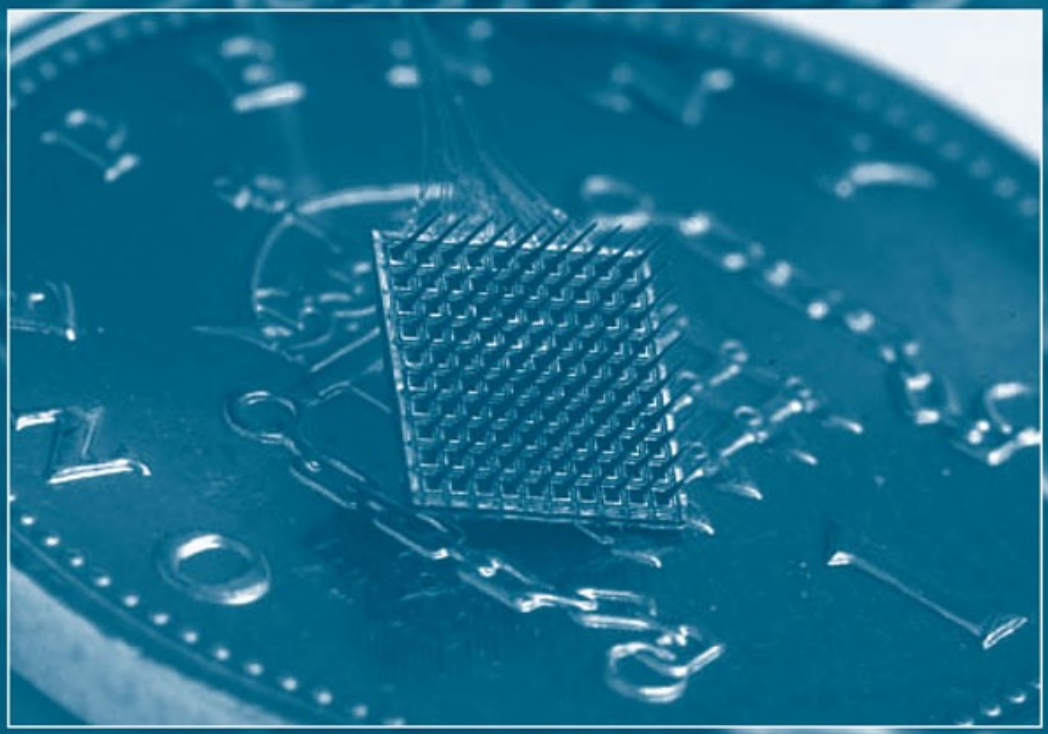
Informatics in Control, Automation and Robotics II

Editors:
Joaquim Filipe
Jean-Louis Ferrier
Juan A. Cetto
Marina Carvalho



Springer

Informatics in Control, Automation and Robotics II

Informatics in Control, Automation and Robotics II

Edited by

Joaquim Filipe
INSTICC/EST,
Setúbal, Portugal

Jean-Louis Ferrier
University of Angers,
France

Juan A. Cetto
Technical University of Catalonia,
Spain

and

Marina Carvalho
INSTICC,
Setúbal, Portugal

Springer

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN-13 978-1-4020-5625-3 (HB)
ISBN-13 978-1-4020-5626-0 (e-book)

Published by Springer,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

www.springer.com

Printed on acid-free paper

TABLE OF CONTENTS

PART 2 – ROBOTICS AND AUTOMATION

PART 3 – SIGNAL PROCESSING, SYSTEMS MODELING AND CONTROL

PREFACE

The present book includes a set of selected papers from the second "*International Conference on Informatics in Control Automation and Robotics*" (ICINCO 2005), held in Barcelona, Spain, from 14 to 17 September 2005.

The conference was organized in three simultaneous tracks: "*Intelligent Control Systems and Optimization*", "*Robotics and Automation*" and "*Systems Modeling, Signal Processing and Control*".

The book is based on the same structure.

Although ICINCO 2005 received 386 paper submissions, from more than 50 different countries in all continents, only 66 where accepted as full papers. From those, only 25 were selected for inclusion in this book, based on the classifications provided by the Program Committee. The selected papers also reflect the interdisciplinary nature of the conference. The diversity of topics is an important feature of this conference, enabling an overall perception of several important scientific and technological trends. These high quality standards will be maintained and reinforced at ICINCO 2006, to be held in Setúbal, Portugal, and in future editions of this conference.

Furthermore, ICINCO 2005 included 6 plenary keynote lectures and 1 tutorial, given by internationally recognized researchers. Their presentations represented an important contribution to increasing the overall quality of the conference, and are partially included in the first section of the book. We would like to express our appreciation to all the invited keynote speakers, namely, in alphabetical order: M. Palaniswami (University of Melbourne, Australia), Erik Sandewall (Linköping University, Sweden), Alberto Sanfeliu (Institute of Robotics and Industrial Informatics, Technical University of Catalonia, Spain), Paolo Rocchi (IBM, ITS Research and Development, Italy), Kevin Warwick (University of Reading, U.K.) and Janan Zaytoon (CReSTIC, URCA, France).

On behalf of the conference organizing committee, we would like to thank all participants. First of all to the authors, whose quality work is the essence of the conference and to the members of the program committee, who helped us with their expertise and time.

As we all know, producing a conference requires the effort of many individuals. We wish to thank all the members of our organizing committee, whose work and commitment were invaluable. Special thanks to Bruno Encarnação and Vitor Pedrosa.

Joaquim Filipe
Jean-Louis Ferrier
Juan A. Cetto
Marina Carvalho

CONFERENCE COMMITTEE

Conference Chair

Joaquim Filipe, INSTICC / EST Setúbal, Portugal

Programme co-Chairs

Jean-Louis Ferrier, University of Angers, France
Juan Andrade Cetto, Technical University of Catalonia, Spain

Organising Committee

Marina Carvalho, INSTICC, Portugal
Bruno Encarnação, INSTICC, Portugal
Vitor Pedrosa, INSTICC, Portugal

Programme Committee

Chaouki Abdallah, U.S.A.
Metin Akay, U.S.A.
Fouad M. AL-sunni, Saudi Arabia
Aníbal Traça de Almeida, Portugal
Eugenio Aguirre, Spain
Frank Allgower, Germany
Peter Arato, Hungary
Helder Araújo, Portugal
Gustavo Arroyo-Figueroa, Mexico
Artur Arsenio, U.S.A.
Marco Antonio Arteaga, Mexico
Nikolaos A. Aspragathos, Greece
Robert Babuska, The Netherlands
Mark Balas, U.S.A.
Aldo Balestrino, Italy
Bijnan Bandyopadhyay, India
Victor Barroso, Portugal
Ruth Bars, Hungary
Mike Belmont, U.K.
Alberto Bemporad, Italy
Karsten Berns, Germany
David Bonyuet, U.S.A.
Patrick Boucher, France
Guido Bugmann, U.K.
Kevin Burn, U.K.
Edmund Burke, U.K.
Clifford Burrows, U.K.
Martin Buss, Germany
Terri Caelli, Canada
Luis M. Camarinha-Matos, Portugal
Marco Campi, Italy
Xiren Cao, Hong Kong
Jorge Martins de Carvalho, Portugal
Christos Cassandras, U.S.A.
Raja Chatila, France
Tongwen Chen, Canada
Albert M. K. Cheng, U.S.A.
Graziano Chesi, Italy
Sung-Bae Cho, Korea
Ryszard S. Choras, Poland
Krzysztof Cios, U.S.A.
Carlos A. Coello Coello, Mexico
Luís Correia, Portugal
António Dourado Correia, Portugal
Yechiel Crispin, U.S.A.
Luis Custódio, Portugal
Keshav Dahal, U.K.
Guilherme Nelson DeSouza, Australia
Rüdiger Dillmann, Germany
Denis Dochain, Belgium

Alexandre Dolgui, France
Marco Dorigo, Belgium
Wlodzislaw Duch, Poland
Heinz H. Erbe, Germany
Simon Fabri, Malta
Ali Feliachi, U.S.A.
Jean-Louis Ferrier, France
Nicola Ferrier, U.S.A.
Florin Gheorghe Filip, Romania
Bruce Francis, Canada
Toshio Fukuda, Japan
Colin Fyfe, U.K.
Dragan Gamberger, Croatia
Lazea Gheorghe, Romania
Fathi Ghorbel, U.S.A.
Maria Gini, U.S.A.
Alessandro Giua, Italy
Luis Gomes, Portugal
Oscar Gonzalez, U.S.A.
John Gray, U.K.
Thomas Gustafsson, Sweden
Maki K. Habib, Malaysia
Hani Hagras, U.K.
Wolfgang Halang, Germany
John Hallam, U.K.
Riad Hammoud, U.S.A.
Uwe D. Hanebeck, Germany
John Harris, U.S.A.
Robert Harrison, U.K.
Dominik Henrich, Germany
Francisco Herrera, Spain
Weng Ho, Singapore
Gábor Horváth, Hungary
Alamgir Hossain, U.K.
Felix von Hundelshausen, U.S.A.
Amir Hussain, U.K.
Carlos Aguilar Ibañez, Mexico
Atsushi Imiya, Japan
Sirkka-Liisa Jämsä-Jounela, Finland
Ray Jarvis, Australia
Ben Jonker, The Netherlands
Visakan Kadirkamanathan, U.K.
Ivan Kalaykov, Sweden
Nicos Karcanias, U.K.
Fakhri Karray, Canada
Nik Kasabov, New Zealand
Dusko Katic, Serbia & Montenegro
Kazuhiko Kawamura, U.S.A.
Graham Kendall, U.K.
Uwe Kiencke, Germany
Rudibert King, Germany
Jozef Korbicz, Poland
Israel Koren, U.S.A.
Bart Kosko, U.S.A.
Elias Kosmatopoulos, Greece
George L. Kovacs, Hungary
Krzysztof Kozlowski, Poland
Gerhard Kraetzschmar, Germany
A. Kummert, Germany
Kostas Kyriakopoulos, Greece
Jean-Claude Latombe, U.S.A.
Loo Hay Lee, Singapore
Graham Leedham, Singapore
W. E. Leithead, U.K.
Kauko Leiviskä, Finland
Jadran Lenarcic, Slovenia
Frank Lewis, U.S.A.
Gordon Lightbody, Ireland
Jan Ligus, Slovenia
Zongli Lin, U.S.A.
Cheng-Yuan Liou, Taiwan
Brian Lovell, Australia
Joachim Lückel, Germany
Peter Luh, U.S.A.
Anthony A. Maciejewski, U.S.A.
Bruno Maione, Italy
Frederic Maire, Australia
Om Malik, Canada
Jacek Mañdziuk, Poland
Ognyan Manolov, Bulgaria
Philippe Martinet, France
Aleix M. Martinez, U.S.A.
Rene Mayorga, Canada
Gerard McKee, U.K.
Seán McLoone, Ireland
Basil Mertzios, Greece
Shin-Ichi Minato, Japan
José Mireles Jr., U.S.A.
Manfred Morari, Switzerland
Vladimir Mostyn, Czech Republic

Leo Motus, Estonia
David Murray-Smith, U.K.
Giovanni Muscato, Italy
Kenneth R. Muske, U.S.A.
Ould Khessal Nadir, Finland
Fazel Naghdy, Australia
Sergiu Nedevschi, Romania
Hendrik Nijmeijer, The Netherlands
Urbano Nunes, Portugal
José Valente de Oliveira, Portugal
Andrzej Ordys, U.K.
Djamila Ouelhadj, U.K.
Michel Parent, France
Thomas Parisini, Italy
Gabriella Pasi, Italy
Witold Pedrycz, Canada
Carlos Pereira, Brazil
Gerardo Espinosa Pérez, Mexico
Maria Petrou, U.K.
J. Norberto Pires, Portugal
Angel P. del Pobil, Spain
Marios Polycarpou, Cyprus
Marie-Noelle Pons, France
Josep M. Porta, Spain
Libor Preucil, Czech Republic
José C. Principe, U.S.A.
M. Isabel Ribeiro, Portugal
Bernardete Ribeiro, Portugal
Robert Richardson, U.K.
John Ringwood, Ireland
Juha Röning, Finland
Lluis Ros, Spain
Agostinho Rosa, Portugal
Danilo De Rossi, Italy
Hubert Roth, Germany
António E. B. Ruano, Portugal
Riko Safaric, Slovenia
Erol Sahin, Turkey
Antonio Sala, Spain
Erik Sandewall, Sweden
Ricardo Sanz, Spain
Nilanjan Sarkar, U.S.A.
Jurek Sasiadek, Canada
Daniel Sbarbaro, Chile
Carsten W. Scherer, The Netherlands
Klaus Schilling, Germany
João Sentieiro, Portugal
João Sequeira, Portugal
Wei-Min Shen, U.S.A.
Chi-Ren Shyu, U.S.A.
Bruno Siciliano, Italy
Rodolfo Soncini-Sessa, Italy
Mark Spong, U.S.A.
Aleksandar Stankovic, U.S.A.
Raúl Suárez, Spain
Ryszard Tadeusiewicz, Poland
Stanislaw Tarasiewicz, Canada
Daniel Thalmann, Switzerland
Gui Yun Tian, U.K.
Ivan Tyukin, Japan
Lena Valavani, Greece
Nicolas Kemper Valverde, Mexico
Marc van Hulle, Belgium
Cees van Leeuwen, Japan
Gerrit van Straten, The Netherlands
Annamaria R. Varkonyi-Koczy, Hungary
Jose Vidal, U.S.A.
Bernardo Wagner, Germany
Axel Walthelm, Germany
Hong Wang, U.S.A.
Jun Wang, China
Lipo Wang, Singapore
Alfredo Weitzenfeld, Mexico
Sangchul Won, Republic of Korea
Kainam Thomas Wong, Canada
Jeremy Wyatt, U.K.
Alex Yakovlev, U.K.
Hujun Yin, U.K.
Franco Zambonelli, Italy
Anibal Zanini, Argentina
Yanqing Zhang, U.S.A.
Cezary Zielinski, Poland
Albert Y. Zomaya, Australia
Detlef Zuehlke, Germany

INVITED SPEAKERS

Kevin Warwick, University of Reading, U.K.

Erik Sandewall, Linköping University, Sweden

Alberto Sanfeliu, Institute of Robotics and Industrial Informatics, Spain

Paolo Rocchi, IBM, ITS Research and Development, Italy

Janan Zaytoon, CReSTIC, URCA, France

M. Palaniswamy, University of Melbourne, Australia

Invited Speakers

COMBINING HUMAN & MACHINE BRAINS
Practical Systems in Information & Control

Kevin Warwick
Department of Cybernetics, University of Reading, Reading, RG6 6AY, United Kingdom
k.warwick@reading.ac.uk

Keywords: Artificial intelligence, Biological systems, Implant technology, Feedback control.

Abstract: In this paper a look is taken at how the use of implant technology can be used to either increase the range of the abilities of a human and/or diminish the effects of a neural illness, such as Parkinson's Disease. The key element is the need for a clear interface linking the human brain directly with a computer. The area of interest here is the use of implant technology, particularly where a connection is made between technology and the human brain and/or nervous system. Pilot tests and experimentation are invariably carried out apriori to investigate the eventual possibilities before human subjects are themselves involved. Some of the more pertinent animal studies are discussed here. The paper goes on to describe human experimentation, in particular that carried out by the author himself, which led to him receiving a neural implant which linked his nervous system bi-directionally with the internet. With this in place neural signals were transmitted to various technological devices to directly control them. In particular, feedback to the brain was obtained from the fingertips of a robot hand and ultrasonic (extra) sensory input. A view is taken as to the prospects for the future, both in the near term as a therapeutic device and in the long term as a form of enhancement.

1 INTRODUCTION

Research is presently being carried out in which biological signals of some form are measured, are acted upon by some appropriate signal processing technique and are then employed either to control a device or as an input to some feedback mechanism (e.g. Penny et al., 2000). In most cases the signals are measured externally to the body, thereby imposing errors into the situation due to problems in understanding intentions and removing noise – partly due to the compound nature of the signals being measured. Many problems also arise when attempting to translate electrical energy from the computer to the electronic signals necessary for stimulation within the human body. For example, when only external stimulation is employed then it is extremely difficult, if not impossible, to select unique sensory receptor channels, due to the general nature of the stimulation.

Wearable computer and virtual reality techniques provide one route for creating a human-machine link. In the last few years items such as shoes and glasses have been augmented with microprocessors, but perhaps of most interest is research in which a miniature computer screen was fitted onto an otherwise standard pair of glasses in order to give the wearer a remote visual experience in which additional information about an external scene could be relayed (Mann, 1997). In general though, despite being positioned adjacent to the human body, and even though indications such as stress and alertness can be witnessed, to an extent at least, wearable computers and virtual reality systems require significant signal conversion to take place in order to interface human sensory receptors with technology. Of much more interest, especially if we are considering a closely coupled combined form of operation, is clearly the case in which a direct link is formed between technology and the nervous system.

Non-human animal studies are often considered to be a pointer for what is likely to be achievable with humans in the future. As an example, in animal studies the extracted brain of a lamprey was used to control the movement of a small wheeled

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 3–9.

robot to which it was attached (Reger et al., 2000). The lamprey exhibits a response to light on the surface of water. It tries to align its body with respect to the light source. When connected into the robot body, this response was made use of by surrounding the robot with a ring of lights. As different lights were switched on and off, so the robot moved around its corral, trying to align itself appropriately.

Meanwhile in studies involving rats, a group of rats were taught to pull a lever in order to receive a suitable reward. Electrodes were then chronically implanted into the rats' brains such that when each rat thought about pulling the lever, but before any actual physical movement occurred, so the reward was proffered. Over a period of a few days, four of the six rats involved in the experiment learned that they did not in fact need to initiate any action in order to obtain a reward; merely thinking about it was sufficient (Chapin, 2004).

The most ubiquitous sensory neural prosthesis in humans is by far the cochlea implant (see Finn and LoPresti, 2003 for a good overview). Here the destruction of inner ear hair cells and the related degeneration of auditory nerve fibres results in sensorineural hearing loss. The prosthesis is designed to elicit patterns of neural activity via an array of electrodes implanted into the patient's cochlea, the result being to mimic the workings of a normal ear over a range of frequencies. It is claimed that some current devices restore up to approximately 80% of normal hearing, although for most recipients it is sufficient that they can communicate in a pretty respectable way without the need for any form of lip reading. The success of cochlea implantation is related to the ratio of stimulation channels to active sensor channels in a fully functioning ear. Recent devices consist of up to 32 channels, whilst the human ear utilises upwards of 30,000 fibres on the auditory nerve. There are now reportedly over 10,000 of these prostheses in regular operation.

In the past, studies looking into the integration of technology with the human central nervous system have varied from merely diagnostic to the amelioration of symptoms (e.g. Yu et al., 2001). In the last few years some of the most widely reported research involving human subjects is that based on the development of an artificial retina (Rizzo et al., 2001). Here small arrays have been successfully attached to a functioning optic nerve. With direct stimulation of the nerve it has been possible for the, otherwise blind, individual recipient to perceive simple shapes and letters. The difficulties with restoring sight are though several orders of magnitude greater than those of the cochlea implant simply because the retina contains millions of photodetectors that need to be artificially replicated. An alternative is to bypass the optic nerve altogether and use cortical surface or intracortical stimulation to generate phosphenes (Dobelle, 2000). Unfortunately progress in this area has been hampered by a general lack of understanding of brain functionality, hence impressive and short term useful results are still awaited.

Electronic neural stimulation has proved to be extremely successful in other areas though, including applications such as the treatment of Parkinson's disease symptoms and assistance for those who have suffered a stroke. The most relevant to this study is possibly he use of a brain implant, which enables a brainstem stroke victim to control the movement of a cursor on a computer screen (Kennedy et al., 2004). Functional magnetic resonance imaging of the subject's brain was initially carried out. The subject was asked to think about moving his hand and the output of the fMRI scanner was used to localise where activity was most pronounced. A hollow glass electrode cone containing two gold wires (Neurotrophic Electrode) was then implanted into the motor cortex, this being positioned in the area of maximum-recorded activity.

Subsequently, with the electrode in place, when the patient thought about moving his hand, the output from the electrode was amplified and transmitted by a radio link to a computer where the signals were translated into control signals to bring about movement of the cursor. Over a period of time the subject successfully learnt to move the cursor around by thinking about different movements. The Neurotrophic Electrode uses tropic factors to encourage nerve growth in the brain. During the period that the implant was in place, no rejection of the implant was observed; indeed the neurons grew into the electrode allowing stable long-term recordings.

Sensate prosthetics can also use a neural interface, whereby a measure of sensation is restored using signals from small tactile transducers distributed within an artificial limb. These can be employed to stimulate the sensory axons remaining in the user's stump which are naturally associated with a sensation. This more closely replicates stimuli in the original sensory modality, rather than forming a type of feedback using neural pathways not normally associated with the information being fed back. As a result the user

can employ lower level reflexes that exist within the Central nervous system, making control of the prosthesis more subconscious.

Functional Electrical Stimulation (FES) can also be directed towards motor units to bring about muscular excitation, thereby enabling the controlled movement of limbs. FES has been shown to be successful for artificial hand grasping and release and for standing and walking in quadriplegic and paraplegic individuals as well as restoring some basic body functions such as bladder and bowel control. It must be pointed out though that controlling and coordinating concerted muscle movements for complex and generic tasks such as picking up an arbitrary object is proving to be a difficult, if not insurmountable, challenge with this method.

In the cases described in which human subjects are involved, the aim on each occasion is to either bring about some restorative functions when an individual has a physical problem of some kind, e.g. they are blind, or conversely it is to give a new ability to an individual who has very limited abilities of any kind due to a major malfunction in their brain or nervous system. In this paper, whilst monitoring and taking on board the outputs from such research I am however as much concerned with the possibility of giving extra capabilities to a human, to enable them to achieve a broader range of skills. Essentially the goal here is to augment a human with the assistance of technology. In particular I wish to focus the study on the use of implanted technology to achieve a mental upgrade. This of course raises a number of ethical and societal questions, but at the same time it does open up a wide range of commercial opportunities.

