_k, t)$, it can be seen that

$$\forall p \in D_{TCS}(E_k, t) : ||p - O_{TCS}|| \leq r(k) \tag{23.11}$$

for r$(k) \in \mathbb{R}$ and $k = 0,1,\ldots m$, i.e. the contact points in $D_{TCS}(E_k, t)$ are bounded by the corresponding tool radius at the tool disc E_k.

The particular angular engagement area for a given disc E_k is given by the minimal angle $\phi_{\min}(k)$ and the exit angle $\phi_{\max}(k)$ on E_k. These two parameters provide a compact form to describe the tool-workpiece engagement at a particular region along the tool axis at the tool length $l = (m + 1) \cdot \Delta z$ (Fig. 23.12). The minimal angle $\phi_{\min}(k)$ and maximal angle $\phi_{\max}(k)$ to each respective disc E_k are assigned by finding the point with minimal and maximal coordinates in x-direction on E_k. Thus, let $p^{\min}=(p_x^{\min}, p_y^{\min}, p_z^{\min})$ be the corresponding point to the minimal x-value in $D_{TCS}(E_k, t)$ and $p^{\max}=(p_x^{\max}, p_y^{\max}, p_z^{\max})$ the maximal, respectively.

The angular values of $\phi_{\max}(k)$ and $\phi_{\min}(k)$ are determined by

$$\phi_{\min}(k) := \begin{cases} \arccos\left(\frac{p_x^{\min}}{p_y^{\min}}\right), & \text{for } p_y^{\min} > 0 \\ -\arccos\left(\frac{p_x^{\min}}{p_y^{\min}}\right), & \text{for } p_y^{\min} < 0 \end{cases} \tag{23.12}$$

and

$$\phi_{\max}(k) := \begin{cases} \arccos\left(\frac{p_x^{\max}}{p_y^{\max}}\right), & \text{for } p_y^{\max} > 0 \\ -\arccos\left(\frac{p_x^{\max}}{p_y^{\max}}\right), & \text{for } p_y^{\max} < 0 \end{cases} \tag{23.13}$$

and $\phi_{\min\,/\,\max}(k) := 0$ for $p_y^{\min\,/\,\max} = 0$. In case of clockwise rotating tool, it is $\phi_{st}(k) = \phi_{\max}(k)$ and $\phi_{ex}(k) = \phi_{\min}(k)$. In case of a counterclockwise rotating tool, $\phi_{st}(k)$ and $\phi_{ex}(k)$ are swapped. The contact angle $\phi_c(k)$ at E_k is estimated by $\phi_c(k) := \phi_{ex}(k) - \phi_{st}(k)$.

23.4 Results and Discussion

The main advantage of process simulation is that process behaviour can be evaluated before cost expensive trials are conducted. The connect the simulation results with data from the process monitoring results in a higher knowledge base about the process interdependencies in the milling process (Fig. 23.13). Therefore, a synchronisation between both data sources has to be developed. As a synchronisation over time scale is not reliable, because the machine tool's acceleration behaviour has to be known exactly, here an approach using synchronisation over position is chosen.

23.4.1 Experimental Validation of Simulated Results

In order to prove the concept of linking simulation data with process monitoring data,test geometry is defined and manufactured. The test geometry consists of simple geometric features. A cuboid, a cylinder and a triangle are placed on a rectangular base (Fig. 23.14, left). As workpiece material, Aluminum is used and the tool is a standard 10 mm diameter shaft mill with two cutting edges. The workpiece is mounted on a Kistler 9255B force measurement platform to acquire the process forces. A Mazak Variaxis 630-5X II t, which is a 5-axis machining centre, is used to carry out the experiments. The toolpath and CAM operations are conducted in the PLM software Siemens NX 8.5. A standard contour parallel milling strategy is used in combination with conventional cutting (Fig. 23.14, left).

A macro simulation of the engagement condition is carried out and shows the contact angle between workpiece and tool over time (Fig. 23.14, right). The macro conditions were calculated in 3 s. 430 ms at a laptop computer (Dell Precision M4600@4x2.2 GHz and 8GB RAM). The toolpath was segmented in 2027 discrete points. The workpiece was discretised by a precision $\delta = 0.3$ mm. From the simulation it can be seen that the created toolpath leads to frequently changing engagement situations. As the contact angle correlates with the process forces the resulting tool load is varying as well. Critical sections with high contact angle are identified and can be optimized through either changing the toolpath or by reducing the feed rate in

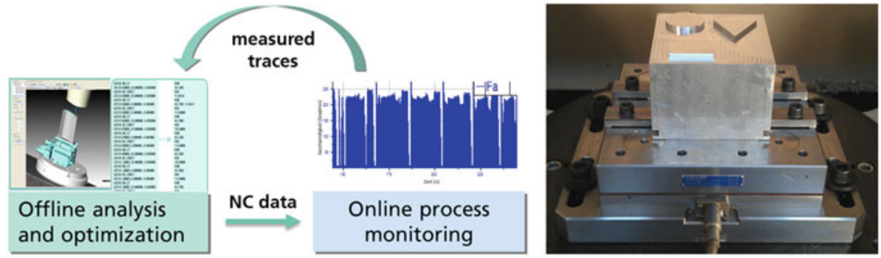


Fig. 23.13 Link between offline and online process analysis

the NC program. Furthermore, sections with rather low contact angles can be optimized in order to reduce process time. In addition to optimizing the milling process itself, process monitoring strategies can be applied. Integrating and linking geometric information from macro process simulation with process monitoring tools opens new ways for online-monitoring of multi-axis milling processes.