2 AUGMENTATION

The interface through which a user interacts with technology provides a distinct layer of separation between what the user wants the machine to do, and what it actually does. This separation imposes a considerable cognitive load upon the user that is directly proportional to the level of difficulty experienced by the user. The main issue it appears is interfacing human biology with technology. In order to fully exploit all human sensory modalities through natural sensory receptors, a machine would have to exhibit a plethora of relatively complex interfacing methods. One solution is to avoid the sensorimotor bottleneck altogether by interfacing directly with the human nervous system. In doing so it is probably worthwhile first of all considering what might be gained from such an undertaking, in terms of what possibilities exist for human upgrading.

In the section which follows the research we have carried out thus far in upgrading a normal human subject will be described. The overall goals of the project are driven by the desire to achieve improved intellectual abilities for humans, in particular considering some of the distinct advantages that machine intelligence exhibits, and attempting to enable humans to experience some of these advantages at least.

Advantages of machine intelligence are for example rapid and highly accurate mathematical abilities in terms of number crunching, a high speed, almost infinite, internet knowledge base, and accurate long term memory. Presently the human brain exhibits extremely limited sensing abilities. Humans have 5 senses that we know of, whereas machines offer a view of the world which includes such as infra-red, ultraviolet and ultrasonic signals. Humans are also limited in that they can only visualise and understand the world around them in terms of a 3 dimensional perception, whereas computers are quite capable of dealing with hundreds of dimensions.

The human means of communication, getting an electro-chemical signal from one brain to another, is extremely poor, particularly in terms of speed, power and precision, involving conversion both to and from mechanical signals, e.g. pressure waves in speech communication. When one brain communicates with another there is invariably a high error rate due to the serial form of communication combined with the limited agreement on the meaning of ideas that is the basis of human language. In comparison machines can communicate in parallel, around the world with little/no error. Overall therefore, connecting a human brain, by means of an implant, with a computer network, in the long term opens up the distinct advantages of machine intelligence to the implanted individual.

Viewed dverall, connecting a human brain, by means of an implant, with a computer network, in the long term opens up the distinct advantages of machine intelligence to the implanted individual. Clearly even the acquisition of only one or two of these abilities could be enough to entice many humans to be upgraded in this way, and certainly is an extremely worthwhile driving force for the research.

3 EXPERIMENTATION

There are two main approaches in the construction of peripheral nerve interfaces, namely extraneural and intraneural. Extraneural, or cuff electrodes, wrap tightly around the nerve fibres, and allow a recording of the sum of the signals occurring within the fibres, (referred to as the Compound Action Potential), in a large region of the nerve trunk, or by a form of crudely selective neural stimulation.

A more useful nerve interface is one in which highly selective recording and stimulation of distinct neural signals is enabled, and this characteristic is more suited to intraneural electrodes. Certain types of MicroElectrode Arrays (MEAs) (shown in Figure 1) contain multiple electrodes which become distributed within the fascicle of the mixed peripheral nerve when inserted into the nerve fibres en block. This provides direct access to nerve fibres and allows for a bidirectional multichannel nerve interface. The implant experiment described here employed just such a MEA, implanted, during a 2 hour neurosurgical operation, in the median nerve fibres of my left arm, acting as a volunteer. There was no medical need for this other than in terms of the investigative experimentation that it was wished to carry out.

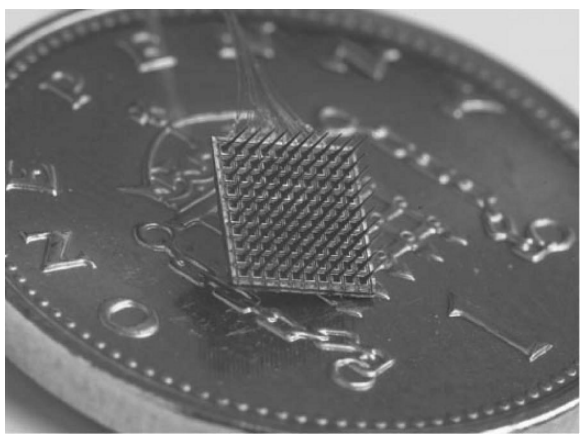


Figure 1: A 100 electrode, 4X4mm MicroElectrode Array, shown on a UK 1 pence piece for scale.

Applications for implanted neural prostheses are increasing, especially now that technology has reached a stage that reliable and efficient microscale interfaces can be brought about. In our experiment we were working hand in hand with the Radcliffe Infirmary, Oxford and the National Spinal Injuries Centre at Stoke Manderville Hospital, Aylesbury, UK – part of the aim of the experiments being to assess the usefulness of such an implant, in aiding someone with a spinal injury.

In passing, it is worthwhile pointing out that there are other types of MicroElectrode Arrays that can be used for interfacing between the nervous system and technology. For example etched electrode arrays, of which there is quite a variety, actually sit on the outside of the nerve fibres. These are, in essence, similar in operation to cuff electrodes which are crimped around the nerve fibres via a surrounding band. The signals obtained are similar to those obtainable via a cuff electrode, i.e. compound signals only can be retrieved, and hence for our purposes this type of array was not selected. To be clear, the type of Microelectrode array employed in the studies described here consists of an array of spiked electrodes that are inserted into the nerve fibres, rather than being sited adjacent to or in the vicinity of the fibres.

Stimulation current allowed information to be sent onto the nervous system, while control signals could be decoded from neural activity in the region of the electrodes.(Further details of the implant, techniques involved and experiments can be found in Warwick et al., 2003; and Gasson et al., 2005). With the movement of a finger, neural signals were transmitted to a computer and out to a robot hand (Figure 2). Signals from sensors on the robot hand's fingertips were transmitted back onto the nervous system. Whilst wearing a blindfold, in tests the author was not only able to move the robot hand, with my own neural signals, but also I could discern to a high accuracy, how much force the robot hand was applying.

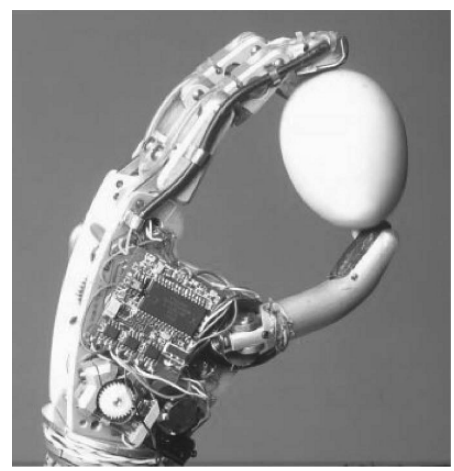


Figure 2: Intelligent anthropomorphic hand prosthesis.

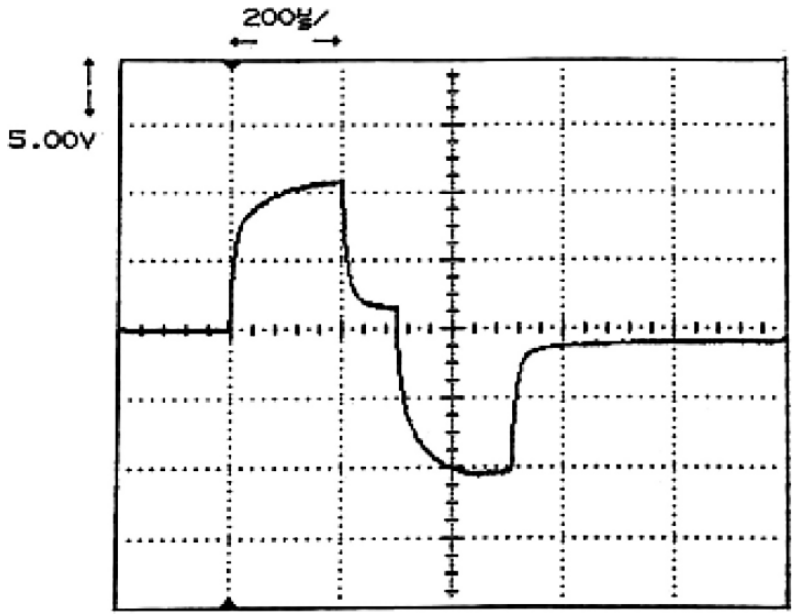


Figure 3: Voltage profile during one bi-phasic stimulation pulse cycle with a constant current of 80µA.

This experiment was also carried out via the internet with KW in Columbia University, New York City, and with the hand in Reading University, in the UK (Warwick et al., 2004). When the nervous system of a human is linked directly with the internet, this effectively becomes an extension of their nervous system. To all intents and purposes the body of that individual does not stop as is usual with the human body, but rather extends as far as the internet takes it. In our case, a human brain was able to directly control a robot hand on a different continent, obtaining feedback from the hand via the same route.

Extra sensory input (signals from ultrasonic sensors), was investigated as part of the experimentation. The author was able to obtain an accurate sense of how far objects were away, even whilst wearing a blindfold (Warwick et al., 2005). The results open up the possibility of senses of different types, for example infra-red or X-Ray also being fed onto the human nervous system and thus into the human brain. What is clear from our one off trial is that it is quite possible for the human brain to cope with new sensations of this type. To be clear on this point, it took almost 6 weeks to train the brain to recognise signals of the type shown in Figure 3 being injected onto the nervous system. When the ultrasonic input experiment was subsequently attempted, this was successful after only a few minutes of testing.

The final part of our experimentation occurred when the author's wife also had electrodes positioned directly into her nervous system. Neural signals were transmitted between the two nervous systems to realise a form of radiotelegraphic communication. The next step in this research is undoubtedly to bring about the same sort of communication between two individuals whose brains are both networked in the same way (Warwick et al., 2004).

4 CONCLUSIONS

The interaction of electronic signals with the human brain can cause the brain to operate in a distinctly different way. Such is the situation with the stimulator implants that are successfully used to counteract, purely electronically, the tremor effects associated with Parkinson's disease. Such technology can also be employed to enhance the normal functioning of the human brain. When we compare the capabilities of machines with those of humans there are obvious differences, this is true both in physical and mental terms. As far as intelligence is concerned, it is apparent that machine

intelligence has a variety of advantages over human intelligence. These advantages then become ways in which a human can be intellectually augmented, providing motivation and reasoning for making the link in the first place. The experiments described here present a glimpse into what might be possible in the future.

By linking the mental functioning of a human and a machine network, a hybrid identity is created. When the human nervous system is connected directly with technology, this not only affects the nature of an individual's (if they can still be so called) identity, raising questions as to a new meaning for 'I', but also it raises serious questions as to that individual's autonomy. Who are you if your brain/nervous system is part human part machine?

ACKNOWLEDGEMENTS

The Author would like to acknowledge the help of Mr. Peter Teddy and Mr. Amjad Shad who performed the neurosurgery described in the applications section, at the Radcliffe Infirmary, Oxford and ensured the medical success achieved thus far. My gratitude is also extended to NSIC, Stoke Manderville, to the David Tolkien Trust for their support. Ethical approval for our research to proceed was obtained from the Ethics and Research Committee at the University of Reading and, in particular with regard to the involved neurosurgery, was given by the Oxfordshire National Health Trust Board overseeing the Radcliffe Infirmary, Oxford, UK.

In particular I would like to thank those members of my team who made the project described actually occur, namely Mark Gasson, Iain Goodhew and Ben Hutt.

This work has been funded in-part by the Institut International de Rechercheen Paraplégie (IRP), Geneva and from financial assistance involving Computer Associates, Tumbleweed Communications and Nortel Networks.

REFERENCES

Chapin, J.K., Using multi-neuron population Recordings for Neural Prosthetics. *Nature Neuroscience*, Vol. 7, pp. 452–454, 2004.

Dobelle, W., Artificial vision for the blind by connecting a television camera to the visual cortex, ASAIO J, Vol. 46, pp. 3–9, 2000.

Finn, W. and LoPresti, P. (eds.), Handbook of Neuroprosthetic methods, CRC Press, 2003.

Gasson, M., Hutt, B., Goodhew, I., Kyberd, P. and Warwick, K., Invasive neural prosthesis for neural signal detection and nerve stimulation, Proc. International Journal of Adaptive Control and Signal Processing, Vol. 19, No.5, pp. 365–375, 2005.

Kennedy, P., Andreasen, D., Ehirim, P., King, B., Kirby, T., Mao, H. and Moore, M., Using human extra-cortical local field potentials to control a switch, *Journal of Neural Engineering*, Vol. 1, Issue.2, pp. 72–77, 2004.

Mann, S., Wearable Computing: A first step towards personal imaging, *Computer*, Vol. 30, Issue.2, pp. 25–32, 1997.

Penny, W., Roberts, S., Curran, E., and Stokes, M., EEG-based communication: A pattern recognition approach, *IEEE Transactions on Rehabilitation Engineering*., Vol. 8, Issue.2, pp. 214–215, 2000.

Reger, B., Fleming, K., Sanguineti, V., Simon Alford, S., Mussa-Ivaldi, F., Connecting Brains to Robots: an artificial body for studying computational properties of neural tissues, *Artificial life*, Vol. 6, Issue.4, pp. 307–324, 2000.

Rizzo, J., Wyatt, J., Humayun, M., DeJuan, E., Liu, W., Chow, A., Eckmiller, R., Zrenner, E., Yagi, T. and Abrams, G., Retinal Prosthesis: An encouraging first decade with major challenges ahead, *Opthalmology,* Vol. 108, No.1, 2001.

Warwick, K., Gasson, M., Hutt, B., Goodhew, I., Kyberd, P., Andrews, B., Teddy, P., Shad, A., The application of implant technology for cybernetic systems. *Archives of Neurology*, Vol. 60 Issue 10, pp. 1369–1373, 2003.

Warwick, K., Gasson, M., Hutt, B., Goodhew, I., Kyberd, P., Schulzrinne, H. and Wu, X., Thought Communication and Control: A First Step Using Radiotelegraphy, *IEE Proceedings on Communications,* Vol. 151, No. 3, pp. 185–189, 2004.

Warwick, K., Gasson, M., Hutt, B. and Goodhew, I., An Attempt to Extend Human Sensory Capabilities by means of Implant Technology. Proc. IEEE Int. Conference on Systems, Man and Cybernetics, Hawaii, to appear, October 2005.

Yu, N., Chen, J., Ju, M., Closed-Loop Control of Quadriceps/ Hamstring activation for FES-Induced Standing-Up Movement of Paraplegics, *Journal of Musculoskeletal Research*, Vol. 5, No.3, pp. 173–184, 2001.

BRIEF BIOGRAPHY

Kevin Warwick is a Professor of Cybernetics at the University of Reading, UK where he carries out research in artificial intelligence, control, robotics and cyborgs. He is also Director of the University TTI Centre, which links the University with SME's and raises over £2 million each year in research income.

Kevin was born in Coventry, UK and left school to join British Telecom, at the age of 16. At 22 he took his first degree at Aston University, followed by a PhD and research post at Imperial College, London. He subsequently held positions at Oxford, Newcastle and Warwick Universities before being offered the Chair at Reading, at the age of 32.

As well as publishing over 400 research papers, Kevin has appeared, on 3 separate occasions, in the Guinness Book of Records for his robotics and Cyborg achievements. His paperback 'In the Mind of the Machine' considered the possibility of machines in the future being more intelligent than humans. His recent Cyborg experiments however led to him being featured as the cover story on the US magazine, 'Wired'. Kevin has been awarded higher doctorates both by Imperial College and the Czech Academy of Sciences, Prague. He was presented with The Future of Health Technology Award in MIT and was made an Honorary Member of the Academy of Sciences, St. Petersburg. In 2000 Kevin presented the Royal Institution Christmas Lectures, entitled "The Rise of the Robots".

REDUNDANCY: THE MEASUREMENT CROSSING CUTTING-EDGE TECHNOLOGIES

Paolo Rocchi
IBM, via Shangai 53, 00144 Roma, Italy
paolorocchi@it.ibm.com

Keywords: Redundancy, Information theory, System theory, Coding theory, Robotics, Networking.

Abstract: Information technology, robotics, automatic control and other leading sectors deal with redundant components that pursue very different scopes: they improve the reliability, they perfect the behavior of an actuator, they make a message more intelligible, they get more secure telecommunication infrastructures etc. These disparate functions entail the calculus of parameters that appear rather heterogeneous and inconsistent from the mathematical viewpoint. Software developers frequently compare and balance redundant solutions and need for converging calculation methods. This paper puts forward a definition of redundancy, which aims at unifying the various measurements in use. The present proposal enhances also the progress toward the exhaustive understanding of the redundancy due to the discussion of its logical significance.

1 WIDE ASSORTMENT OF MATHEMATICAL EXPRESSIONS

People commonly take a long-winded speech as a redundant piece and double components reinforce a building. Since the antiquity the information technology and civil engineering set up redundant solutions from two separated and distinct perspectives. Nowadays redundancy infiltrates several leading edge technologies; I quote randomly: transports, weapons, telecommunications, informatics, nuclear plants, robotics, automatic processes, survey devices, man-machine systems. Those broad usages enlarge the set of heterogeneous measurements and prevent our understanding of redundant phenomena. I briefly comment some of them.

1.1 Reliability Theory

The duplication of parts enhances the reliability of a system (Ramakumar, 1993). Let P the probability of failure for one device, the probability P_n of altering n parallel devices diminishes according this law

$$P_n = P^n \quad P \tag{1.1}$$

Authors usually introduce the *degree of redundancy* as intuitive

$$r_S = n \tag{1.2}$$

1.2 Networking

A mesh has the possibility in responding to a knockout by means of the use of alternative routes. The ability of a network to maintain or restore an acceptable level of performance during a failure relies on redundancy, which has been defined as the average number of spanning trees (ANSI, 1993)

$$r_N = \frac{n_T}{n_v} \tag{1.3}$$

Where n_v are the vertices and n_T counts the spamming trees in the graph.

1.3 Robotics

In robotics for a given manipulator's vector of values q and a given representation of motion $\dot{x}$, a corresponding manipulator Jacobian $J(q)$ exists such that

$$\dot{x} = J(q)\,\dot{q} \tag{1.4}$$

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 11–16.

When the dimension of the task space n_x, to wit the dimension of $\dot{x}$, is less than the dimension n_q of vector q, the expression (1.4) has infinite solutions and the manipulator is kinematically redundant (Lewis, et al., 1993). In this case the redundancy seems to be

$$r_B = (n_q - n_x) \tag{1.5}$$

1.4 Information Theory

Claude Shannon establishes the definitive relations for efficient transmission through the entropy (Shannon, 1948). He provides a verbal definition of redundancy that we translate in the following form

$$r_H = 1 - \frac{H}{H*} \tag{1.6}$$

Where $H*$ is the maximum entropy, and H the actual entropy.

The redundancy causes the increase of unnecessary volumes in information and communication technology (ICT), hence it has been empirically introduced the *redundancy factor* r_C (Dally, 1998) which relates the length of the code in use with respect to the length of the optimal encoding

$$r_C = \frac{L}{L*} \tag{1.7}$$

1.5 Summary

The quantities r_S , r_N , r_B , r_H and r_C do not coincide, moreover some phenomena show diverging features:

A) The redundancy causes the repetition of components in a machine, whereas a redundant codeword appears *inflated* with respect to the optimal code.
B) The measurements pertinent to the same class of objects are inconsistent such as r_H and r_C.
C) The redundancy reinforces the reliability of a classical machine instead it makes more sophisticated the behavior of a robot.
D) Engineers exploit the passive redundancy for dynamical systems, which instead seems ignored in the informational realm.
E) Natural sciences and engineering deal with similar forms of redundancy (Puchkovsky, 1999).
F) Redundant information and redundant equipment may fulfil the same duty (e.g. the transmitter resends the signal, or the coding handles double-length messages for secure transmission).
G) Redundancy regards the whole system (e.g. a machine, a language), and even the parts (e.g. the circuits, the codewords).
H) Different forms of redundancy are to be balanced in some software applications e.g. in fault-tolerant systems. Engineers confront various redundant resources in the design of robots, networks, and complex appliances.

These heterogeneities contrast with some facts that instead bring evidence of the general coverage of redundancy:

Disparate theoretical constructions cause duplicated operations and loss of times on the part of technicians and managers. The arguments from E to H encourage the unified study of redundancy and authors are searching for the essential key. Shannon has defined a convincing theoretical framework for redundancy, and some authors generalize his perspective. They basically apply r_H in different territories (Jun Yang and Gupta, 2000) or otherwise tend to extend the interpretation of the entropy H (Szpankowski and Drmotu, 2002). These studies although have not produced convincing outcomes, so I moved along a different direction.

This new approach needs to be illustrated through the following introductory section.

2 PRELIMINARIES

Redundancy deals with a large variety of resources thus the first problem is to determine the argument of this measurement. I plan to unify machines and information in point of mathematics before searching for the comprehensive definition of redundancy.

I assume that the *module* is the algebraic entity ε that performs the function μ, the *system* S is a set of pairs

$$S = \{\varepsilon, \mu\} \tag{2.1}$$

I have to show how this expression, which normally applies to dynamical organizations and operations, can calculate information.

The complete nature of information is still a vexed argument, but authors find a convergence in the first stages of the theory (Saussure, 1971). They agree with the following ideas:

- An item of information is a physical and distinguishable entity.
- An item of information symbolizes something.

I translate these ideas into the mathematical language:

***Definition* 2.1:** An item of information is the entity ε which differs from a close entity ε*

$$\varepsilon \ \gamma \ \varepsilon^* \tag{2.2}$$

E.g. The ink word "Madrid" is information as it is distinct from the white paper ε* where ε is written. When the contrast lightens, (2.2) is not true and information vanishes. Shannon calculates the entropy of *m* symbols

$$H = -k\sum_{i}^{m} P_i \log_B(P_i) \tag{2.3}$$

Thus (2.2) specifies they are to be distinct. In particular if ε and ε* are huge entities, we can do but appreciate (2.2) by qualitative methods. Instead Definition 2.1 yields quantitative measurements for tiny pieces of information. For ease, when ε and ε* are point in the metric space, from (2.2) we get

$$\Delta\varepsilon = \varepsilon - \varepsilon^* \ \gamma \ 0 \tag{2.4}$$

This means that two bits must be distant even if noise and attenuation get them close. If ε and ε* are binary codewords, we quantify the diversity by means of the distance *d* which is the number of different corresponding bits and Hamming's distance is the minimal *d* for a code. As Definition 2.1 applies to both complex and simple informational items, it scales to the problem G.

Boundless literature shares the idea that information stands for something (Nöth, 1990), namely ε 'symbolizes' the object η. Some semioticians call ε 'information carrier' that is to say the piece ε 'bears the content' η. The verbs 'symbolizes', 'bears the content', 'stands for' illustrate the work μ accomplished by ε and yields this statement:

***Definition* 2.2:** The item of information ε executes the representative function μ

$$\varepsilon \xrightarrow{\ \mu\ } \eta \tag{2.5}$$

E.g. The word "Madrid" stands for the large town η; and μ is the job carried on by the piece of ink written over the present page.

In conclusion, the structure (2.1) unifies machines and information in point of mathematics. It provides the necessary basis for the logical integration of the various interpretations of redundancy.

3 REDUNDANT IS ABUNDANT

The term '*redundancy*' comes from the Latin verb '*redundare*' that means '*to overflow*' and stands for something *abundant* and *repetitive*. In accordance to this popular idea, *the number of surplus modules that accomplish the same job quantifies the redundancy*.

3.1 One-function Systems

First I assume the system *S* is equipped with *n* modules that accomplish one operation.

***Definition* 3.1:** The redundancy *r* of *S* equals to the excess of the components capable of executing μ

$$r = n - 1 \tag{3.1}$$

This definition consists with the degree of redundancy r_S up to the unit. It fits even with (1.3) because all the edges in a graph do the same job μ and r_N expresses the relative redundancy with respect to the potential spamming trees that n_v vertices can link. Eqn. (3.1) quantifies the redundancy of informational structures thanks to the semantic function (2.5). Duplicated messages make a redundant communication when all of them convey the same content. For example, a highway ends with a semaphore with six triplets of lights. The redundancy of these signals makes more secure their detection

$$S_f = \{(\varepsilon_1, \mu), (\varepsilon_2, \mu), (\varepsilon_3, \mu), (\varepsilon_4, \mu), (\varepsilon_5, \mu), (\varepsilon_6, \mu)\}$$

$$r_f = 6 - 1 = 5 \tag{3.2}$$

If we calculate the redundancy of the hardware units eqn. (3.1) gives a result symmetrical to (3.2). This means that *r* leads to identical conclusions both from the hardware viewpoint and from the software perspective.

3.2 Multiple-functions Systems

I assume the system performs *m* functions. As redundancy means abundance, I derive the following definition from (3.1).

***Definition* 3.2:** The summation of partial redundancies yields the redundancy *r* of *S* capable of executing $\mu_1, \mu_2, ..., \mu_m$

$$r = \sum_1^m {}_i r_i = \sum_1^m {}_i (n_i - 1)_i = n - m \qquad (3.3)$$

Where n_i counts the modules carrying on the same process μ_i. The variable *r* is null if the modules are just enough. The negative redundancy tells the system is deficient.

Definition 3.2 matches with the kinematic redundancy (1.5). E.g. we obtain the redundancy of a robot subtracting the essential degrees of freedom *m* from the factual degrees of freedom *n*.

Definitions 3.1 and 3.2 specify how redundancy is abundance, namely the notion of surplus is independent from the duties fulfilled. Hence redundancy may enhance the reliability or overload transmissions, it makes a movement more sophisticate or perfect the detection etc. The coverage of *r* crosses different provinces and we are able to discuss its drawbacks and advantages.

The calculus of *r* is easy in some fields. E.g. technicians reduce the redundancy of a relational database through the normalization rules that are mechanical. In other sectors the determination of *n* and *m* opposes difficulties. For instance, various indices, such as flexibility (Lenarcic, 1999), manipulability (Doty et al., 1995), isotropy etc. specify the performances of redundant manipulators. Engineers detect the redundancy of expert systems with special techniques (Suwa et al., 1982). Sometimes the determination of *m* and *n* requires subtle analysis. Take for example the ten-bits codeword 0001110101 and the parity bit 1 for check. The system 0001110101-1 includes two modules. The latter tells the bits 1 of the previous module are odd. The former conveys the decimal value 117, moreover it tells the ones are odd. We conclude the module ε_1=0001110101 executes two semantic functions namely $S_w = \{(\varepsilon_1, \mu_1), (\varepsilon_1, \mu_2), (\varepsilon_2, \mu_2)\}$. Both ε_1 and ε_2 tell the ones of ε_1 are odd and the receiver is capable of detecting errors thanks to this positive redundancy

$$r_w = 3 - 2 = 1 \qquad (3.4)$$

When information is complex, such as texts, pictures etc., the definition of the contents $\eta_1, \eta_2, ..., \eta_m$ is requisite for the definition of the semantic functions, and complicates (Klemettinen et al., 1994). In particular if two or more contents intersect

$$\eta_j \; 3 \; \eta_k \; \gamma \Leftrightarrow \; j\gamma \; k \qquad (3.5)$$

Conventional tactics are to be adopted to count *m*.

4 REDUNDANCY IN DIGITAL TECHNIQUES

Probably the most intricate questions arise in the software field, due to the variety of redundant forms which a sole application holds.

4.1 Spare Modules

A redundant machine becomes expensive when parallel devices are contemporary running, since they absorb energy and resources. Engineers implement a passive redundant solution by means of stand-by components. They prompt *m* modules ($m<n$) while the remaining devices ($n - m$) are ready to start in case of failure. These are still available but operate on-demand (Fiorini et al., 1997).
We can study this same method in ICT thanks to the generality of Definition 3.2. For the sake of simplicity, let the system *S* consist of *n* codewords with fixed length *L* and base *B*

$$n = \left(B^L \right) > m \qquad (4.1)$$

I select *m* codewords of *S* to signify the objects $\eta_1, \eta_2, ..., \eta_m = \{\eta\}$, while the remaining codewords are unused. This means that ($n - m$) modules of *S* could represent objects but lie at disposal for future implementations, namely they are on stand-by and *constitute a case of passive redundancy in the information field*. In fact Definition 3.2 specifies how the reserve codewords make *S* redundant.

$$r = (n - m) > 0 \qquad (4.2)$$

As an example, engineers want to control the ten-bits code with a parity check bit. They make 2^{11} codewords in all but use only 1024 items to symbolize letters, characters and figures. This encoding is redundant

$$r_a = 2048 - 1024 = 1024 > 0 \qquad (4.3)$$

Note how the redundancy r_w of the single word (3.4) does not coincide with the redundancy r_a of the code (4.3). In other terms, one coding encompasses two different forms of redundancy.

4.2 Minimal Length

Let the *minimal coding* S^* have m codewords with fixed length L^*, just enough to represent the items $\{\eta\}$. I make explicit n and m in (4.2)

$$r = \left(B^L - B^{L^*} \right) > 0 \qquad (4.4)$$

The base B is larger than the unit, hence (4.4) is true iff L exceeds the minimal length

$$r > 0 \;\; \omega \;\; (L - L^*) > 0$$
$$B \mu 2 \qquad (4.5)$$

The redundancy generates words longer than necessary and (4.5) brings the proof. The present theory agrees with the idea of Shannon that a redundant message exceeds the volume just sufficient to convey its contents. Expression (4.5) suggests appreciating the redundancy caused by spared codewords by means of the difference between the lengths

$$r_L = L - L^* \qquad (4.6)$$

I make explicit the minimal length

$$L^* = \log_B m \qquad (4.7)$$

I put (4.7) into (1.7) and calculate L with the symmetric formula

$$r_C = \frac{L}{L^*} = \frac{\log_B n}{\log_B m} = \log_m n \qquad (4.8)$$

This means that r_C expresses the relative increase of length and r_L tells the absolute extension. Both of them logically consist with (3.3) as they depend on the same variables. The base B and the set $\{\eta\}$ are usually given in the professional practice, hence the calculus of L^* is critical for engineers. They need the accurate value because equation (4.7) neglects the statistical effect, namely it presumes the codewords are equiprobable

$$P_i = 1/m \qquad i = 1, 2, .. \, m \qquad (4.9)$$

Shannon who quantifies the uncertainty of a source of signals through the entropy, has found the complete calculus for the minimal encoding. The following quantity provides the average number of symbols L_H^* necessary and sufficient to encode m given codewords

$$H = -k \sum_i^m P_i \log_B (P_i) = L_H^*$$
$$k > 0 \qquad (4.10)$$

If (4.9) is true, the entropy reaches the maximum and equals to (4.7)

$$H^* = -\sum_i^m 1/m \log_B (1/m) =$$
$$= \sum_i^m 1/m \log_B (m) = \log_B m = L^* \qquad (4.11)$$

In other terms, L^* is the correct value when we neglect the probabilistic distribution of symbols, while L_H^* provides the correct length of the minimal coding through the appropriate probabilities

$$L_H^* \; [\; L^* \qquad (4.12)$$

The relative increase of length due to the omission of the probabilistic distribution coincides with the Shannon redundancy.

$$\frac{L^* - L_H^*}{L^*} = \frac{H^* - H}{H^*} = 1 - \frac{H}{H^*} = r_H \qquad (4.13)$$

In short, r_H is a distinguished measurement of redundancy that consists with the criterion (4.5). The present theory suits with the calculations of Shannon and the entropy function is the essential cornerstone for this calculus. Although H grounds over several constraints that bring evidence how the Shannon redundancy is special and covers a restrict area.

5 CONCLUSIONS

Leading researches show how redundancy has effects on multiple directions. Different equations calculate the data redundancy, the actuator redundancy, the analytical redundancy, the communication redundancy, the biological redundancy, the compression redundancy, the sensors redundancy etc. This large variety obstructs the easy management

of software projects and the compare of solutions. The present paper tries to bridge this gap by means of the notion of *abundance*, which provides the lens for interpreting redundancies in distant environments and for establishing odd scopes.

I put forward two ensembles of formal expressions.

- The former part (see Section 2) unifies machines, biosystems and information, in such a way they answer the points E, F, G and H.
- The latter (see Sections 3 and 4) derive the parameters r_S , r_N , r_B , r_H, r_L and r_C from a unique definition. This approach clarifies the questions A, B, C, D.

I highlight Sections 4.1 and 4.2 as they bring evidence that:

- Passive redundancy works also in the information territory.
- Shannon's redundancy obeys to several constraints and has a specialistic coverage.

I believe that the reunification of the redundancy calculus makes easier the jobs on the practical plane and enhance our comprehension on the intellectual plane.

These findings make a part in within a broader study (Klemettinen et al., 1994) and this is the last feature written down here.

REFERENCES

ANSI Standards Committee on Telecommunications, 1993 – Network Survivability Performance – Technical Report 24, T1.TR.24-1993.

Dally W.J., Poulton J.W., 1998 – *Digital Systems Engineering* – Cambridge Univ. Press.

Doty K.L., Melchiorri C., Schwartz E.M., Bonivento C., 1995 – Robot Manipulability – *IEEE Transactions on Robotics and Automation*, 11(3), pp. 462–468.

Fiorini G.L., Staat M., Lensa W.von, Burgazzi L., 1997 – Reliability Methods for Passive Safety Functions – *Proc. of Conf. on Passive Systems,* Pisa.

Jun Yang, Gupta R., 2000 – Load Redundancy Removal through Instruction Reuse – *Proc. Intl. Conf. on Parallel Processing*, pp.61–68.

Klemettinen M., Mannila H., Ronkainen P., Toivonen H., Verkamo A.I., 1994 – Finding Interesting Rules from Large Sets of Discovered Association Rules – *Third International Conference on Information and Knowledge Management*, ACM Press, pp. 401–407.

Lenarcic J., 1999 – On the Quantification of Robot Redundancy – *Proc. IEEE Intl. Conf. on Robotics and Automation*, 4, pp. 3159–3164.

Lewis F.L., Abdallah C.T., Dawson D.M., 1993 – *Control of Robot Manipulators* – Macmillan Publishing.

Nöth W., 1990 – *Handbook of Semiotics* – Indiana University Press, Indianapolis.

Puchkovsky S.V., 1999 – Redundancy of Alive System: Notion, Definition, Forms, Adaptivity – *J. of General Biology*, 60(6).

Ramakumar R., 1993 – *Reliability Engineering: Fundamentals and Applications* – Prentice-Hall.

Rocchi P., 2000 – *Technology + Culture = Software* – IOS Press, Amsterdam.

Saussure F. de, 1971 – *Cours de Linguistique Générale* – Payot, Paris.

Shannon C.E., 1948 – A Mathematical Theory of Communication – *Bell Syst. Tech. J.*, 27, pp. 379–423.

Suwa M., Scorr A.C., Shortliffe E.H., 1982 – An Approach to Verifying Completeness and Concistency in a Rule-based Expert System - *AI MAgazin*e, 3, pp. 16–21.

Szpankowski W., Drmota M., 2002 – Generalized Shannon Code Minimizes the Maximal Redundancy – *Proc. LATIN'02*, Cancun, Mexico, pp. 1–12.

BRIEF BIOGRAPHY

Paolo Rocchi received the degree in Physics at the University of Rome in 1969. He worked in the same University in 1970, then he entered IBM. He is still working as docent and researcher in the same company. Rocchi has been a pioneer in the applications on natual language processing and linguistic computing. In the eighties he started an ample plan of investigations upon the foundations of computer science that has produced stimulating outcomes in various directions such as the reliability theory, the coding theory, software methodologies, the probability calculus, didactics. Rocchi's scientific production has been appreciated even beyond the scientific community. He has received three prizes from IBM for his publications (1978, 1999, 1992) and has a biographical entry in Who's Who in the World (2002, 2004).

HYBRID DYNAMIC SYSTEMS
Overview and Discussion on Verification Methods

Janan Zaytoon
CReSTIC, University of Reims, Moulin de la Housse, BP 1039, 51687, Reims Cédex 2, France
Janan.zaytoon@univ-reims.fr

Keywords: Verification, reachability, abstraction, approximations, Hybrid automata, decidability.

Abstract: This paper presents an overview of hybrid systems and their verification. Verification techniques are usually based on calculation of the reachable state space of a hybrid automaton representing the system under study. Decidability issues related to the verification algorithms require the use of approximation and abstraction techniques. In particular, discrete abstraction of hybrid systems and over-approximation techniques to accelerate the convergence of the reachability-based algorithms are emphasized.

1 INTRODUCTION

1.1 Overview of Hybrid Systems

The development of systematic methods for efficient and realiable realisation of hybrid systems is a key issue in industrial information and control technology and is therefore currently of high interest in many application domains. The engineering methods for hybrid systems should deal with issues related to modelling, specification, analysis, verification, control synthesis, and implementation.

Automation systems usually consist of continuous, sampled-data and discrete-event dynamic subsystems: the plant dynamics (for the most part) are continuous and can be described by (partial) differential equations or differential-algebraic systems, control algorithms are implemented as sampled-data systems or (formerly) by electronic or pneumatic devices with continuous dynamics, and discrete measurements, alarms, failure signals, on-off valves or electronic switches are processed or triggered by logic programs (or hardware) with memories, i.e. discrete-event dynamic systems. Although the discrete-event elements of automation systems have been present for decades, and in an industrial automation system usually account for the dominating part of the application code, this fact has not been reflected in the scientific literature until about 20 years ago. The pioneering work of Ramadge and Wonham drew the scientific attention to the area of discrete controls, an area that had been – and still is in industrial practice – untouched by theoretical efforts similar to those devoted to the analysis and synthesis of continuous controls. In the last 15 years, hybrid dynamic systems in which discrete-event and continuous dynamics interact closely have been a key area of scientific investigation in automation and control.

Hybrid systems theory is not only a vibrant and growing area of scientific research but it has also generated results and techniques which can be applied successfully to real-world problems and help to develop control and automation systems in a more systematic and less failure-prone fashion (Engell et al., 2004). The area of hybrid systems research can be structured into the areas of modelling and simulation, verification, and synthesis or design.

Modelling and simulation on the one hand side is concerned with the faithful representation and user-friendly (i.e. modular and partly graphical) modelling of complex continuous-discrete systems as they arise in real applications. Significant progress has been made in the effort to develop unified modelling frameworks for integrating the theories of Discrete Event and Hybrid Systems. Specifically, extending the classical setting of state automaton used in Discrete Event Systems, hybrid automata have gained popularity in dealing with hybrid systems. However, there are still many alternative models proposed. Perhaps, the simplest class of Hybrid Systems models is that of linear

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 17–26.

switched systems. Such systems have been the focus of research work originating both from the classical control theory community which views them as standard linear systems with occasional changes in the model parameters, as well as from the Discrete Event Systems community which views them as hybrid automata.

Verification of hybrid systems means the rigorous proof that a model of the system under consideration satisfies certain properties, e.g. that it never gets into an unwanted (dangerous) state or that the system does not get stuck in some state or a set of states.

In recent years, the issue of control synthesis and optimal control design for hybrid systems has been investigated increasingly as an alterative to verification after a "manual" design. A recent trend in the analysis of hybrid systems is an effort to design continuous signal to finite symbol mappings. This leads to symbolic descriptions and methods for system control, including coding in finite-bandwidth control applications and applying formal language theory to continuous system domain.

A major driving force behind recent developments in Hybrid Systems is the complexity of these systems. Therefore, ongoing work has been geared towards understanding complexity and overcoming it through a variety of approaches. Related to this development is a trend towards using quantization in control or using receding horizon concepts dealing with control problems in hybrid systems.

Despite all these developments, practical tools for designing and analysing hybrid systems are still lacking. Several efforts along these lines are ongoing, including the development of simulation tools for hybrid systems.

Emerging technologies are inevitably the drives for many of the activities in the Hybrid Systems area. Whereas manufacturing was one of the prevalent application areas in the 1980s and 1990s, communication networks and computer systems now provide a much broader and rapidly evolving playground for hybrid system researchers. Embedded systems and sensor networks are the latest developments that will foster the long anticipated convergence of communications, computing and control.

1.2 Structure of the Paper

The aim of this lecture is to present a state-of-the-art related to the verification of hybrid systems. Analysis and verification techniques are usually based on the calculation of the reachable state space of a hybrid automaton representing the system under study. Decidability issues concerning the verification algorithms require the use of approximation and abstraction techniques to accelerate the convergence of the analysis methods.

The verification approaches, presented in Section 2.1, require the use of a hybrid model of the system to be verified. Although different modelling formalisms, such as MLD (Bemporad and Morari, 1999), have been proposed in the literature, hybrid automata, which are presented in Section 2.2, are the most commonly used formalism for the specification and verification of hybrid systems. The problem that underlies safety verification for hybrid automata is reachability: can an unsafe state be reached from an initial state by running the system? In practice, many verification problems can be posed naturally as reachability problems. The traditional approach to reachability attempts to compute the set of reachable states iteratively using symbolic model checking. This computation can be automated and is guaranteed to converge in some special cases (Alur et al., 1995) for which the reachability problem is decidable. In general, however, this approach may not be automated, or may not converge. Therefore, one of the main issues in algorithmic analysis of hybrid systems is decidability, because it guarantees that the analysis will terminate in a finite number of steps. Issues related to linear hybrid automata, reachability analysis and decidability results are presented in Section 3. Henzinger et al. (1995a) showed that checking reachability for a very simple class of two-slop hybrid automata is undecidable. Therefore, other approaches to the reachability problem have been pursued. The most common of these approaches are based on abstracting the hybrid system with a discrete-event- based model for the purpose of analysis (Section 4). To overcome the termination problem, over-approximation analysis techniques, which are discussed in Section 5, are also used to enforce convergence of the iterations by computing upper approximations of the state space.

2 VERIFICATION OF HYBRID SYSTEMS

2.1 Verification

There are two major approaches to the verification of hybrid systems. In the first, the verification is directly related to the model of the hybrid system; in the second, the hybrid model is first transformed into

a discrete-event model to be explored by the verification algorithm. Traditionally, three types of property can be analysed: (i) safety properties that express the non-authorized configurations to be avoided in all the possible evolutions of the system, (ii) liveness properties that describe the possibility or the eventuality of some required system evolutions, and (iii) timeliness properties that express the constraints on the minimum and/or maximum times separating some characteristic events or evolutions.

In the first verification approach, related to the model of the hybrid system, the safety properties are particularly emphasized. These properties are specified in terms of hybrid regions (hybrid state space) that the system should avoid or remain in. Such an expression can be formulated either directly (for example, when a continuous-state variable should never cross a given threshold value) or by an observer—i.e., an automaton (to be composed with the model for verification without modifying the system behaviour) that evolves towards a particular state according to the satisfaction, or non-satisfaction, of the property (Halbwachs, 1993). Timeliness properties can also be expressed for hybrid systems using "forbidden-state observers", as time is represented implicitly in these systems. Conversely, liveness properties cannot easily be expressed in terms of forbidden states if they express the eventuality of occurrence. Only liveness properties related to the possibility of occurrence of a required event at a given time can be expressed in terms of a forbidden state.

The other approach related to verification consists of substituting the hybrid model with a discrete-event model. In this case, the classical verification algorithms of state machines can be used, such as those developed by Clarke et al. (2000). The hybrid aspect of these approaches consists of translating the hybrid model into a discrete-event model that is suitable for verification, that is, such that the conclusions of the verifications of the discrete-event system are valid relative to the properties of the original hybrid system. An ideal situation would be one in which each and every evolution of the discrete-event model corresponded to an evolution of the hybrid model, and vice versa (bi-similarity); in this case, any property that was valid relative to the discrete-event system would also be valid with respect to the hybrid system. Unfortunately, in general this goal is elusive, and the commonly used technique consists of determining a discrete-event model that is an abstraction of the hybrid model—i.e., one in which each and every evolution of the hybrid model corresponds to an evolution of the discrete-event model but the inverse is not necessary. The verification of a property characterizing all the evolutions of the discrete-event model guarantees satisfaction of the equivalent property relative to the hybrid model. The properties that can be verified using these approaches are the same as those that can be verified directly with respect to the hybrid model.

2.2 Hybrid Automata

Hybrid automata are finite automata enriched with a finite state of real-valued variables. In each location, the variables evolve continuously according to different flow fields, as long as the location's invariant remains true; then, when a transition *guard* becomes true, the control may proceed to another location and reset some of the variables to new values according to the *Jump* function of the transition. A hybrid automaton (Alur et al., 1995) is given by $H = (Q, X, \Sigma, A, Inv, F, q_0, \mathbf{x}_0)$, where:

- Q is the set of locations including the initial location, q_0;
- $X \subset \Re^n$ is the continuous-state space and $\mathbf{x}_0$ is the initial continuous state;
- Σ is the set of events;
- A is the set of transitions given by the 5-uple (q, *guard*, σ, *Jump*, q'), where q and q' represent the upstream and the downstream locations of the transition respectively, guard is a condition given in terms of the continuous-state vector, σ is an event, and Jump is a function that resets some of the variables to new values when the transition is taken;
- *Inv* is an invariant that assigns a domain of the continuous-state space to each location;
- F assigns to each location a flow field, which is usually given in terms of a differential equation but can also take imperative or less explicit forms, such as differential inclusions.