Using the Kienzle equation, Eq. (23.3), the expected force depending on the local engagement situation is calculated. This allows a process monitoring system to compare online, the expected force with the current force and decide whether the milling process is in a stable situation. Compared to standard monitoring systems no teaching activities are necessary. The result of calculated monitoring force and real process force is evaluated in time domain (Fig. 23.15).

23.5 Conclusions and Further Research

In order to efficiently find optimal process parameters, the level of production automation for developing processes is important. A simulation approach that allows the determination and evaluation of the actual status of the machine and the process at any time contributes to realize a systematic optimization of machining operations. However, complex calculations are used for the simulation of multi-axis milling processes which are not yet appropriately controllable despite improved hardware

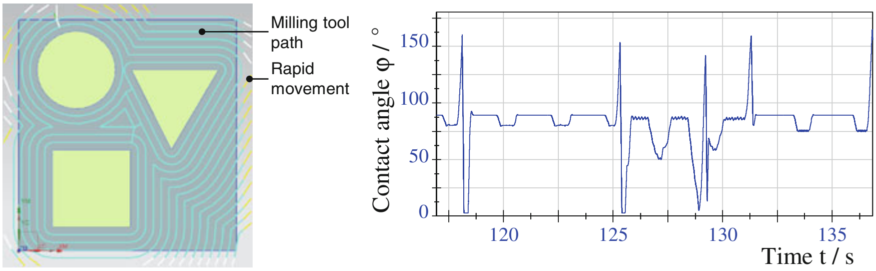


Fig. 23.14 Generated toolpath (*left*). Simulated Contact angle over time *t* (*right*)

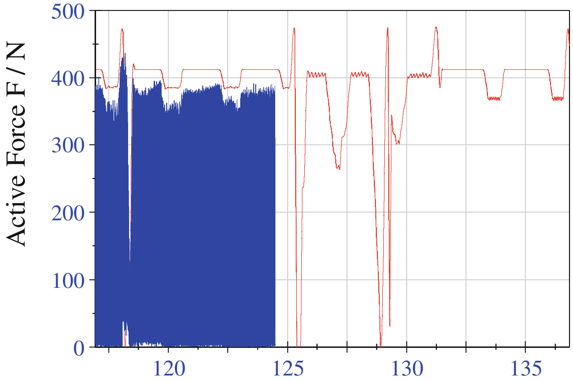


Fig. 23.15 Comparisopn of calculated forces (*red*) and measured forces (*blue*)

technologies and parallelization of algorithms. Important as the use of powerful hardware and parallel algorithms may be, novel concepts are needed to minimize the complexity of the problems to be analysed on the essential level of abstraction.

In this paper, the macro simulation for multi-axis machining processes has been introduced which focusses on significant parameters in the process. It enables the reliable detection of disturbances such as process instabilities or overloads of tool and/or machine which lead to an excess of tolerances and hence undesired process deviations. Thus, the presented work contributes to a significant reduction of expensive damages and system failures in production. Apart from the prediction of engagement conditions, further research is needed to modify process parameters based on the simulation. Complex toolpaths in particular can be optimized regarding the manufacturing and design parameters. The adjustment of feed rates, the toolpath trajectory and the tool definition offer a potential for reducing the process time and creating a more robust machining process. Hence, future research will focus on optimizing these parameters based on macroscopic engagement conditions.

Acknowledgments The authors would like to thank the German Research Foundation DFG for the support of the depicted research within the Cluster of Excellence “Integrative Production Technology for High-Wage Countries”.

References

1. Arntz, K.: Technologie des Mehrachsfräsens von vergütetem Schnellarbeitsstahl, Aachen (2013)
2. Zabel, A.: Prozesssimulation in der Zerspanung. Vulkan-Verlag, Dortmund (2010)
3. Glaeser, G.: Efficient volume-generation during the simulation of NC-milling, In: Proceedings of the International Workshop on Visualization and Mathematics’97, pp. 89–106 Springer, Heidelberg (1997)
4. Jerard, R.: Approximate methods for simulation and verification of numerically controlled machining programs. Vis. Comput. **5**, 329–348 (1989)

5. Robert, L.: Discrete simulation of NC machining. In: Proceedings of the Third Annual Symposium on Computational Geometry, Ontario (1987)
6. Stautner, M.: Simulation und Optimierung der mehrachsigen Fräsbearbeitung, Vulkan Verlag Essen (2005)
7. CGTech (2013). http://cgtech.com, (Online)
8. Meinecke, M.: Prozessauslegung zum fünfachsigenzirkularen Schruppfräsen von Titanlegierungen, Aachen (2009)
9. DiehlA.: Increasing Productivity with Engagement-Generated Toolpaths, CNC Machining, Vol 14, Iss 47 (2011)
10. Klocke, F.: Manufacturing Processes 1. Springer, Berlin (2011)
11. Ståhl, E.: Metal Cutting. Theories and Models, Elanders, Lund (2012)
12. Stifter, S.: Simulation of NC Machining Based on the Dexel Model: A Critical Analysis. Int. Journal for Advanced Manufacturing Technology, Springer, London (1995)
13. Ren, Y.: Feature Conservation and Conversion of Tri-dexel Volumetric Models. Computer-Aided Design and Application, CAD Solutions (2008)

Author Index

J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*,
Lecture Notes in Electrical Engineering 325, DOI 10.1007/978-3-319-10891-9