The state of a hybrid automaton is given by the couple $(q, \mathbf{x})$, which associates a location with the value of the continuous-state vector. This state can advance:

- either by the progression of time in the current location, which results in a continuous evolution of the continuous-state vector according to the corresponding flow field, F;
- or by an instantaneous transition that proceeds to a new location and changes the value of the

continuous state according to the *Jump* function of the transition. Such a transition can be taken when the continuous state satisfies the guard and provided that the resulting state vector (through the application of the *Jump* function) satisfies the invariant of the downstream location.

Complex systems can be modelled in a modular fashion by using a number of parallel automata that can be composed to produce the global model of the system. However, modelling with hybrid automata is not always an easy task because the semantics associated to this formalism is behaviour-analysis oriented (Guéguen et al., 2001). A number of approaches have therefore been proposed to generate hybrid automata starting from other modelling formalisms that are more suitable for control engineers (Stursberg et al., 1998), (Guillemaud and Guéguen, 1999).

3 HYBRID REACHABILITY

A region for a hybrid automaton is a pair (q, P), where q is a location and P is a linear predicate on the continuous-state vector. A state (q, $\boldsymbol{x}$) is included in the region (q, P), if $\boldsymbol{x}$ satisfies P—that is, if by replacing each variable in P with its value given by $\boldsymbol{x}$, one obtains a true statement. The reachability problem for hybrid automata is: given hybrid automaton H, and a set P of regions, is there a reachable state of H that is contained in some region in P? The reachability problem can be used to verify safety properties. For example, if the predicate P represents a "good" set of states in which the system should always stay (i.e., a desirable invariant of the hybrid system), then the hybrid automata satisfies this set if and only if all the states of the hybrid automata that can be reached starting from the initial region belong to P. Equivalently, the complementary predicate $B = P^c$ is a "bad" set of states that the system should avoid. Symbolic model checking techniques (Alur et al., 1995), (Henzinger et al., 1995a) can be used to compute automatically the reachable state space of a hybrid system. These techniques are based on verification algorithms that perform reachability analysis to check whether trajectories of the hybrid system can reach certain undesirable regions of the state space. When such computational algorithms are applied to systems with infinite state spaces, they are in danger of never terminating. This makes the issue of decidability, which guarantees termination of the algorithm, very important.

3.1 Hybrid Reachability Calculus

Consider a hybrid automaton for which it is required to determine the set of reachable states, starting from a region characterized by the active location, $q \in Q$, and the state space I that includes the current continuous state. As explained above the possible runs of the hybrid automaton are given as a succession of continuous transitions and discrete transitions. The calculation of the reachable state space, is based on this succession and can be summarized as follows.

Starting from a region (q, I):

- calculate the intersection of the invariant of q, $Inv(q)$, with the expansion of I at q—i.e., the set of states that can be reached from each state in I by applying the flow fields $F(q)$; the state space resulting from this intersection is given by I';
- determine Post(q, I') by iterating the following calculations for each downstream transition of q:
  - (i) calculate I_1, by intersecting I' with the set defined by the guard of the considered transition;
  - (ii) calculate the image I_2 of I_1 by applying the Jump function of the transition;
  - (iii) calculate I_3, by intersecting I_2 with the invariant of location q', $Inv(q')$, where q' is the upstream location of the considered transition;
  - (iv) calculate I_4, by intersecting the extension of I_3 at q' with the invariant of location q', $Inv(q')$;
- the resulting set of regions (q',I_4) is then added to the reachable space, and the previous step is reiterated to calculate $Post(q', I_4)$.

These calculations are reiterated until convergence —i.e., until a step is reached where the set of reachable regions does not evolve. To illustrate this calculation, consider the example of the hybrid automaton of Figure 1, which models a water-level controller that opens and shuts the outflow of a water tank. The variable x represents a clock of the water-level controller and the variable y represents the water level in the tank. Because the clock x measures time, the first derivative of x

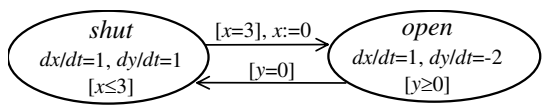


Figure 1: The water-tank automaton.

is always 1 (i.e., $dx/dt = 1$). In location *shut*, the outflow of the water tank is shut and the water level increases 1 centimetre per second ($dy/dt = 1$); in location *open*, the outflow of the water tank is open and the level decreases 2 centimetres per second ($dy/dt = -2$).

The transition from open to shut (stop the outflow) is taken as soon as the water tank becomes empty: the guard $y = 0$ on the transition ensures that the transition may be taken only when the water level is 0; the constraint $y \geq 0$ on *open* ensures that the transition to *shut* must be taken before the water level becomes negative. The transition from *shut* to *open* (open the outflow) is taken every 3 seconds: the transition is taken whenever $x = 3$, and the transition restarts the clock x at time zero.

Initially, the tank is empty and the outflow is shut. To tolerate a slight uncertainty in the water level around the zero level, the initial region is chosen to be $S_0 = (shut, I)$ with $I = -0.2 \leq y \leq 0.2 \wedge x = 0$. Now, if it is required to determine whether or not it is possible to attain the target region ($shut$, $-1 \leq x \leq 3 \wedge x = y + 1$) starting from the initial region, the reachability calculation will be as follows:

- the initial region is ($shut$, I);
- the extension of I by the flow field of location *shut* is given by $-0.2 \leq x\text{-}y \leq 0.2$, because $dx/dt = dy/dt$, and its intersection with the invariant of this location is $I' = (-0.2 \leq x - y \leq 0.2 \wedge x \leq 3)$.

The reachable state space is therefore initialized to $R_0 = (shut, -0.2 \leq x - y \leq 0.2 \wedge 0 \leq x \leq 3)$. The reachability calculation proceeds as follows:

- the transition leading to location *open* is the only one that can be taken, because its guard is $[x = 3]$, I_1 is given by $I_1 = I' \cap (x = 3)$; therefore, $I_1 = (x = 3 \wedge 2.8 \leq y \leq 3.2)$;
- by applying the jump function of the transition, $x := 0$, and considering the invariant of location *open*, the sets I_2 and I_3 are given by $I_2 = I_3 = (x = 0 \wedge 2.8 \leq y \leq 3.2)$;
- the flow field of location *open* results in: $I_4 = (2.8 \leq y + 2x \leq 3.2 \wedge x \geq 0 \wedge y \geq 0)$.

The reachable space after this first iteration is therefore given by the set: $R_1 = R_0 \cup (open, 2.8 \leq y + 2x \leq 3.2 \wedge x \geq 0 \wedge y \geq 0)$. The second iteration consists of applying the *Post* operator to the region ($open$, I_4) to calculate the sets: J_1, J_2, J_3 and J_4, for example $J_4 = (1.4 \leq x\text{-}y \leq 1.6 \wedge y \geq 0 \wedge x \leq 3)$. In this case, it is possible to verify the non-convergence of this iterative calculation of the reachable regions. However, a similar backward reachability analysis, starting from $S_{target} = (shut, 1 \leq x \leq 3 \wedge x = y+1)$, allows one to conclude, after two iterations, that the region S_{target} can be reached both from itself and from the region ($open$, $y + 2x = 2 \wedge y \geq 0$). Because S_0 has an empty intersection with these two regions, it is possible to conclude that S_{target} cannot be reached from S_0.

This trivial example shows that it is sometimes easier to verify the hybrid system using backward reachability, and that forward and backward analysis are complementary because it is not possible to know in advance which to apply. The example also shows that even for simple examples, the convergence of the reachability calculation is not guaranteed.

3.2 Decidability Considerations

Hybrid systems in which the reachability problem can be solved algorithmically in a finite number of steps are called decidable hybrid systems.

A hybrid automaton is linear if (Alur et al., 1995):

- the invariants of all the locations, the initial region and the guards are given by a conjunction of linear predicates (equalities or inequalities) over the continuous variables with rational coefficients,
- the flow fields (of all locations) are linear differential inclusions given by a conjunction of linear predicates (equalities or inequalities) over the first derivatives of the continuous variables with rational coefficients; and
- the jump functions are given by non-deterministic assignments in intervals the boundaries of which are linear expressions.

The interesting point about these automata is that all the regions of their continuous-state space (invariants, guards) are convex linear regions, and that the images of linear regions by their continuous and discrete transitions are also linear regions (Henzinger and Rusu, 1998). However the reachability problem is not decidable even for this class of hybrid automata (Alur et al., 1995) unless some restrictions are introduced. Indeed the reachabilty problem is decidable for timed automata and for automata that can be transformed into timed automata. The most important criterion for a linear hybrid automaton to be decidable then appears to be that it is initialised, i.e. that the continuous state is always the same when a location is activated. The restrictions on jump

functions of the transitions are also very important to be able to extend the class of hybrid automata that are decidable to automata with more complex dynamics (Lafferriere et al., 1999).

A practical consequence of these results is that the calculation algorithms may not converge, and it therefore becomes necessary to use some termination criteria for these algorithms. On the other hand, the user should take the risk of non-convergence into account, and should try to avoid this by adapting a verification strategy that is based on, for example, alternating forward and backward computations.

4 DISCRETE EVENT ABSTRACTION

To construct a discrete-event model of a hybrid system for the verification of a required property, it is necessary to be able to establish the correspondence between the traces of the hybrid system and those of the discrete-event model. The construction of a discrete-event model is basically concerned with partitioning the continuous part of the state space into a number of domains to each of which is associated a discrete state, then establishing a transition between two discrete states if there exists a continuous trajectory that leads from a point in the domain of the first discrete state to the domain of the second. In practice, partitioning of the entire continuous domain in terms of location invariants is complex, and it is therefore necessary to limit this approach to a certain form of model that can be restricted to some particular domains. For example, CheckMate (Chutinan and Krogh, 2003) treats the hybrid automata whose jump functions are given by the identity function and whose transition guards are given by unions of the boundaries of the locations' invariants. Consequently, these boundaries can be used to construct a discrete-event model.

Starting from a first partition (S) that corresponds, for example, to the faces of the invariants, the partition is refined by applying the following algorithm (Lafferriere et al., 1999), where the Pr function calculates the set of points representing the predecessors of a region according to the vector field:

While $\exists P, P' \in S$ *such that* $\varnothing \neq P \cap \Pr(P') \neq P$

$$P_1 = P \cap \Pr(P'); P_2 = P - \Pr(P')$$

$$S = (S - \{P\}) \cup \{P_1, P_2\}$$

As shown in Figure 2, this algorithm is based on the calculation of the set of points representing the predecessors of a region, and then the calculation of

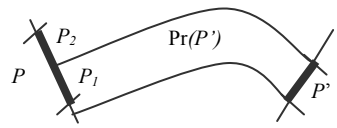


Figure 2: Partitioning.

the intersection of the predecessors with the other regions of the partition. If this intersection is empty, it is not possible to attain P' from P and, consequently, the discrete-event model does not include a transition between the corresponding states. Conversely, if the intersection is equal to P, all the points of P lead to P', and the firing of the transition between the states associated with P and P' in the discrete-event model corresponds to an evolution of the hybrid system. If neither of the two cases holds, there exists a transition in the discrete model, which relates a subset of P to P'. P is therefore partitioned into P_1 and P_2 that correspond to the two cases above: P_2 has an empty intersection with Pr(P') whereas the intersection of Pr(P') with P_1 is equal to P_1.

If this algorithm terminates, the resulting discrete-event model is bi-similar to the hybrid model—i.e., there exists a one-to-one corresponddence between the execution traces of both models, and therefore every property that is valid relative to one of these models will also be valid for the other. However, because this algorithm is based on the reachability calculation (Pr), the decidability results and the implementation difficulty are the same as for those in the case of the reachability algorithms presented in Section 3.1. Therefore, it is not possible, in general, to construct a bi-similar discrete-event model of a hybrid system, and the alternative solution consists of developing a discrete-event model that is an abstraction of the hybrid model—i.e., to each trace of the hybrid model there exists a trace in the discrete-event model but the inverse does not necessarily hold. To construct such an abstraction, it is sufficient to proceed as follows: (i) choose a partition of pertinent domains (for example, the phases of the invariants), (ii) calculate the space that is reachable by the continuous dynamics starting from each of the elements of the partition, (iii) calculate the intersection of this space with each of the other elements of the partition, and (iv) add a transition relating the corresponding discrete states if this intersection is not empty. Spurious traces that do not exist in the original hybrid model are introduced by such a construction. For example, the partition in Figure 3 implies that (P1→P3→P5) is a possible trace, which is not the case in the hybrid model.

If it is possible to obtain a partition implying that a given property is verified relative to all the traces

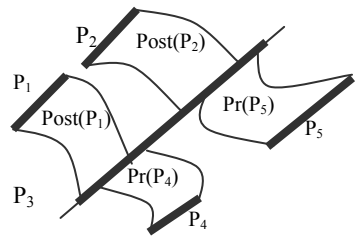


Figure 3: Spurious traces.

of the discrete-event system, then it is possible to conclude that this property is also valid with respect to the hybrid system (Chutinan and Krogh, 2001). On the other hand, owing to the abstraction and the introduction of spurious traces, the invalidity of a property relative to the set of discrete-event traces does not imply that this property is also invalid with respect to the hybrid traces. In this case, it is necessary to refine the abstractions to eliminate some of the spurious traces. Certain strategies can be used in this refinement process to accelerate the iterative abstraction process and to avoid the generation of an excessively complex discrete-event model. One strategy introduced by Stursberg et al. (2003) is to identify the traces invalidating a given property and to refine the partition along these traces.

5 CONTINUOUS EXPANSION COMPUTATION

Property verification, whether carried out through reachability analysis or by discrete-event abstractions, requires the manipulation of regions (sets) in state space (initial region, specification domain, guard conditions, location invariants, …) and the calculation of the intersections or unions of these regions and their image with the flow fields of the locations and with the jump functions. These manipulations can be performed using, for example, ellipsoidal calculations (Kurzhanski and Varaiya, 2000), but in most of the cases, the sets to be treated are polyhedra involving linear equalities or inequalities over continuous-state variables. However even for these polyhedra it may be useful to use operators such as the convex hull in order to make simpler the sets of inequalities that are manipulated and to get over-approximations of the reachable state to enhance the convergence of the fixed-point computation (Halbwachs et al., 1994).

In the approaches presented in Sections 3 and 4, the main issue is the calculation of the reachable space of the system starting from an initial region, on the basis of the continuous dynamics of the system. The region of the state space that is reachable by applying the flow field at location q, starting from region D is given by the following set.

$$R_q(D) = \left\{ x \in X \middle| \begin{array}{l} \exists t, \exists y \in D \; s.a. \; x = \Phi(y,t) \\ and \; \forall \; \tau \in [0,t], \\ \Phi(y,\tau) \in Inv(q) \\ \wedge \dot{\Phi}(y,\tau) \in F\big(q, \Phi(y,\tau)\big) \end{array} \right\}$$

The manipulation of this set—for example, calculating its intersection with a guard condition—requires the elimination of the quantifiers to obtain an expression that is not parameterized as a function of time (Laferriere et al., 1999). The set of operations required to characterize the reachable space is rather complex and difficult to implement. Furthermore, even in the case where the initial region is polyhedral, the reachable state is not necessarily polyhedral. The practical implementation of these calculations is only possible in the case where the space reachable from a polyhedron is a polyhedron—i.e. when the automaton is linear and the flow fields are given by differential inclusions. The reachable space is therefore calculated by iterative application of the convex-hull operator, starting from the vertices of the initial polyhedron and the faces of the invariants. If the flow fields of the locations are not given by differential inclusions, the reachable space is not polyhedral. Two approaches can be adopted in this case to over-approximate the reachable space by a union of polyhedra: abstraction of the continuous dynamics by a linear automaton and over-calculation of the reachable space at different times. Only the first approach is presented in the following sub-section. Information related to the second approach can be fond in (Dang, 2000; Chutinan and Krogh, 2003).

5.1 Hybrid Linear Abstractions

Starting from a given hybrid automaton, the aim of this approach is to construct a hybrid linear automaton whose flow fields are differential inclusions and whose reachable state space is an over-approximation of the state space of the original automaton (Henzinger et al., 1998).

Start by considering a hybrid automaton with a single location. The differential inclusion can be calculated by taking the invariant of the location and determining a polyhedron that contains the

derivative vector defined by the flow field at each point of the invariant. For a linear system, it is sufficient to consider the convex hull of the derivatives at the points representing the vertices of the polyhedron.

For example, consider the determination of the reachable region that is included in the invariant $I = \{0 \le x_1 \le 3 \wedge 0 \le x_2 \le 3\}$, starting from region $I = \{1 \le x_1 \le 2 \wedge x_2 = 0\}$ for the two-dimensional system whose dynamics are described using the following equation,

$$\dot{\mathbf{x}} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \mathbf{x}$$

The differential inclusion corresponding to its invariant is depicted in Figure 4a. Based on this differential inclusion, calculation of the region that can be reached from the region *I*, corresponding to the grey zone in Figure 4b, is simple. This region contains the exact reachable region of the original automaton.

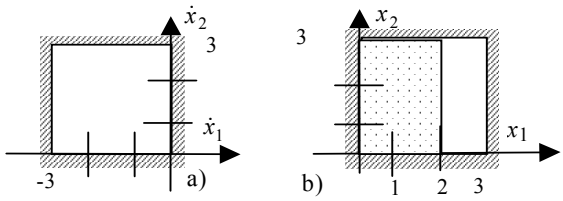


Figure 4: Abstraction by differential inclusion.

If the results of this calculation are too coarse for conclusions to be made, it is necessary to refine the abstraction. This can be done by defining a partition for the invariant, associating a location to each element of the partition by defining the associated invariants, and then calculating the differential inclusion for each location by determining a polyhedron that contains the derivative vector for each point of the new invariant. A transition is then included between two locations if there exists a continuous trajectory that crosses the boundary between the corresponding elements of the partition. It is then possible to calculate the reachability on the basis of this abstraction. However some pertinent criteria based on the properties of the continuous dynamics have to be used to refine the partition in a way that improves the efficiency of the reachability calculation (Lefebvre et al., 2002a).

For example Lefebvre et al. (2002b) proposed to used half-lines defined by the equilibrium point to determine the characteristic regions defining the partition of the state space for affine planar systems to get a trade-off between the precision of the over-approximation on the one hand, and the simplicity of the automaton and the calculations on the other. For the above example, the simple partition of Figure 5a defines the abstract automaton of Figure 5b whose reachability-calculation results are given in Figure 5c.

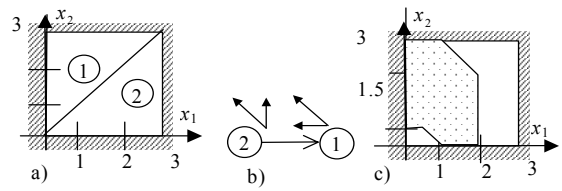


Figure 5: Improving the partition.

This refinement of the partition and, hence, the abstraction can be repeated until a sufficiently precise over-approximation of the reachable space is obtained with the view of reaching a valid conclusion. This approach enables the calculation of the continuous expansion of a given region within a location of the hybrid system. It can be used either to construct the discrete-event abstraction of the hybrid system or as a part of an algorithm for the calculation of the reachable state space of the system. However, it is generally used to construct a new automaton that models the behaviour of the system and can be used as the basis for verification. This construction requires the substitution of each location by a sub-automaton representing the abstraction of its continuous behaviour, the determination of the transitions between these new locations and the original locations. These translations may result in complex automata that are difficult to verify. Again, it is necessary to have a methodology that guides the refinement of the partition, especially for the regions that are considered to be critical for the verification of a given property to reduce the complexity of the resulting automaton. These abstractions enable the use of the existing results and tools such as Hytech or KRONOS (Yovine, 1993), and UPPAAL (Bengtsson et al., 1996) if abstraction by timed automata is possible (Henzinger et al., 1998).

6 CONCLUSIONS

This paper has presented an overview of current issues related to hybrid systems as well as the principles of analysis and verification of these systems. This verification can be performed relative to either the hybrid model by calculation of the

reachable state space and its intersection with a forbidden region or a discrete-event model representing an abstraction of the hybrid model. In both cases, the major problem is related to the determination of the continuous expansion of a region, which is generally manipulated in terms of unions of polyhedra. Because this expansion cannot be represented exactly as union of polyhedra, over-approximation techniques may be used for this purpose. These over-approximations can be based on the abstraction of continuous behaviour with a set of differential inclusions, or the calculation of the image of the initial region at certain times.

It is possible to experiment with these verification techniques thanks to the availability of software tools, such as HYTECH (Henzinger et al., 1995c) for linear hybrid automata, KRONOS (Yovine 1993) or UPPAAL (Bengtsson et al., 1996) for timed automata, and Verdict (Stursberg et al., 1998) or CheckMate (Chutinan et al., 2000) for abstraction techniques. Many authors highlight the advantages, the difficulties and the open issues related to the use of these tools (Silva et al., 2001; Stauner et al., 1997). To improve engineering procedures, some of these tools are based on environments that are commonly used by control engineers, such as StateFlow-Simulink for CheckMate. To be efficient in real-sized industrial applications, these tools should be used in a framework of global methodologies that provide high-level modelling facilities to enable the capture of the models of complex systems and determines the pertinent abstraction and approximation techniques required for the verification of a given property. The development of such an integrated methodology represents one of the main challenges to be faced by the hybrid systems community in the coming years, and requires close collaboration between control engineers and computer scientists.

REFERENCES

Alur R., Courcoubetis C., Halbwachs N., Henzinger T.A., Ho P.H., Nicollin X., Olivero A., Sifakis J., Yovine S. "The algorithmic analysis of hybrid systems", *Theoritical Computer Science*, no. 138, 1995, pp. 3–34.

Bemporad A, Morari M, "Verification of Hybrid Systems via mathematical programming", in (Vaandrager et al., 1999), *Hybrid Systems: Computation and Control*, pp. 31–45.

Bengtsson J., Larsen K.G., Larsson F., Pettersson P., Yi W., "UPPAL: a tool suite for automatic verification of real-time systems", *in Hybrid Systems III*, LNCS vol. 1066, 1996, pp. 232–243.

Chutinan A., Krogh B., Milan D., Richeson K., de Silva B.I., "Modeling and verifying hybrid dynamic systems using CheckMate", in (Engell S., Kowalewski S., Zaytoon J., Eds), *Hybrid Dynamical Systems. Proceedings of 4th Int. Conf. on Automation of Mixed Processes*, Dortmund, Germany, 2000, pp. 323–328.

Chutinan A., Krogh B., "Verification of Infinite-State Dynamics Systems using approximate quotient transition systems", *IEEE Trans. On Automatic Control* Vol. 46 no. 9, Sept. 2001, pp. 1401–1410.

Chutinan A., Krogh B., "Computational techniques for hybrid systems verification", IEEE *Trans. Automatic Control* Vol. 48 no. 1 January 2003. pp. 64–75.

Clarke E.M., Grumberg O., Long D., *Model Checking* Cambridge MA, MIT Press, 2000.

Dang T., "Vérification et synthèse des systèmes hybrides", Thèse de doctorat de l'INP Grenoble, Verimag, 2000.

Engell S., Guéguen H., Zaytoon J. (Eds.), "Analysis and Design of Hybrid Systems", Special issue of Control Engineering Practice, Vol. 12, no. 10, 2004.

Flaus J.M., "Stabilité des systèmes dynamiques hybrides", in (*Zaytoon 2001*), pp. 237–254.

Guéguen H., Valentin-Roubinet C., Pascal J.C., Soriano T., Pingaud H., "Modèles mixtes et structuration des modèles complexes", in (*Zaytoon 2001*), pp. 157–188.

Guillemaud L., Guéguen H., "Extending Grafcet for the specification of control of hybrid systems", Proceedings of IEEE SMC 99, Tokyo, Japan, 1999.

Halbwachs N., "Delay analysis in synchronous programs", *Proceedings of 5th Int. Conference on Computer-Aided Verification,* LNCS-697, Springer, 1993, pp. 333–346.

Halbwachs N., Raymond P., Proy Y.E., "Verification of linear hybrid systems by means of convex approximations", *Proceedings of Static Analysis Symposium*, LNCS-864, 1994, pp. 223–237.

Henzinger T.A., Kopke P.W., Puri A., Varaiya P., "What's decidable about hybrid automata? The algorithmic analysis of hybrid systems", *Proceedings of 27th. annual ACM Symposium on Theory of Computing,* 1995, pp. 373–382.

Henzinger T.A., Ho P.H., "A note on abstract interpretation strategies for hybrid automata", in *Hybrid Systems II*, LNCS, Vol 999, Spinger, 1995, pp. 252–263.

Henzinger T.A., Ho P.H., Wong-Toi H., "A user guide to HYTECH", *In: Tools and Algorithms for the Construction and Analysis of Systems*, LNCS-1019, 1995, pp. 41–71.

Henzinger T.A., Ho P.H., Wong-Toi H., "Algorithmic analysis of non-linear hybrid systems", in (Antsaklis P.J., Nerode A., Eds.), Special Issue on Hybrid Systems, *IEEE Transactions on Automatic Control,* Vol. 43, n° 4, April 1998, pp. 540–554.

Henzinger T.A., Rusu V., "Reachability verification for hybrid automata", *Proceedings of 1st Int. Workshop on Hybrid Systems: Computation and Control*, LNCS-1386, 1998, pp. 190–204.

Katayama T., McKelvey T., Sano A., Cassandras C., Campi M., "Trends in Systems and Signals – status report presented by the IFAC Coordinating

Committee on Systems and Signals", in Proceedings of 16th IFAC World Congress, pp. 111–121, 2005.
Kurzhanski A. B., Varaiya P., "Elipsoidal techniques for reachability analysis", in (Lynch et al., 2000), pp. 202–214.
Lafferriere G., Pappas G.J., Yovine S., "A new class of decidable hybrid systems", in (Vaandrager et al., 1999), pp. 137–152.
Lefebvre M.A., Guéguen H. Buisson J. "Hybride approximations for continuous systems", *Journal Européen des Systèmes Automatisés (JESA)*, Vol. 36, no. 7, 2002, pp. 959–971.
Lefebvre M.A., Guéguen H. Buisson J., "Structured hybrid abstractions of continuous systems", IFAC world Congress B02, Barcelonne, 2002b.
Lynch N. Krogh B.H. (Eds.) *Hybrid Systems Computation and Control, HSCC2000*, LNCS, Vol. 1790, Spinger, 2000.
Silva B.I., Stursberg O., Krogh B. H., Engell S., "An assessment of the current status of algorithmic approaches to the verification of Hybrid Systems", in *Proc. of the IEEE Conf. On Decision and Control*, Orlando, December 2001, pp. 2867–2874.
Stauner T., Müller O., Fuchs M., "Using HyTech to verify an automative control system", *Proceedings of Hybrid and Real-Time Systems Workshop*, LNCS-1201, Springer, 1997, pp. 139–153.
Stursberg O., Kowalewski S., Preuβig J., Treseler H., "Block diagram based modelling and analysis of hybrid processes under discrete control", *Journal Européen des Systèmes Automatisés,* Vol. 32, no. 9–10, 1998, pp. 1097–1118.
Stursberg O., Fehnker A., Han Z., Krogh B., "Specification-guided analysis of hybrid systems using hierarchy of validation methods", in: Engell S., Guéguen H., Zaytoon J., (Eds.), *Proceeding of IFAC ADHS'03 Analysis and Design of Hybrid Systems*, June 2003, Elsevier, 2003, pp. 289–294.
Vaandrager F.W., van Schuppen J.H. *Hybrid Systems, Computation and Control*, HSCC99, LNCS, 1569, Springer, 1999.
Yovine S., Méthode et outils pour la vérification symbolique de systèmes temporisés, Thèse de l'INP Grenoble, 1993.
Zaytoon J. (Ed.), Systèmes Dynamiques Hybrides, éditions Hermès, 2001.

BRIEF BIOGRAPHY

Janan Zaytoon received the PhD degree from the National Institute of Applied Sciences (INSA) of Lyon, France in 1993. From 1993 to 1997 he was an assistant professor, and since 1997 he has been a Professor at the University of Reims Champagne-Ardenne. He is the Director of the CReSTIC Research Centre of the University of Reims, the Deputy-Director of French GDR MACS of CNRS, the leader of the French national group on hybrid dynamical systems, the Chairman of the IFAC French National Member Organizer, and the Co-Chair of the IFAC Technical Committee on Discrete Event and Hybrid Systems.

Janan Zaytoon has published more than 150 journal papers, books, book chapters, and communications in international conferences. His main research interests are in the fields of Discrete Event Systems and Hybrid Dynamical Systems. He is the Chair (or Co-Chair) of 7 international conferences, 6 of which are IFAC events, 2 national conferences and 1 International School. He is also an Associate Editor of Control Engineering Practice, the Keynote speaker for 3 conferences, and the Guest Editor for 8 special issues on Discrete Event Systems and/or Hybrid Dynamical Systems in the following journals : Control Engineering Practice (2), Discrete Event Dynamic Systems, European Journal of Automation, JESA (3), e-STA, and "Revue d'Elecrtronique et d'Electrotechnique".

TARGET LOCALIZATION USING MACHINE LEARNING* †

M. Palaniswami, Bharat Sundaram, Jayavardhana Rama G. L. and Alistair Shilton
Dept of Electrical and Electronics Engineering
The University of Melbourne, Parkville, Vic-3101, Australia
swami@ee.unimelb.edu.au

Abstract: The miniaturization of sensors has made possible the use of these tiny components in hostile environments for monitoring purposes in the form of sensor networks. Due to the fact that these networks often work in a data centric manner, it is desirable to use machine learning techniques in data aggregation and control. In this paper we give a brief introduction to sensor networks. One of the first attempts to solve the Geolocation problem using Support Vector Regression (SVR) is then discussed. We propose a method to localize a stationary, hostile radar using the Time Difference of Arrival (TDoA) of a characteristic pulse emitted by the radar. Our proposal uses three different Unmanned Aerial Vehicles (UAVs) flying in a fixed triangular formation. The performance of the proposed SVR method is compared with a variation of the Taylor Series Method (TSM) used for solving the same problem and currently deployed by the DSTO, Australia on the Aerosonde Mark III UAVs. We conclude by proposing the application of the SVR approach to more general localization scenarios in Wireless Sensor Networks.

1 INTRODUCTION TO SENSOR NETWORKS

Advances in hardware development have made available low cost, low power miniature devices for use in remote sensing applications. The combination of these factors has improved the viability of utilizing a sensor network consisting of a large number of intelligent sensors, enabling the collection, processing, analysis and dissemination of valuable information, such as temperature, humidity, salinity etc. gathered in a variety of environment.

A sensor network is an array (possibly very large) of sensors of diverse types interconnected by a communications network. Sensor data is shared between the sensors and used as input to a distributed or centralized estimation system which aims to extract as much relevant information from the available sensor data as possible. The fundamental attributes of sensor networks are reliability, accuracy, flexibility, cost effectiveness and ease of deployment. Sensor networks are predominantly data-centric rather than address-centric. That is, queries are directed to a region containing a cluster of sensors rather than to specific sensor addresses. Given the similarity in the data obtained by sensors in a dense cluster, aggregation of the data is performed locally. That is, a summary or analysis of the local data is prepared by an aggregator node within the cluster, thus reducing communication bandwidth requirements. Aggregation of data increases the level of accuracy and incorporates data redundancy to compensate for node failures.

There are many potential applications for sensor networks ranging from environment monitoring to defence. They find application in most areas where human deployment is uneconomical and/or risky. Habitat monitoring on Great Duck Island (Mainwaringa et al., 2002) and on the Great Barrier Reef (Kininmonth et al., 2004) are the best examples of environmental monitoring. Major efforts have been made to use sensor networks to monitor bush/forest fires. Investigations are also being undertaken into using sensor networks in health monitoring and emergency response. One such projects is the Code Blue project by Harvard University. They propose a new architecture for wireless monitoring and tracking of patients and first responders. Blood glucose, pulse monitoring and ECG monitoring are the commonest attributes monitored using wearable sensors. They have also been used to create smart homes. For a comprehensive list of applications, one can refer to the survey paper by Akyildiz et al. (Akyildiz et al., 2002).

Defence is one area where these networks find huge application in command, control, intelligence,

*An earlier version of this paper has been published in 2005-06 IEEE Proceedings of the Third International Conference on Intelligent Sensing and Information Processing

†Parts of this work is supported by ARC Research Network on Sensor Networks (www.sensornetworks.net.au) and Dept of Education Science and Technology (DEST-ISL) grant on Distributed Sensor Networks

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 27–33.

surveillance, reconnaissance and targeting systems (Martincic and Schwiebert, 2005). It is possible to monitor the personnel and equipment by fitting them with sensors. They can be widely used in enemy target tracking and monitoring, and play a major role in detecting agents in biological and chemical warfare (Martincic and Schwiebert, 2005).

The ability to cram more and more processing abilities into a smaller space with lower power requirements has enabled the deployment of such sensors in large numbers as well as development of more powerful machines with a large number of these sensors and processors: for example, Unmanned Aerial Vehicles (UAVs) with low payloads (of around 1 kg) are able to deliver astounding performance equipped with GPS, receiver/transmitter, processor and auto-pilot navigation. On-field battle surveillance using UAVs was used as early as the Gulf War and UAVs were extensively used in reconnaissance missions in the ongoing war in Iraq. It is estimated, and quite reasonably so, that UAVs will achieve a much greater presence in civilian applications in the coming decades in Road traffic monitoring, bushfire monitoring and resource monitoring to name a few. The UAV market is set to reach an annual gross of US$4.5 billion within a decade (Group, 2006). In this competitive scenario, there has been an enormous focus on algorithm development for UAVs geared towards development of more applications of these machines. The cornerstone of algorithm development for UAV applications has been localization, specifically target localization, which is crucial to all UAV applications in the army and will continue to be so in the case of civilian applications.

2 LOCALIZATION IN SENSOR NETWORKS

Localization in sensor network refers to the process of estimating the (x, y, z) coordinates of a particular object in the monitoring environment to orient any given node with respect to a global coordinate system (Martincic and Schwiebert, 2005; Bachrach and Taylor, 2005; Girod et al., 2002). Location discovery is critical in several applications and has been widely discussed in literature. Although GPS systems can be used, due to the line of sight problem, energy requirement and costs, other methods are proposed. Multilateration by distance measurement, Ad Hoc positioning systems and angle of arrival (AoA) are the most commonly used localization schemes. Localization can be either distributed or centralized. Centralized algorithms allow the use of sophisticated mathematical algorithms resulting in higher accuracy. However, data from every sensor node must be transferred back and forth to the base station resulting in high bandwidth and power requirements. For this reason beacon based distributed methods are rather more popular in localization. Multiple base stations with relatively higher computing capacities (called beacons) are used to aggregate the data locally (Langendoen and Reijers, 2003; Bachrach and Taylor, 2005). Several algorithms (He et al., 2006; Bahl and Padmanabhan, 2000) have been proposed including APIT, RF based diffusion schemes and bounding box schemes. More recently Nguyen et al. (Nguyen et al., 2005) showed that the coarse-grained and fine-grained localization problems for ad hoc sensor networks can be posed and solved as a pattern recognition problem using kernel methods from statistical learning theory. In this article, we describe a sample problem of estimating the location of fixed terrestrial RADAR using three Unmanned Aerial Vehicles (UAVs) where UAVs are considered to be mobile sensors. The target can be fixed or mobile; different strategies exist for tackling each problem. We use Support Vector Regression (Smola and Scholkopf, 1998). The following sections concentrate on the problem of RADAR localization using unmanned aerial vehicles.

2.1 Radar Localization using Support Vector Regression

For fixed target tracking, Time Difference of arrival (TDoA) is an important attribute. TDoA refers to the time delay for the same signal from the target to reach two different nodes (UAVs). This approach requires that the UAVs be far apart to have significant measurements of TDoA. It can also be extended to mobile targets but the tracking algorithm needs to be real-time in order that this can be achieved. TDoA based approaches to fixed target tracking are well documented in (Drake, 2004; Abel and Smith, 1987; Abel, 1976).

In (Kanchanavally et al., 2004) target detection is done using surrogate optimization based search, and coordination and tracking is achieved using a constrained diffusion of probability density on the hospitability maps via solution to the Fokker-Planck equation. In (Abel and Smith, 1987), the authors describe a spherical Interpolation based closed-form solution while (Abel, 1976) presents a Divide and conquer approach with Least squares estimation to solve the problem. Abel et al. (Abel, 1976) describes the estimation of a parameter vector θ that is to be used evaluate the mean $\mu(\theta)$ of a Gaussian distribution observation X. The Maximum Likelihood estimate, though statistically desirable, is difficult to compute when $\mu(\theta)$ is a non-linear function of θ. An estimate formed by combining the estimates from subsections of the data is presented as the divide and conquer approach and an application

to range difference based position estimation is described. In (Chan and Ho, 1994), another method for position estimation using TDoA is presented that is applicable to networks with large number of nodes tracking a single target, i.e. the number of TDoA measurements is greater than the number of unknowns (position vector of target that is 2-D or at max, 3-D).

To obtain a precise position estimate at reasonable noise levels, the Taylor-series method (Foy, 1976; Torrieri, 1984) is commonly employed. It is an iterative method that starts with an initial guess and improves the estimate at each step by determining the local linear least-squares (LS) solution. An initial guess close to the true solution is needed to avoid local minima. The selection of such a starting point is not simple in practice. Moreover, convergence of the iterative process is not assured. It is also computationally intensive since LS computation is required in every iteration. In this paper we describe the use of support vector regression for target tracking. We compare our algorithm with the existing Taylor series method and show that our method performs better.

2.2 Methodology

Suppose, three or more UAVs fly in a triangular formation, over a region believed to contain a RADAR installation. Each UAV acts as a receiver, carrying a sensor capable of detecting a RADAR pulse. Some prior knowledge is required of at least the frequency range of the enemy RADAR pulses so that the UAVs can "listen" in that range of frequencies. The receiver is not capable of detecting pulses over the entire spectrum of possible RADAR frequencies simultaneously. To solve this problem, bandwidth is divided into a number of frequency bins, which are selected for receiving a signal. Depending on the distance of each UAV from the RADAR, the pulse will take a different amount of time to reach a particular UAV. This time difference, along with the known position of each UAV, can be used to deduce the position of the RADAR. Figure 1 illustrates three UAVs located at an arbitrary distance from the RADAR.

The RADAR signal propagates at the speed of light. Therefore the exact time each UAV detects the pulse is a function of its distance from the RADAR. The time of arrival t_i of a pulse at a receiver i is given by:

$$t_i = t_{rad} + \frac{d_i}{c} \tag{1}$$

where t_{rad} is the time of emission of the pulse from the RADAR, d_i is the distance between receiver i and the RADAR and c is the speed of light. d_i can be expressed as $\|\vec{r}_i - \vec{r}_{rad}\|$ where $\vec{r}_i$ and $\vec{r}_{rad}$ denote the position vector of the i^{th} UAV receiver and the RADAR, respectively. Thus the Time of Arrival

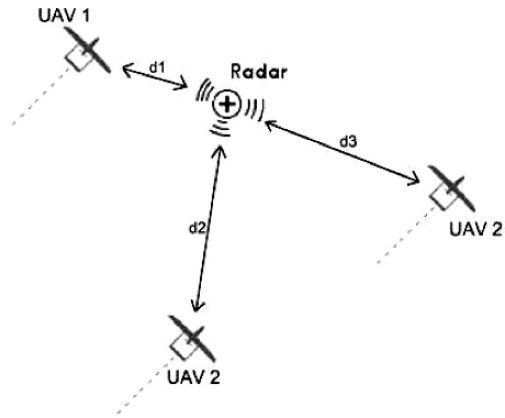


Figure 1: Schematic of 3 UAVs tracking a fixed hostile RADAR.

(ToA) of the pulse at receiver i can be expressed as:

$$t_i = t_{rad} + \frac{\|\vec{r}_i - \vec{r}_{rad}\|}{c} \tag{2}$$

Since the receivers cannot estimate the time of emission t_{rad}, we eliminate it by subtracting two such ToA equations to get the TDoA equation. Thus we obtain two TDoA equations using three ToA equations:

$$t_1 - t_2 = \Delta t_{12} = \frac{\|\vec{r}_1 - \vec{r}_{rad}\|}{c} - \frac{\|\vec{r}_2 - \vec{r}_{rad}\|}{c} \tag{3}$$

$$t_1 - t_3 = \Delta t_{13} = \frac{\|\vec{r}_1 - \vec{r}_{rad}\|}{c} - \frac{\|\vec{r}_3 - \vec{r}_{rad}\|}{c} \tag{4}$$

2.2.1 Comments on the RADAR Pulse

In the above equations it is assumed that all the UAVs receive the same pulse at different time instants. For this assumption to hold, the time interval between consecutive pulses emitted by the RADAR must be greater than the time taken by a pulse to reach the UAVs. This assumption holds in general because of the numbers involved. The RADAR scans the space around it by sending out pulses and waiting for reflected pulses. At a given time, it emits a pulse in a given direction with a main beam width of around 2°. This beam rotates with a frequency that is a few orders of magnitude less than the speed of light. In addition, there are side beams of lower power which ensures that the pulse is emitted in all directions in space and each of the UAVs will receive the pulse (maybe with different power levels, as the UAVs are definitely separated by more than the main beam width.) The frequency of RADAR pulse emission being in KHz and the time taken by a pulse to reach the UAVs being in microseconds ensures that the same pulse is received at all the UAVs. Note that we are referring to Discrete pulse emission frequency and not to radars that have a continuous time waveform as the output, in which

case the frequency of the RADAR wave has a range starting from a few MHz to more than 100 GHz (Association, 2006).

The unknowns in Eqs. 3 and 4 are x and y coordinates of the RADAR expressed in the vector $\vec{r}_{rad}$. We do not estimate the z-coordinate because a knowledge of (x, y) will give us the z-coordinate using the local relief map of the area when the RADAR is localized. The locus of $\vec{r}_{rad}$ is a hyperbola and the solution lies at the intersection of the two hyperbolae denoted by the two equations. Finding the solution is not easy due to their non-linear nature. When the number of unknowns equal the number of equations (as in the case above), an explicit solution can be obtained (Fang, 1990). Due to errors in the measurement of the TDoA, such an explicit solution is practically useless. Moreover, we need to investigate methods that are scalable to large number of UAVs tracking the same target and such methods must be able to make use of the extra information provided by the TDoA equations that outnumber the unknowns. First we present a Taylor series method (Drake and Dogancay, 2004) to solve the above equations. This method is currently being used by the DSTO in their simulations that are being developed to be used for RADAR tracking with Aerosonde Mark III UAVs.

2.3 The Taylor Series Method

As a first step to find the solution of Eqs. 3 and 4 an initial guess (x_0, y_0) is made using the asymptotes of each hyperbola to gain a relatively rough estimate of the intersection (Drake and Dogancay, 2004). This estimate is improved with each iteration by determining the local least-squares solution (Drake, 2004; Drake and Dogancay, 2004). The non-linear Eqs. 3 and 4 can be rewritten as:

$$c\Delta t_{1i} = \sqrt{(x_1 - x_{rad})^2 + (y_1 - y_{rad})^2} - \sqrt{(x_i - x_{rad})^2 + (y_i - y_{rad})^2} \tag{5}$$

where $i = 2, 3$.

The two equations are linearized and increments of x and y are computed as follows [27]:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \left(G_t^T Q^{-1} G_t\right)^{-1} G_t^T Q^{-1} h_t \tag{6}$$

where the matrices are given as:

$$G_t = \begin{bmatrix} \frac{x_1 - x_{rad}}{\|\vec{r}_1 - \vec{r}_{rad}\|} - \frac{x_2 - x_{rad}}{\|\vec{r}_2 - \vec{r}_{rad}\|} & \frac{y_1 - y_{rad}}{\|\vec{r}_1 - \vec{r}_{rad}\|} - \frac{y_2 - y_{rad}}{\|\vec{r}_2 - \vec{r}_{rad}\|} \\ \frac{x_1 - x_{rad}}{\|\vec{r}_1 - \vec{r}_{rad}\|} - \frac{x_3 - x_{rad}}{\|\vec{r}_3 - \vec{r}_{rad}\|} & \frac{y_1 - y_{rad}}{\|\vec{r}_1 - \vec{r}_{rad}\|} - \frac{y_3 - x_{rad}}{\|\vec{r}_3 - \vec{r}_{rad}\|} \end{bmatrix} \tag{7}$$

$$h_t = \begin{bmatrix} \|\vec{r}_{21}\| - (\|\vec{r}_2 - \vec{r}_{rad}\| - \|\vec{r}_1 - \vec{r}_{rad}\|) \\ \|\vec{r}_{31}\| - (\|\vec{r}_3 - \vec{r}_{rad}\| - \|\vec{r}_1 - \vec{r}_{rad}\|) \end{bmatrix} \tag{8}$$

All computations involving the RADAR coordinates (x_{rad}, y_{rad}) are done using the initial guess (x_0, y_0). Also, in the expression for h_t the values of $\|\vec{r}_{21}\|$ and $\|\vec{r}_{31}\|$ are known from the localization information on the UAVs themselves. The increment vector is computed until its magnitude becomes sufficiently small.

2.4 Support Vector Regression (SVR) Method

The SVR method looks at the RADAR localization problem in a function estimation sense. Support Vector Regression has evolved from Support Vector Machine (SVM) research on function estimation and is now a stand-alone research field of its own. In this paper, a basic familiarity with SVMs is assumed (Suykens et al., 2002; Smola and Scholkopf, 1998; Palaniswami and Shilton, 2000). We will provide a brief overview of the Least Squares-SVM (LS-SVM) methodology that has been used to implement the regression for the RADAR localization problem.

Given a set of training data $\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_n, y_n)\}$, where $\vec{x}_i \in \Re^{d_L}$ (a vector in *input space*) denotes the input and $y_i \in \Re$ the output, the goal of support vector regression is to find a function $f : \Re^{d_L} \to \Re$ that is optimal in-so-far as the error estimating y is minimized in some sense. In LS-SVR, the problem is formulated as follows.

We assume that the training data has been produced by some unknown map $g : \Re^{d_L} \to \Re$ and then corrupted by noise, so $y_i = g(\vec{x}_i) + \text{noise}$. Let us now define a mapping function $\vec{\varphi} : \Re^{d_L} \to \Re^{d_H}$ from input space to *feature space*. Using this we define an approximation to g:

$$f(\vec{x}) = \vec{w}^T \vec{\varphi}(\vec{x}) + b \tag{9}$$

In LS-SVMs, we seek to minimize the sum of square of errors in estimating y using the above approximation $f(\vec{x})$, subject to regularisation. The problem can be stated as follows:

$$\begin{aligned} &\min_{\vec{w},b,\vec{e}} J_P(\vec{w}, \vec{e}) = \tfrac{1}{2}\vec{w}^T\vec{w} + \tfrac{\gamma}{2}\sum_{k=1}^{n} e_k^2 \\ &\text{such that: } y_k = \vec{w}^T\vec{\varphi}(\vec{x}_k) + b + e_k \quad \forall k = 1, \ldots, n \end{aligned} \tag{10}$$

We will see that we are able to allow the mapping $\vec{\varphi}(\vec{x})$ to be non-linear as well as (potentially) infinite dimensional which in turn makes $\vec{w}$ (potentially) infinite dimensional. So, in order to solve the optimization problem, we work with the dual (Sundaram et al., 2005; Smola and Scholkopf, 1998; Suykens et al., 2002). Defining the dual variables $\vec{\alpha}$, the dual form is:

Solve in $\vec{\alpha}, b$:

$$\begin{bmatrix} 0 & \vec{1}_v^T \\ \vec{1}_v & \vec{\Omega} + \vec{I}/\gamma \end{bmatrix} \begin{bmatrix} b \\ \vec{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \vec{y} \end{bmatrix} \tag{11}$$

where $\vec{y} = [y_1; \ldots; y_n]$, $\vec{1}_v = [1; \ldots; 1]$, $\vec{\alpha} = [\alpha_1; \ldots; \alpha_n]$ and:

$$\begin{aligned} \Omega_{kl} &= \vec{\varphi}(\vec{x}_k)^T \vec{\varphi}(\vec{x}_l) \\ &= K(\vec{x}_k, \vec{x}_l) \qquad \forall k, l = 1, \ldots, n \end{aligned} \tag{12}$$

The resulting LS-SVM model for regression becomes

$$f(\vec{x}) = \sum_{k=1}^{n} \alpha_k K(\vec{x}, \vec{x}_k) + b \tag{13}$$

where the α_k, b are the solution to the dual problem which is a linear system. The most crucial property of SVM formulation is the nature of $K(\vec{x}_k, \vec{x}_l)$. Although $\vec{\varphi}(\vec{x})$ may be infinite dimensional (or at least very large), $K(\vec{x}_k, \vec{x}_l)$ is just a real number and may be calculated without use of $\vec{\varphi}$. K is called the kernel function in the SVM formulation and there are number of possible choices for kernels each with its own advantages and disadvantages. The most common kernels used are RBF, Sigmoid and Linear. In our simulations we use RBF kernels with $\gamma = 5.5$ and $\sigma = 1.3$.

For the RADAR localization problem, the SVR is used to estimate the mapping between the TDoA data and the RADAR's coordinates. Thus two separate SVRs were trained, one for each co-ordinate. This map is estimated over a rectangular grid aligned with the triangle formed by the UAVs and sharing the centroid of the triangle formed by the UAVs. Note that this mapping is invariant under translation of the UAVs as long as they maintain their relative positions over an approximately flat terrain. The flat terrain requirement may be eliminated by estimating the z-coordinate w.r.t. mean sea level, but this would require at least four UAVs. Three grids of consecutively lower dimensions 10km $\times$ 10km, 5km $\times$ 5km and 80m $\times$ 80m are trained for each coordinate thus making six SVRs in total. The implementation uses the SVM heavy library[3] (Shilton et al., 2005).

The training data was generated for each of the three grids by fixing the UAV locations and generating random RADAR locations within the grid. The size of the training set was 1000 since it was observed that there was no significant change in average error during testing for larger training sets.

Once the SVRs are trained, the testing proceeds as follows:

1. If the RADAR is hypothesized to be within the 10km $\times$ 10km grid, the corresponding SVR is activated by the received instantaneous TDoA data and hypothesizes a new location of the RADAR within the grid.
2. The UAVs are then navigated in such a fashion that the centroid of their triangular formation coincides with the latest hypothesized location. Then a better estimate of the location is hypothesized using the 5km $\times$ 5km grid
3. The UAVs are then navigated in such a fashion that the centroid of their triangular formation coincides with the latest hypothesized location. Then a better estimate of the location is hypothesized using the 80m $\times$ 80m grid

Hierarchical grids were chosen to get better estimates of the actual RADAR position. Average error given by eq. 14 show good performance as compared with the Taylor Series method which is why these grid measurements were frozen.

$$e_{avg} = \frac{1}{N_{test}} \sqrt{\sum_{k=1}^{N_{test}} \left\| \vec{r}_{RAD\_actual} - \vec{r}_{RAD\_est} \right\|^2} \tag{14}$$

2.5 Results and Discussions

The major source of error in the measure of the TDoA is called clock error. This refers to the error in measurement of ToA due to a lack of synchronization among the clocks in the UAVs. This error is typically of the order of a few hundred nano-seconds; as the TDoA itself is in hundreds of nanoseconds, this poses a challenge to any TDoA based estimation method. So, the testing of the Localization methods was carried with clock error. This was simulated as error in measurement of the TDoA and generated as a random noise variable added to the measured TDoAs. The noise was modeled as a Gaussian whose width was varied to simulate different levels of clock error. Table 1 shows the average localization error made by both methods with increasing values of the standard deviation of the clock error.

The other major source of error in the localization is due to relative movement of the UAVs that upsets the triangular configuration for which the SVRs are trained. The simplest way to minimize such errors is to take the TDoA measurements when the UAVs are in the required triangular configuration. However, the localization performance was simulated allowing random perturbations of different magnitudes in UAV positions. In this case too, these perturbations were generated as a Gaussian variable added to the position vector of the UAVs. The width of the Gaussian variable was varied from a few meters to up to 500 meters. The performance of the SVR trained for no perturbations is shown in Table 2 for situations with perturbed UAVs. The clock error in all these cases was kept at 50ns.

For a given problem with three UAVs, the computation time required for the TSM method for each TDoA measurement was 3.2 seconds on a Intel P4,

[3] http://www.ee.unimelb.edu.au/staff/swami/svm/

Table 1: Average Localization Error.

Clock Error (ns)	SVR Error (m)	TSM error (m)
0	4.9	0
50	19	75
100	38	190
200	77	304
300	115	420
400	149	655
500	188	696

Table 2: Average Localization Error with Perturbation.

Clock Error (ns)	Strength of Perturbation (m)	SVR error (m)
50	0	19
50	50	40
50	100	55
50	200	90
50	300	160
50	400	205
50	500	250

2.4GHz processor. The SVM testing, on the other hand was in the order of milliseconds. This is expected due to the fact that SVM testing is just a single computation of eq. 13. Clearly, the proposed SVR method outperforms the Taylor Series method, both in terms of computation time and accuracy. The TSM is clearly not scalable to multiple UAVs (UAV swarms) tracking a single target due to computational complexities. The SVR, on the other hand has the prominent advantage of easy extension to UAV swarms. Furthermore, the repetitive information provided by multiple UAVs is better utilized by the SVR approach.

2.6 Conclusion and Future Work

In this paper, we have introduced sensor networks and reviewed some important localization schemes. We have formulated the problem of target tracking as radar localization using multiple mobile sensors in the form of UAVs. A new method based on Support Vector Regression has been proposed for localization and compared with the existing system currently deployed by the DSTO, Australia on the Aerosonde Mark III UAVs.

An extension of this work is localization in three dimensions, which would require at least four UAVs. This would remove the dependence on terrain of the SVR performance, enhancing the effectiveness of this approach in unknown, hostile territory, which is crucial in applications such as reconnaissance and habitat monitoring. The next quantum jump is to use the SVR method for Mobile target localization in terrestrial Wireless Sensor Networks. This is one area where the low power and communication requirement of the SVR method are likely to prove beneficial.

REFERENCES

Abel, J. (1976). A divide and conquer approach to least-squares estimation. *IEEE Trans. Aerosp. Electron. Syst.*, 26:423–427.

Abel, J. and Smith, J. (1987). The spherical interpolation method for closed-form passive source localization using range difference measurements. In *IEEE Proceedings of ICASSP - 1987*, pages 471–474.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002). Wireless sensor networks: A survey, 38(4):393–422.

Association, A. E. W. (Feb. 14, 2006). http://www.aewa.org/library/rf_bands.html.

Bachrach, J. and Taylor, C. (2005). *Handbook of Sensor Networks - Ed. Stojmenovic, I.*, chapter 9: Localization in Sensor Networks. Wiley Series on Parallel and Distributed Computing.

Bahl, P. and Padmanabhan, V. N. (2000). Radar: An in-building rfbased user location and tracking system. In *IEEE Infocom*, pages 775–784.

Chan, Y. T. and Ho, K. C. (1994). A simple and efficient estimator for hyperbolic location. *IEEE Trans. Acoust., Speech, Signal Processing*, 42:1905–1915.

Drake, S. (2004). Geolocation by time difference of arrival technique. *Defence Science and Technology Organisation, Electronic Warfare and RADAR Division, Restricted Paper*.

Drake, S. P. and Dogancay, K. (2004). Geolocation by time difference of arrival using hyperbolic asymptotes. In *IEEE Proceedings of ICASSP - 2004*, pages 61–64.

Fang, B. T. (1990). Simple solutions for hyperbolic and related position fixes. *IEEE Trans. Aerosp. Electron. Syst.*, 26:748–753.

Foy, W. H. (1976). Position-location solutions by taylor-series estimation. *IEEE Trans. Aerosp. Electron. Syst.*, AES-12:187–194.

Girod, L., Bychkovskiy, V., Elson, J. and Estrin, D. (2002). Locating tiny sensors in time and space: A case study. In *Proceedings of the International Conference on Computer Design*, pages 16–18.

Group, T. (June 14, 2006). http://www.tealgroup.com.

He, T., Huang, C., Blum, B. M., Stankovic, J. A. and Abdelzaher, T. F. (2006). Range-free localization schemes for large scale sensor networks.

Kanchanavally, S., Zhang, C., Ordonez, R. and Layne, J. (2004). Mobile targettracking with communication

delays. In *43rd IEEE Conference on Decision and Control*, volume 3, pages 2899–2904.

Kininmonth, S., Bainbridge, S., Atkinson, I., Gill, E., Barral, L. and Vidaud, R. (2004). Sensor networking the great barrier reef.

Langendoen, K. and Reijers, N. (2003). Distributed localization in wireless sensor networks: a quantitative comparison. 34:499518.

Mainwaringa, A., Szewczyk, R., Culler, D. and Anderson, J. (2002). Wireless sensor networks for habitat monitoring. In *Proc. of ACM International Workshop on Wireless Sensor Networks and Applications*, pages 16–18.

Martincic, F. and Schwiebert, L. (2005). *Handbook of Sensor Networks - Ed. Stojmenovic, I.*, chapter 1: Introduction to Wireless Sensor Networking. Wiley Series on Parallel and Distributed Computing.

Nguyen, X., Jordan, M. I., and Sinopoli, B. (2005). A kernel-based learning approach to adhoc sensor network localization. 1(1):124–152.

Palaniswami, M. and Shilton, A. (2000). Adaptive support vector machines for regression. In *Proceedings of the 9th International Conference on Neural Information Processing, Singapore*, volume 2, pages 1043–1049.

Shilton, A., Palaniswami, M., Ralph, D., and Tsoi, A. C. (2005). Incremental training of support vector machines. *IEEE Transactions on Neural Networks*, 16(1):114–131.

Smola, A. and Scholkopf, B. (1998). A tutorial on support vector regression. *Neurocolt Technical report NC-TR-98-030, University of London*, pages 748–753.

Sundaram, B., Palaniswami, M., Reddy, S. and Sinickas, M. (2005). Radar localization with multiple unmanned aerial vehicles using support vector regression. In *IEEE Proceedings of Third International Conference on Intelligent Sensing and Information Processing (2005-06)*.

Suykens, J., Van Gestel, T., De Brabanter, J., De Moor, B. and Vandewalle, J. (2002). *Least Square Support Vector Machines*. World Scientific Publishing Company.

Torrieri, D. J. (1984). Statistical theory of passive location systems. *IEEE Trans. Aerosp. Electron. Syst.*, AES-20:183–199.

BRIEF BIOGRAPHY

Marimuthu Palaniswami obtained his B.E. (Hons) from the University of Madras, M.Eng. Sc. from the University of Melbourne, and PhD from the University of Newcastle, Australia.

He is an Associate Professor at the University of Melbourne, Australia. His research interests are the fields of computational intelligence, Nonlinear Dynamics, and Bio-Medical engineering. He has published more than 150 papers in these topics.

He was an Associate Editor of the IEEE Trans. on Neural Networks and is on the editorial board of a few computing and electrical engineering journals. He served as a Technical Program Co-chair for the IEEE International Conference on Neural Networks, 1995 and was on the programme committees of a number of internal conferences including IEEE Workshops on Emerging Technologies and Factory Automation, Australian Conferences on Neural Networks, IEEE Australia-New Zealand Conferences on Intelligent Information Processing Systems. He has given invited tutorials in conferences such as International Joint Conference on Neural Networks, 2000, 2001, and is invited to be tutorial speaker for the World Conference on Computational Intelligence. He has also given a number of keynote talks in major international conferences mainly in the areas of computational intelligence, computer vision and Biomedical Engineering. He has completed several industry sponsored projects for National Australia Bank, ANZ Bank, MelbIT, Broken Hill Propriety Limited, Defence Science an Technology Organization, Integrated Control Systems Pty Ltd, and Signal Processing Associates Pty Ltd.

He also received several ARCs, APA(I)s, ATERBS, DITARD and DIST grants for both fundamental and applied projects. He was a recipient of foreign specialist award from the Ministry of Education, Japan.

PART 1

Intelligent Control Systems and Optimization

MODEL PREDICTIVE CONTROL FOR DISTRIBUTED PARAMETER SYSTEMS USING RBF NEURAL NETWORKS

Eleni Aggelogiannaki, Haralambos Sarimveis
School of Chemical Engineering,NTUA, 9 Heroon Polytechniou str. Zografou Campus, 15780 Athens, Greece
Email: elangel@chemeng.ntua.gr, hsarimv@central.ntua.gr

Keywords: Distributed parameter systems, Model Predictive Control, Radial Basis Function Neural Networks.

Abstract: A new approach for the identification and control of distributed parameter systems is presented in this paper. A radial basis neural network is used to model the distribution of the system output variables over space and time. The neural network model is then used for synthesizing a non linear model predictive control configuration. The resulting framework is particular useful for control problems that pose constraints on the controlled variables over space. The proposed scheme is demonstrated through a tubular reactor, where the concentration and the temperature distributions are controlled using the wall temperature as the manipulated variable. The results illustrate the efficiency of the proposed methodology.

1 INTRODUCTION

In distributed parameter systems (DPS) inputs, outputs as well as parameters may change temporally and spatially due to diffusion, convection and/or dispersion phenomena. Such systems are quite common in chemical industries (tubular reactors, fluidized beds and crystallizers) and are mathematical described by systems of partial differential equations (PDE), where time and spatial coordinates are the independent variables.

The conventional approach for the synthesis of implementable control schemes for DPSs is based on methodologies that reduce the infinite order model to a finite (low) order model, which can capture the dominant behavior of the system. A comprehensive analysis of the recent developments in this direction can be found in Christofides (2001a). The most common approach found in the literature for an accurate model reduction implements a linear or a non linear Galerkin method to derive ODE systems that capture the slow (dominant) modes of the original DPS. In Christofides (2001b) one can find the analytical description of the linear Galerkin procedure as well as the nonlinear model reduction method which implements the concept of approximate inertial manifold. The resulting models are then used for synthesizing low dimensional robust output feedback controllers for quasi linear and nonlinear parabolic systems (Christofides and Daoutidis, 1996; 1997; Christofides, 1998; Shvartsman and Kevrekidis, 1998; Christofides and Baker 1999; Chiu and Christofides, 1999; El-Farra *et al.*, 2003; El-Farra and Christofides, 2004).

However, the analytical solution of the eigenvalue problem of the spatial differential operator is not always possible and consequently the selection of the appropriate basis to expand the PDEs is not an easy task. A systematic data driven methodology to address this problem is the Karhunen-Loève expansion (KL), also called proper orthogonal decomposition (POD) or empirical eigenfunctions (EEF) or principal component analysis. The KL expansion uses data snapshots and constructs the empirical eigenfunctions as a linear combination of those snapshots (Newman, 1996a; 1996b; Chatterjee, 2000). The resulting EEFs have been used as basis functions in the Galerkin procedure in a number of publications for accurate modelling and control in one-dimensional or two-dimensional systems. (Park and Cho, 1996a; 1996b; Park and Kim, 2000; Baker and Christofides, 1999; Shvartsman and Kevrekidis, 1998; Armaou and Christofides, 2002;)

The Galerkin procedure, mentioned so far uses analytical or empirical eigenfunctions and requires the mathematical description of the process, namely the exact system of PDEs. In case the PDEs are unknown, Gay and Ray (1995) proposed an identification procedure based on input-output data. The methodology employs integral equation models

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 37–42.

to describe the DPS and the singular value decomposition (SVD) of the integral kernel to produce an input/output model, suitable for model predictive control (MPC) methodologies. A comparison of the efficiency of this data driven model with the methods mentioned earlier can be found in Hoo and Zheng (2001). More recently, an identification method that combines KL and SVD for low order modeling and control have been presented (Zheng and Hoo, 2002; Zheng *et al.*, 2002a; 2002b; Zheng and Hoo, 2004). The discrete form of the SVD-KL method has also been used in MPC configurations with improved performance, comparatively to linear feedback controllers.

A neural network approach for the identification of DPSs has been attempted by Gonzáles-García *et al.* (1998) and more recently a combination of POD and neural networks has been proposed by Shvartsman *et al.* (2000). Padhi *et al.* (2001) used two sets of neural networks to map a DPS and a discrete dynamic programming format for the synthesis of an optimal controller. The same concept, also exploiting the POD technique for a lower order model, is presented by Padhi and Balakrishnan (2003).

In the present work, a radial basis function (RBF) neural network is proposed for the identification of non linear parabolic DPSs. RBF neural networks are quite popular for lumped system modeling because of their comparatively simple structure and their fast learning algorithms (Sarimveis *et al.*, 2002). In this paper the RBF neural network is formulated, so that it is able to predict the distribution of the output variables over space. This way, an estimation of the system outputs is available in any position. The RBF model is then implemented in a nonlinear MPC configuration to predict the controlled variables in a finite number of positions.

The rest of the article is formulated as follows: In section 2 the structure of the RBF neural network for DPSs is presented. In section 3 the non linear MPC configuration is described in more details. The proposed methodology is tested through the application described in subsection 4.1 The hidden efficiency of the RBF neural network is examined in subsection 4.2 and the controller performance in 4.3. In section 5, the final conclusions are summarized.

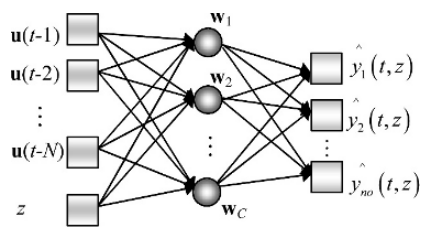


Figure 1: A radial basis function neural network of *C*. nodes for a distributed parameter system.

2 RBF NEURAL NETWORKS FOR MODELING DISTRIBUTED PARAMETER SYSTEMS

2.1 Quasi-linear Parabolic DPS

In general, a quasi linear parabolic distributed parameter system is described by a set of partial differential equations and boundary conditions of the form of Eq. (1):

$$\begin{aligned} &\frac{\partial \boldsymbol{\upsilon}(t,z)}{\partial t} = \mathbf{a}\frac{\partial^2 \boldsymbol{\upsilon}}{\partial z^2} - \mathbf{v}\frac{\partial \boldsymbol{\upsilon}}{\partial z} + \mathbf{u}(t,z) + \mathbf{G}(t,z), \quad \mathbf{a},\mathbf{v} > 0 \\ &\mathbf{y}(t,z) = \mathbf{C}(z)\cdot\boldsymbol{\upsilon}(t,z) \\ &\boldsymbol{\upsilon}(0,z) = \boldsymbol{\upsilon}_o, \quad 0 \le z \le L \\ &\frac{\partial \boldsymbol{\upsilon}}{\partial z}(t,0) = \mathbf{g}_o(t), \quad \frac{\partial \boldsymbol{\upsilon}}{\partial z}(t,L) = \mathbf{g}_l(t), \quad t > 0 \end{aligned} \tag{1}$$

where $\boldsymbol{\upsilon}(t,z)$ are the state variables, $\mathbf{u}(t,z)$ the manipulated variables and $\mathbf{y}(t,z)$ the controlled variables. $\mathbf{G}(t,z)$ is an additional non linear term of the model and $\mathbf{C}(z)$ is a function determined by the location of the sensors. Vectors $\boldsymbol{\upsilon}_o(z)$ and $\mathbf{g}_o(t)$, $\mathbf{g}_l(t)$ describe the initial and the Neumann boundary conditions of the system, respectively.

2.2 RBF Neural Network for DPS

Radial basis function networks are simple in structure neural networks that consist of three layers, namely the input layer, the hidden layer and the output layer. Development of an RBF network based on input-output data includes the computation of the number of nodes in the hidden layer and the respective centers and the calculation of the output weights, so that the deviation between the predicted and the real values of the output variables, over a set of training data, is minimized

An RBF neural network for modeling a DPS is constructed so that it can predict the values of the output variables at a specific spatial point (Figure 1). The input vector of such network at time point $t=kT_a$ (where T_a is the sample time) contains past values of the input variables and the coordinates in space, where we wish to obtain a prediction:

$$\mathbf{x}(t,z)=\left[\mathbf{u}^T(t-1)\ \mathbf{u}^T(t-2)\ \ldots\ \mathbf{u}^T(t-N)\quad z\right]^T \tag{2}$$

For simplification we limit our analysis in only one dimension in space. Generalization to three dimensions is straightforward.

The neural network output is a vector containing the values of the process output variables at the location that is specified in the input vector:

$$\hat{\mathbf{y}}_{\mathrm{RBF}}(t,z)=\left[\hat{y}_1(t,z)\quad \hat{y}_2(t,z)\ \ldots\quad \hat{y}_{no}(t,z)\right]^T \tag{3}$$

$$\hat{y}_j(t,z)=\sum_{c=1}^{C} w_{j,c}\cdot f\left(\left\|\mathbf{x}(t,z)-\mathbf{x}_c\right\|_2^2\right),\quad j=1,\ldots,no \tag{4}$$

In the previous equations N is the number of past values for the input vector, *no* is the number of the process output variables, C is the number of hidden nodes, $\mathbf{w}_c$ is the weight vector corresponding to the output of the *cth* node, f is the radial basis function and $\mathbf{x}_c$ is the center of the c node. The method utilized to train neural networks in this work is based on a fuzzy partition of the input space and is described in details in Sarimveis *et al.* (2002).

3 NONLINEAR MPC FOR DPS

The nonlinear MPC configuration that is proposed in this work for controlling DPSs, uses the RBF model to predict the values of the controlled variables over a future finite horizon *ph* at a number of locations *ns,* where measurements are available. Then, an optimization problem is solved, so that both the deviations of the controlled variables from their set points over the prediction horizon and the control moves over a control horizon *ch*, are minimized. The objective function is of the following form:

$$\min_{\substack{\mathbf{u}(t+k|t)\\ k=0,\ldots,ch-1}}\left(\sum_{j=1}^{ns}\sum_{k=1}^{ph}\left\|\mathbf{W}_{k,j}\left(\hat{\mathbf{y}}(t+k,z_j\,|\,t)-\mathbf{y}_j^{sp}\right)\right\|_2^2+\sum_{k=0}^{ch-1}\left\|\mathbf{R}_k\Delta\mathbf{u}(t+k\,|\,k)\right\|_2^2\right) \tag{5}$$

$$\hat{\mathbf{y}}(t+k,z_j\,|\,t)=\hat{\mathbf{y}}_{\mathrm{RBF}}(t+k,z_j)+\mathbf{d}(t,z_j\,|\,t),\quad j=1,\ldots,ns,\quad k=1,\ldots,ph \tag{6}$$

$$\mathbf{u}_{\min}\le\mathbf{u}(t+k\,|\,t)\le\mathbf{u}_{\max},\quad k=0,\ldots,ch-1 \tag{7}$$

where $\hat{\mathbf{y}}(t+k,z_j\,|\,t)$ is the prediction made at time point t for the output vector at time $t+k$ and at location z_j, *ns* is the number of sensors, $\mathbf{d}(t,z_j\,|\,t)$ is the estimated disturbance at time point *t,* considered constant over the prediction horizon and $\mathbf{y}_j^{sp}$ is the set point at the location of the j sensor. For $k=ch,\ldots,ph$ the manipulated variables are considered to remain constant. $\mathbf{W}_k$ and $\mathbf{R}_k$ are weight matrices of appropriate dimensions.

4 APPLICATION

4.1 Description of the Process

One typical distributed parameter system in chemical engineering is a tubular reactor, where variables depend on both time t and reactor length z. The mass and energy balances, concerning a first order reaction, diffusion and convention phenomena, are described by two quasi-linear PDEs with Neumann boundary conditions (Eqs. (8)-(11)).

$$\frac{\partial T}{\partial t}=\frac{1}{P_{eh}}\frac{\partial^2 T}{\partial z^2}-\frac{1}{L_e}\frac{\partial T}{\partial z}+\eta c\exp\left[\gamma\left(1-\frac{1}{T}\right)\right]+\mu\left(T_w(t,z)-T\right) \tag{8}$$

$$\frac{\partial c}{\partial t}=\frac{1}{P_{em}}\frac{\partial^2 T}{\partial z^2}-\frac{\partial c}{\partial z}-D_a c\exp\left[\gamma\left(1-\frac{1}{T}\right)\right] \tag{9}$$

$$z=0,\ -\frac{\partial T}{\partial z}(t,0)=P_{eh}\cdot\left(T_i(t)-T(t,0)\right),\ z=1,\ \frac{\partial T}{\partial z}(t,1)=0 \tag{10}$$

$$z=0,\ -\frac{\partial c}{\partial z}(t,0)=P_{em}\cdot\left(c_i(t)-c(t,0)\right),\ z=1,\ \frac{\partial c}{\partial z}(t,1)=0 \tag{11}$$

where $T(t,z)$, $c(t,z)$ are dimensionless temperature and concentration respectively inside the reactor, $T_i(t)$, $c_i(t)$ are dimensionless temperature and concentration at the entrance of the reactor and $T_w(t,z)$ is the wall temperature. The values of the parameters of Eqs. (8)-(11) can be found in previous publications (Hoo and Zheng, 2001; 2002).

4.2 RBF Model Efficiency

An input-output training set was created using the wall temperature T_w, at $z=$ [0 0.33 0.66] as the manipulated variable, while the output variables (temperature and concentration) were recorded at 21 spatial locations. The PDEs were solved using the PDE Matlab toolbox. More specifically, we simulated the system by changing randomly the input variables and recording the output responses using a sample period of T_a=0.5 time units. The training set consisting of 2000 data points was generated considering N=3 past values of each manipulated variable. Deviation variables were used by subtracting from all the input and output values the corresponding steady states. Several neural

network structures were developed by changing the initial fuzzy partition in the fuzzy means training algorithm. The produced neural networks were tested using a new validation data set of 500 data that was developed in the same way with the training set, but was not involved in the training phase. The sum of squares errors (SSEs) for the different RBF structures are presented in Table 1. In Figure 2, the actual values and the predictions of the neural network consisting of 152 nodes are compared.

Table 1: Performance of RBF neural networks.

Hidden nodes C	SSE T	SSE c
13	0.2012	0.8873
27	0.1251	0.7523
68	0.0535	0.3628
86	0.0404	0.2232
152	0.0332	0.1420
207	0.0295	0.1104

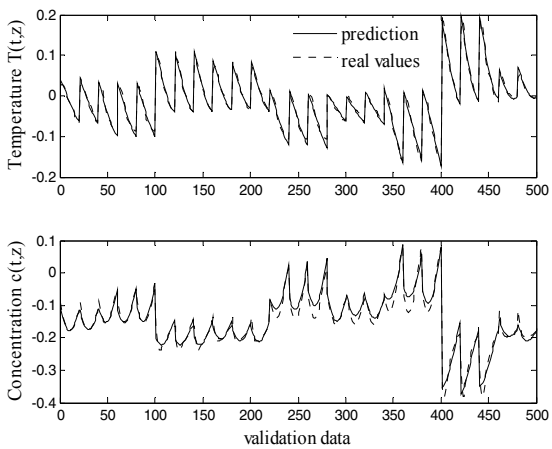


Figure 2: Actual values and predictions of the deviation variables for a neural network consisting of 152 hidden nodes.

4.3 MPC Performance

To test the proposed MPC configuration, we first simulated the example presented in Zheng and Hoo (2002). In that case, the temperature is the only controlled variable at z=[0:0.25:1] where we assume that sensors are available, while concentration is measured at z=1 but is not controlled. A disturbance is introduced to the system by decreasing the feed concentration C_i by 5%. We tested the proposed MPC scheme using for prediction the RBF network that consists of 27 nodes and the following parameter values: ch=6, ph=10, $\mathbf{W}$=1, $\mathbf{R}$=5·$\mathbf{I}_3$. The optimization problem that was formulated at each time instance was solved using the *fmincon* Matlab function. The performance of the controller is depicted in Figure 3, where the temperature distributions at the initial steady state and after 7 time units are compared. The responses at locations where sensors are available are also presented in the same figure. The proposed controller managed to reject the disturbance and produce zero steady state error. The obtained responses outperform the performances of a PI controller and an MPC configuration that utilizes the SVD-KL model. The responses of the two controllers are presented in Hoo and Zheng, (2002) and are not shown here due to space limitations. The temperature at the exit of the reactor returns to its initial value after 1.5 time units, while 6 time units are required by the system to produce zero steady state error along the length of the reactor.

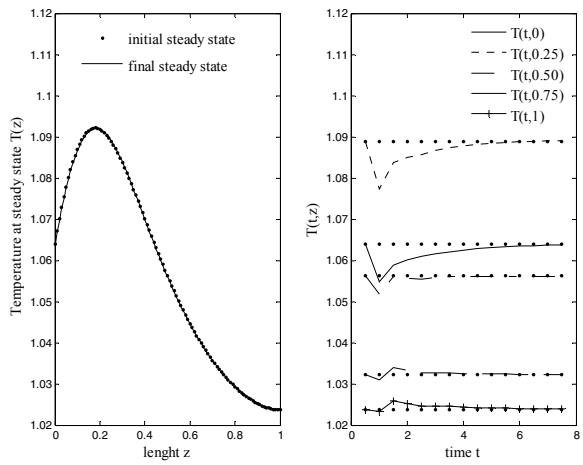


Figure 3: The final temperature distribution and the dynamic response to a 5% decrease in C_i using the RBF model.

A second performance test forces the system to reach a new steady state distribution. The actual steady state, where the temperature finally settles, is compared with the desired set point in Figure 4. The dynamic responses at locations where sensors are available are also presented in the same figure. The responses show that the system approaches the desired values quickly, avoiding overshoots. The behavior of the manipulated variables is depicted in Figure 5.

The last simulation presented in this work uses concentration at the reactor exit as an additional controlled variable. As far as the temperature profile is concerned, the target is to reach the same set point change as previously. Figures 6 and 7 present the responses of the temperature (at locations where sensors are available) and the concentration (at the reactor exit) respectively. They also present the final

distribution of both variables, after 20 time units. Figure 8 depicts the control actions over time. It is obvious that due to the additional controlled variable the performance of the controller is slightly deterio–rated as far as the dynamic behavior is concerned. However, the desired steady state is still approached satisfactorily.

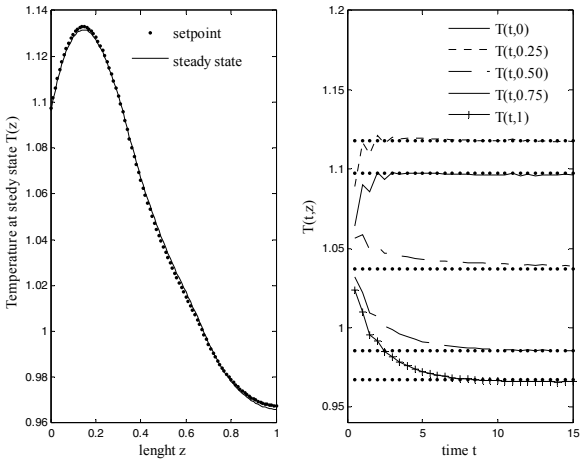


Figure 4: The temperature distribution after 15 time units and the dynamic response to a set point change.

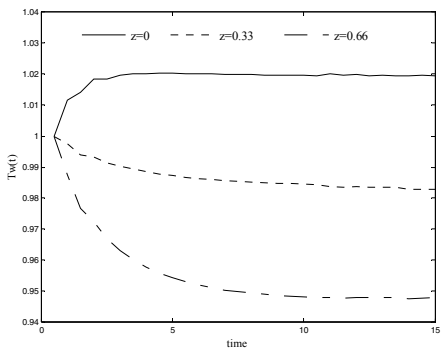


Figure 5: The manipulated variable $T_w(t)$ at z=0, 0.33 and 0.66.

5 CONCLUSIONS

A nonlinear input/output identification method for distributed parameter systems is proposed is this paper. An RBF neural network capable to predict the output variables over space is developed. The accuracy of the neural network was established through a tubular reactor simulation. The model is then used for the synthesis of a MPC configuration that minimizes the deviation of the prediction of the controlled variables at a finite number of positions, where a sensor is assumed to exist. The proposed method produced satisfactory results in both disturbance rejection and set point change problems. The performance of the controller was found to be superior to PI controllers or linear MPC configu-rations presented in former publications.

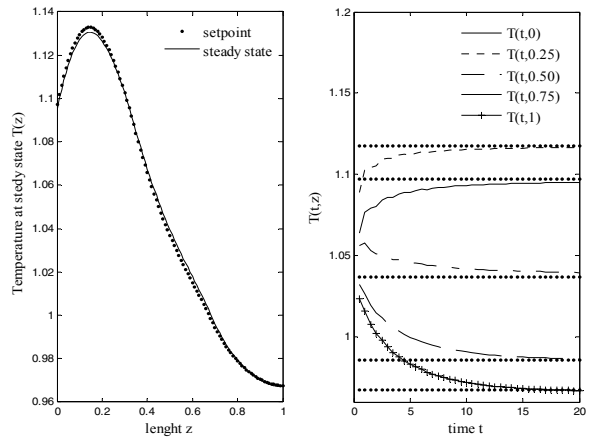


Figure 6: The temperature distribution after 20 time units and responses to a set point change when considering $c(t,1)$ as an additional controlled variable.

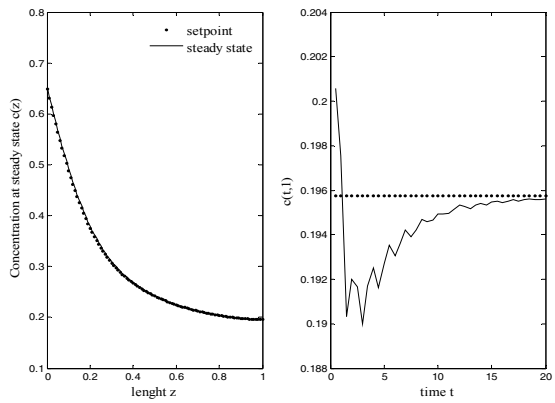


Figure 7: The concentration distribution after 20 time units and the response of $c(t,1)$.

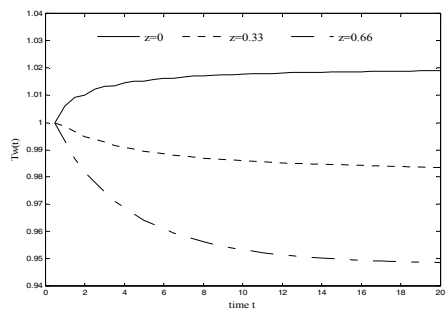


Figure 8: The manipulated variable $T_w(t)$ at z=0, 0.33 and 0.66 when considering $c(t,1)$ as an additional controlled variable.

REFERENCES

Armaou, A., Christofides, P. (2002). Dynamic optimization of dissipative PDE systems using nonlinear order reduction. *Chem. Eng. Sc.*, 57, 5083-5114.

Baker, J., Christofides, P. (1999). Output feedback control of parabolic PDE systems with nonlinear spatial differential operators. *Ind. Eng. Chem. Res.*, 38, 4372-4380.

Chatterjee, A. (2000). An introduction to the proper orthogonal decomposition. *Current Science*, 78, 808-817.

Chiu, T., Christofides, P. (1999). Nonlinear control of particulate processes. *AIChE Journal*, 45, 1279-1297.

Christofides, P. (1998). Robust control of parabolic PDE systems. *Chem. Eng. Sc.,* 53, 2949-2965.

Christofides, P. (2001a). Control of nonlinear distributed process systems: recent developments and challenges. *AIChE Journal*, 47, 514-518.

Christofides, P. (2001b). *Nonlinear and Robust Control of PDE Systems: Methods and Applications to transport reaction processes.* Birkhäuser, Boston.

Christofides, P. J. Baker, (1999). Robust output feedback control of quasi-linear parabolic PDE systems. In *Systems & Control Letters*, 36, 307-316.

Christofides, P., Daoutidis, P. (1996). Nonlinear control of Diffusion-Convection-Reaction processes. *Comp. Chem. Eng.,* 20, 1071-1076.

Christofides, P., Daoutidis, P. (1997). Finite-Dimensional Control of Parabolic PDE Systems Using Approximate Inertial Manifolds. *Journal of mathematical analysis and applications,* 216, 398-420.

El-Farra, N., Armaou, A., Christofides, P. (2003). Analysis and control of parabolic PDE systems with input constraints. *Automatica,* 19, 715-725.

El-Farra, N., Christofides, P. (2004). Coordinating feedback and switching for control of spatially distributed process. *Comp. Chem. Eng.,*28, 111-128.

Gay, D., Ray, W. (1995). Identification and control of distributed parameter systems by means of the singular value decomposition. *Chem. Eng. Sc.*, 50, 1519-1539.

Gonzáles-García, R., Rico-Martínez, R., Kevrekidis, I. (1998). Identification of distributed parameter systems: A neural net based approach. *Comp. Chem. Eng.,* 22, 965-968.

Hoo, K., Zheng, D. (2001). Low order control-relevant models for a class of distributed parameter systems. *Chem. Eng. Sc.,* 56, 6683-6710.

Newman, A. (1996a). Model reduction via the Karhunen-Loève Expansion Part I: An exposition. *Technical Report 96-32,* University of Maryland, College Park, MD.

Newman, A. (1996b). Model reduction via the Karhunen-Loève Expansion Part II: Some elementary examples. *Technical Report 96-33,* University of Maryland, College Park, MD.

Padhi, R., Balakrishnan, S. (2003). Proper orthogonal decomposition based optimal neurocontrol synthesis of a chemical reactor process using approximate dynamic programming. *Neural Networks,* 16, 719-728.

Padhi, R., Balakrishnan, S., Randolph, T. (2001). Adaptive-critic based optimal neuro control synthesis for distributed parameter systems. *Automatica*, 37, 1223-1234.

Park, H., Cho, D. (1996a). The use of Karhunen-Loeve decomposition for the modelling of distributed parameter systems. *Chem. Eng. Sc.*, 51, 81-89.

Park, H., Cho, D. (1996b). Low dimensional modelling of flow reactors. *Int. J. Heat Mass Transfer*, 39, 3311-3323.

Park, H., Kim, O. (2000). A reduction method for the boundary control of the heat conduction equation. *Journal of Dynamic Systems Measurement and Control*, 122, 435-444.

Sarimveis, H., Alexandridis, A., Tsekouras, G., Bafas, G. (2002). A fast and efficient algorithm for training radial basis function neural networks based on a fuzzy partition of the input space. *Ind. Eng. Chem. Res.*, 41, 751-759.

Shvartsman, S., Kevrekidis, I. (1998). Nonlinear model reduction for control of distributed systems: a computer –assisted study. *AIChE Journal*, 44, 1579-1595

Shvartsman, S., Theodoropoulos, C., Rico-Martínez, R. Kevrekidis, I., Titi, E., Mountziaris, T. (2000). Order reduction for nonlinear dynamic models of distributed reacting systems. *Journal of Process Control,* 10, 177-184.

Zheng D., Hoo, K. (2002). Low-order model identification for implementable control solutions of distributed parameter systems. *Comp. Chem. Eng.,* 26, 1049-1076.

Zheng, D., Hoo, K. (2004). System identification and model-based control for distributed parameter systems. *Comp. Chem. Eng.,* 28, 1361-1375.

Zheng, D., Hoo, K., Piovoso, M. (2002a). Low-order model identification of distributed parameter systems by a combination of singular value decomposition and the Karhunen-Loève expansion. *Ind. Eng. Chem. Res.,* 41, 1545-1556.

Zheng, D., Hoo, K., Piovoso, M. (2002b). Finite dimensional modeling and control of distributed parameter systems. In *2002 American Automatic Control Conference,* A.A.C.C.

FUZZY DIAGNOSIS MODULE BASED ON INTERVAL FUZZY LOGIC: OIL ANALYSIS APPLICATION

Antonio Sala
Systems Engineering and Control Dept., Univ. Politécnica de Valencia
Cno. Vera s/n, 46022 Valencia, Spain
Email: asala@isa.upv.es

Bernardo Tormos, Vicente Macián, Emilio Royo
CMT Motores Térmicos, Univ. Politécnica de Valencia
Cno. Vera s/n, 46022 Valencia, Spain
Email: betormos,vmacian,emrocar@mot.upv.es

Keywords: Fuzzy expert systems, fuzzy diagnosis, uncertain reasoning, interval fuzzy logic.

Abstract: This paper presents the basic characteristics of a prototype fuzzy expert system for condition monitoring applications, in particular, oil analysis in Diesel engines. The system allows for reasoning under absent or imprecise measurements, providing with an interval-valued diagnostic of the suspected severity of a particular fault. A set of so-called metarules complements the basic fault dictionary for fine tuning, allowing extra functionality. The requirements and basic knowledge base for an oil analysis application are also outlined as an example.

1 INTRODUCTION

In diagnosis of industrial processes, there is a significant practical interest in developing technologies for a more effective handling of the information available to ease the procedures of inspection and maintenance (I/M) by means of greater automation.

Computer-aided diagnosis is one of the earliest fields of applications of artificial intelligence tools (Russell and Norvig, 2003). Logic and statistical inference have been tried in previous applications (Wang, 2003), being medical diagnosis the most widely considered target application (Chen et al., 2005). However, industrial fault diagnosis (Chiang et al., 2001; Russell et al., 2000) is an appealing problem in itself, in sometimes more reduced contexts allowing for better accuracy.

The full diagnostic problem under uncertain data would need to be considered in a probabilistic framework. Taking into account the "gradualness" of the symptoms and possible diagnostics of varying degree of severity (captured by fuzzy logic), the most complete approach would be setting up a continous Bayesian network (or an hybrid one). The Bayesian network paradigm arose in the last decade as a probabilistic alternative to reasoning, superior to truth-maintenance approaches in some cases (see (Russell and Norvig, 2003) and references therein).

Indeed, truth maintenance systems cannot cope with contradictory information, whereas probabilistic settings may integrate any piece of information, as long as nonzero a priori probabilities are assigned to all events. Also, with a reduced amount of information, logic-based systems tend to produce many possible diagnostics; probabilistic ranking is out of their capabilities, at least in a basic setting: note for instance that, in principle, fuzzy truth values are not related to probability ones (Dubois and Prade, 1997). There are expert systems with carry out probabilistic inference (Lindley, 1987).

However, inference on the Bayesian network paradigm is intractable in a general case (NP-hard). If the amount of uncertainty is low (if a significant subset of the possible measurements is always obtained and the "determinism" of the underlying system is acceptable), then fuzzy logic-based approaches to reasoning may be a viable solution in practice. This is the case of some industrial diagnosis problems, such as oil analysis, to which the system in development is targeted.

Some works in literature discussing condition monitoring (diagnostic and supervision tasks) using fuzzy logic are, for instance (Carrasco and *et. al.*, 2004;

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 43–50.

Chang and Chang, 2003). Logic uncertainty can be accommodated, for instance, by possibility theory (Cayrac et al., 1996), or by interval-valued fuzzy logic (Entemann, 2000), related to the so-called intuitionistic fuzzy sets (Szmidt and Kacprzyk, 2003; Atanasov, 1986): membership to a fuzzy set has a minimum and a maximum and an ordinary fuzzy set or truth value is obtained considering them as equal. This latter approach is the one followed in this work.

Condition monitoring can also be dealt with with model-based approaches (Isermann and Ballé, 1997; Chiang et al., 2001), if enough quantitative descriptions of the system are available; however, such model-based approaches including algebraic and differential equations, engineering tables, *etc.* can only be understood by specialists on a particular process and their conclusions are obscure and difficult to be explained to an end-user which does not know the internals of the system in consideration.

This paper presents the structure of a fuzzy inference module that incorporates some innovations easing the setting up of rules and improving the quality of the final diagnostic conclusions. In particular, the use of interval fuzzy logic, the methodology to deal with exceptions and the possibility of expressing different alternatives for the same diagnostic and, if they do not agree, firing a fuzzy contradiction warning.

An application of the system is presently being tried on an oil analysis task whose main requirements appear in (Macián et al., 1999). Basically, metallic components (mostly from wear particles), oil properties (viscosity, detergency, pH, *etc.*) and contaminants (silicon, water, *etc.*) are determined in a standard analysis and some condition states of the Diesel engine from which the oil sample was taken should be determined.

This paper presents, in two sections, the structure of the fuzzy condition monitoring module being developed and the key concepts of the oil analysis application in which the possibilities of the system are being tested. The first section discusses, in some subsections, specific components of the diagnosis module. A conclusion section summarises the main ideas and results.

2 THE FUZZY CONDITION MONITORING MODULE

The presented fuzzy condition monitoring module takes raw data from a file (oil analysis results in the particular application test bench) and outputs a file with the identified diagnostic results, ordered by severity (intensity of fault), associated to the degree of truth of a particular conclusion.

It is structured in the following main submodules:

- measurement preprocessing
- fuzzy rule base inference
- postprocessing of the conclusions

Each of them will be discussed in the sequel.

2.1 Measurement Preprocessing

Raw data from sensors may need some sort of preprocessing prior to rule evaluation. Indeed, if nonlinearity inversion, statistical calculations, dynamic processing, etc. are carried out beforehand, then the subsequent rules will be simpler. This preprocessing is, however, application-specific in most cases (see Section 3). In general, normalising by substraction of mean and division by variance is useful in most applications. In other cases, expressing the measured variables in terms of percentage of a reference value is also beneficial for user understanding of the subsequent rules.

Incomplete or absent measurements: In the case of incomplete information, the measurements can be given in interval form, and fuzzy reasoning will be carried out via generalisation of ordinary rules to interval-valued logic values, as described in Section 2.3, giving rise to an interval output of estimated severities. An absent measurement will produce $[0, 1]$ as the possible interval of truth values.

2.2 Fuzzy Inference Module

The fuzzy inference module has been also built with different submodules:

- variable definition submodule,
- rulebase definition submodule, and
- inference engine.

Variable definition. The name, operating range and applicable "concepts" (fuzzy sets) on the variables are defined via a suitable syntax. An example appears below on the variable "CU" (copper concentration):

```
CU NORMAL 5 1
CU HIGH 5 20 50 100
CU VERYHIGH 50 150
```

the first number defining the support of the fuzzy set, the second one defining the core.

The last line defines, for instance, that the concentration of copper is not "very high" if it is below 50 ppm, and then, gradually starts to be considered "very high" up to 150 ppm where it is considered 100% abnormal, indicating that rules related to this concept would fire a "severe" fault. In the intermediate ranges, the rules would conclude a fault with an "intermediate" severity. The middle line `CU HIGH` describes a trapezoidal membership function via its four characteristic points (left support extreme, left core extreme, right core extreme, right support extreme).

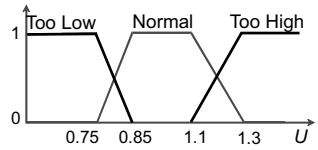


Figure 1: Fuzzy partition.

With trapezoidal and left and right limit functions (the three shapes appearing in Figure 1) the needs of most users are satisfied, and so it was in the application to be later referred. In this way, by restricting to piecewise linear membership functions, calculation of maximum and minimum membership for an interval input is easy, as the extremal values in a piecewise linear $\mu(x)$ are reached either at the extremes of the interval of x or at some of the characteristic points of μ (*i.e.*, those where a change of slope does occur).

Rulebase definition. The rulebase is defined by means of a set of rules in the form:

```
Disorder  Symptom-List END
```

They conform the core of the rulebase, and the elements in the symptom list are assumed to be linked by an "AND" connective. They will detect "basic" events from which a more complicated diagnosis may be later extracted. For examples, see Section 3.

Inference is carried out by evaluating the minimum of the severity of the symptoms in the symptom list of a particular disorder, and assigning that value to the severity of the associated disorder (see later). If some "OR" connectives were to be used, it can be done by means of the so-called metarules to be defined.

Symptom relevance modifier. Each symptom may be affected by a coefficient indicating that its presence confirms the fault, but its absence (in the presence of the rest) indicates a milder severity. For instance, the rule:

```
FAULT2
    SYMPTOM1 ABNORMAL
    SYMPTOM2 ABNORMAL 0.5
END
```

indicates that simultaneous presence of symptoms 1 and 2 fires fault 2 with full severity; however, presence of only symptom 1 indicates a fault of intermediate severity.

Metarules. The basic rules can be used to detect particular situations of interest, with an estimated interval of severity as a result. These situations are, many times, the ultimate faults to be detected.

However, there are occasions where they must be combined. This combination may have a logical interpretation in terms of AND, OR, NOT; in this case, the so-called metarules are introduced to handle the situation. One possible structure is:

```
DISORDER IF LOGIC-EXPRESSION
```

where `LOGIC-EXPRESSION` is any user-defined combination of symptoms or previously inferred atomic disorders, with conjunction, disjunction and negation operators. This rules will be denoted as `MIF` rules.

Note that the basic rules could have been embedded into this syntax. However, the proposed one allows for the incorporation of relevance modifiers that might be more cumbersome in the middle of a logic expression, improving the readability for the end-user.

Another type of metarule is the one in the form:

```
DISCARD Disorder IF LOGIC-EXPRESSION
```

used, for instance, to discard a "general" fault if a more specific situation sharing the same symptoms (plus some other ones) is encountered, or to set up "exceptions" to rules. Again, in Section 3, some examples appear.

Other types of metarules (UNION and ALTERNATIVE) will be later described.

2.3 Inference Methodology

This section discusses the inference mechanism used to evaluate the presented rules and metarules.

The core of the inference system works with fuzzy *interval uncertain propositions* (Entemann, 2000). In these propositions, their truth value is an *interval* $[\nu, \pi]$, where $0 \leq \nu \leq 1$ and $0 \leq \pi \leq 1$. It describes partial knowledge: the minimum value it can attain given the present information is called *necessity* ν and the maximum value will be denoted as *possibility* π. If they do coincide, then the proposition can be considered an ordinary fuzzy proposition[1].

Connectives. Connectives allow combination of atomic propositions into more complex ones by conjunction, disjunction and negation. Assuming a particular connective for a fuzzy proposition is given (such as T-norms for AND, T-conorms for OR, etc. (Weber, 1983) in a non-intervalic fuzzy logic) in the fuzzy-uncertain framework, the resulting interval will be evaluated with interval arithmetic, *i.e.*, given a fuzzy connective $C : [0,1] \times [0,1] \rightarrow [0,1]$, $Y = C(X_1, X_2)$, its interval version is[2]:

$$[\nu_Y, \pi_Y] = \left[\min_{\substack{\mu_1 \in [\nu_1, \pi_1] \\ \mu_2 \in [\nu_2, \pi_2]}} C(\mu_1, \mu_2), \max_{\substack{\mu_1 \in [\nu_1, \pi_1] \\ \mu_2 \in [\nu_2, \pi_2]}} C(\mu_1, \mu_2) \right] \quad (1)$$

[1] Note that the interpretation of possibility and necessity is related but not equivalent to other approaches such as (Cayrac et al., 1996)

[2] Those expression apply when the propositions refer to *independent, non-interactive* variables. As in ordinary interval arithmetic, the above expression (1) in complex propositions yields conservative (excessively uncertain) results when correlation and multi-incidence in the arguments is present.

Regarding negation, the truth degree of $\neg p_1$ is defined as the interval $[\nu = 1 - \pi_1,\ \pi = 1 - \nu_1]$.

For instance, using the minimum and maximum as conjunction and disjunction operators, the intervalisation of AND and OR operations is given by the expressions:

$$[\nu_1, \pi_1] \wedge [\nu_2, \pi_2] = [\min(\nu_1, \nu_2), \min(\pi_1, \pi_2)] \quad (2)$$

$$[\nu_1, \pi_1] \vee [\nu_2, \pi_2] = [\max(\nu_1, \nu_2), \max(\pi_1, \pi_2)] \quad (3)$$

Let us consider now how inference is carried out in the basic rules and in the metarules.

Basic Rules: The AND intervalic operation (2) above is used (or its trivial generalisation to more intervals).

Furthermore, if a particular symptom has an "irrelevance factor" ρ its membership value μ is transformed to $\rho * (1 - \mu) + \mu$ before carrying out the interval conjunction.

For instance in the `FAULT2` example in the previous page, if symptom 1 fired with intensity 0.85 and symptom 2 did with intensity 0.8, symptom 2 will be transformed to 0.5*0.2+0.8=0.9 before carrying out conjunction (with a final result of 0.85).

In fact, the implemented approach considers the so-called inference error (Sala and Albertos, 2001) so that given a logic value μ, the inference error is $e = 1 - \mu$ (extended to interval arithmetic). Then, given a list of q symptoms in a rule, the overall inference error is:

$$E = \sqrt[p]{\sum_{i=1}^{q} (e_i)^p} \quad (4)$$

So with $p = 2$ the methodology could be denoted as Euclidean inference. With $p \to \infty$, the result is the same as the interval AND (using minimum) described above. By fixing the value of p the user may specify a different behaviour (the lower p is, the more severity is substracted due to partially non-fired symptoms).

The formula (4) can be generalised to intervals, obtaining the lowest inference error by using the norm of the minimum inference error of each symptom, and the highest one with the maxima of the inference error intervals. Membership values are recovered from the resulting inference error figures by means of a negation formula.

Metarules: The logic operations will use the above intervalar expressions when evaluating any LOGIC-EXPRESSION in `MIF` metarules.

In the `DISCARD` metarules, the interval-arithmetic substraction will be used, *i.e.*:

`DISCARD` $[\nu_1, \pi_1]$ `IF` $[\nu_2, \pi_2]$

will give as a result the interval $[\nu_1', \pi_1']$:

$$\nu_1' = \max(0, \nu_1 - \pi_2) \qquad \pi_1' = \max(0, \pi_1 - \nu_2)$$

Membership value transformations. In some cases, one would like to introduce a set of rules detecting conditions for an intermediate fault and different conditions for a severe one (say, F1):

IF COND1 THEN F1 INTERMEDIATE
IF COND2 THEN F1 SEVERE (5)

To deal with this case, the tool under discussion allows *linear* transformations of membership via the so-called `UNION` metarule:

```
UNION FAULTNAME
  IDENT1 l11 l12 IDENT2 l21 l22 ...
```

The coefficients l_{i1} and l_{i2} define a linear transformation $\mu = l_{i1} * (1 - \mu) + l_{i2} * \mu$ to be carried out on the membership of identifier "IDENT-i". Afterwards, an interval-logic OR is carried out. Obviously, to combine a particular condition with no membership transformation, the setting $l_{i1} = 0$ and $l_{i2} = 1$ must be used. For instance, the above case (5) would be encoded by:

```
UNION F1
   COND1 0 0.5 COND2 0.5 1
```

In fact, the linear mapping actually implemented is not exactly the one above. It is:

$$\mu' = \begin{cases} l_{i1} * (1 - \mu) + l_{i2} * \mu & \mu \geq 0.02 \\ 0 & \mu < 0.02 \end{cases} \quad (6)$$

In that way, it can be specified that an intermediate severity fault must be suspected if any nonzero activation of a particular condition holds, but no firing will occur if none of the conditions are active above a significant threshold (0.02).

Alternatives. A closely related situation arises when several alternatives for diagnosing the same fault exist. If all measurements were available, all of them should produce the same result so specifying only one of them in the rulebase will do. However, to improve results accounting for missing or imprecise measurements, several of these alternative rules may be intentionally specified. In that case, to allow combining different alternatives into one diagnostic, the *intersection* of the intervals produced by inference on each of them will be the produced conclusion of the inference.

This is implemented in the current tool by an `ALT` metarule, with a syntax similar to the union metarule:

```
ALT FAULTNAME
  IDENTIFIER1 l11 l12
  IDENTIFIER2 l21 l22
  ...
```

allowing also a linear membership transformation, identical to (6), before the interval intersection is calculated.

For instance, let us assume that, after the membership transformations, if any, alternative A yields $[\nu_1, \pi_1]$ and alternative B yields $[\nu_2, \pi_2]$ as estimated severity intervals. If $\pi_1 > \nu_2$, then the intersection is

not empty and the following estimated severity interval is produced:

$$[\max(\nu_1, \nu_2), \min(\pi_1, \pi_2)]$$

Otherwise, the system outputs the interval $[\pi_1, \nu_2]$ flagged with a *contradiction warning*, as the intersection is *empty*. If $\nu_2 - \pi_1$ is small, then the contradiction level is small and the above interval can be accepted as an orientative result. If it is a large number, it means that different alternatives give totally different results so an error in the rulebase definition or a fault in one of the measurement devices providing the data must be suspected.

Post-processing. The results of the inference is a list of truth values of the disorders. Those truth values are to be interpreted as the "severities" (from incipient to severe) of the associated disorders.

The output of the expert system (interval of estimated severity) is translated onto a summarised statement. Each value of severity is mapped to a linguistic tag:

"negligible", "incipient", "medium", "severe"

defining an interval of severities for each tag partitioning the full [0,1] range. If both extreme severities of the conclusions have the same tag, then a conclusion in the form:

> Fault FAULTNAME is TAG (MINIMUM, MAXIMUM severity)

is extracted. Otherwise, the produced sentence is:

> Fault FAULTNAME severity might range from TAG(MIN severity) to TAG(MAX severity)

3 OIL ANALYSIS APPLICATION

Oil analysis is a key technology in predictive maintenance of industrial Diesel engines. Indeed, by determining the amount of wear particles, the composition of them, and other chemicals in the oil, a sensible set of rules can be cast to allow a reasonably accurate prediction of the oil condition and/or some likely engine malfunction. For instance, oil samples may be taken from metropolitan transport fleet vehicles in scheduled maintenance work, for subsequent analysis (in facilities such as the one in Figure 2); in this way, preventive actions would help reducing costs and avoiding passenger complaints due to breakdown while the vehicle is in service.

Expert systems based on binary logic have been developed for the application (Macián et al., 2000), but the use of a fuzzy logic inference engine is considered advantageous and it is being evaluated.

An application of the above ideas is under development at this moment. Let us discuss some issues on its development.

Figure 2: Scheduled maintenance facilities for metropolitan transport fleet vehicles.

Preprocessing. When acquiring information from a particular engine, some observations have the same meaning for all engines to be diagnosed. However, other ones need the use of historical data to generate "normalised" deviations taking into account statistical information for a particular engine *brand* or *model*, or a particular *unit* with special characteristics. In this way, the rulebase conception can be more general (applied without modification to a larger number of cases) if the data are suitably scaled and displaced prior to inference or, equivalently, fuzzy sets are modified according to the particular engine being diagnosed.

In some measurements, the procedure involves normalising the deviation from the mean expressing it in variance units, and generating an adimensional quantity. The statistical data are calculated from a database of previous analysis, classified by brand (manufacturer) and model (and also from historical records from the same engine).

Other variables are transformed to "engineering" units, having a more suitable interpretation than the raw readings (for instance, oil viscosities are expressed as a percentage of a reference value from fresh oil characteristics, instead of the centiStoke measurement).

Also, in order to consider real engine behaviour, oil consumption and fresh oil additions are considered leading to obtain a *compensated* wear element concentration more representative of engine status (Macián et al., 1999).

Knowledge base. At this moment, the team is in process of acquiring and refining a knowledge base with diagnosis rules.

The basics of the knowledge to be incorporated on the expert system lie on the following facts.

System is focused on automotive engines diagnosis (trucks, buses and general and road construction equipment), and so, the different parameters to measure are (Macián et al., 1999):

- Oil properties: viscosity, Total Base Number (TBN) and detergency.
- Oil contaminants: Insoluble compounds, fuel dilution, soot, ingested dust (silicon), water and glycol.
- Metallic elements: iron, copper, lead, chrome, aluminum, tin, nickel, sodium and boron.

Other measurements could be performed upon the oil sample, but with these basic parameters a good diagnosis can be achieved. Systems developed for other types of engines could choose other parameters taking into account the particularities of these types of engines.

Let us consider, as an example, the kind of knowledge involved on the dust contamination detection.

Silica and silicates are present at high concentrations in natural soils and dusts. It is for this reason that silicon is used as the most important indicator of dust entry into an engine. There have been several studies done on the causes of premature wear in components and results vary from study to study but one thing is clear: external contamination of lube oil by silicon is a major cause of accelerated wear.

Particles of airborne dust vary in size, shape and abrasive properties. In an engine, the ingress of atmospheric dust takes place primarily through the air intake. Those dust particles, not retained by filters, and similar in size to the oil film clearance in the main lubricated parts of the engine do the maximum damage. Once the dust particle has entered an oil film, it establishes a direct link between the two surfaces, nullifying the effect of the oil film; in this way, the immediate consequence is a "scratching" of the surface as the particle is dragged and rolled across the surfaces.

The second and potentially more serious problem is that once the dust particle is introduced in between the two surfaces, it changes the loading of the surface from an even distribution to a load concentrated on the particle with a huge increase in pressure at this point. The increase in pressure causes a deflection of the surface, which will eventually result in metal fatigue and the surface breaking up. As soon as a dust entry problem occurs there is an increase in the silicon concentration into the oil and an acceleration of the wear pattern.

As long as the oil samples are being taken at regular intervals in the correct manner, the dust entry will be detected at a very early stage. If an effective corrective action is taken, the life span of the component will be significantly increased, reducing maintenance costs.

A summary of the engineering knowledge related to diagnosis of silicon contamination (dust ingestion) is described below:

- If normal wear patterns combine with high silicon readings in oil analysis, there are three possibilities: a silicone sealant, grease or additive is in use mixing with engine oil, an accidental contamination of the sample has occurred or dust ingestion is in the first stage and no others wear patterns are present yet (too lucky situation). Action recommend to the maintenance technicians must be to check if an additive, grease or sealant has been used recently on the engine and make sure that the correct sampling technique was used. An inspection of the air admission system on the engine will be necessary if previous action results negative.

- Increased engine top-end wear (iron, chromium, or aluminium concentration rises up). This increased engine top-end wear is caused by airborne dust that has been drawn into the combustion chamber being forced down between the ring, piston and cylinder. Dust origin is caused by a defective air cleaner or a damaged induction system. Actions to be taken by maintenance technicians are inspect the air filter element thoroughly, and check its seals and support frame for damage and distortion and check too the pleats for damage. If there is any doubt about a filter element, it should always be changed. If the leak was found, it is necessary to repair the leak and determine the condition of the engine by checking compression or blowby.

- Increased engine bottom-end wear (lead, tin or copper concentration rises up). This situation indicates that dirt is basically getting into the lube oil directly and not past the pistons and rings. The likely sources are: leaking seals, defective breather, damaged seal on oil filler cap or dipstick, or dirty storage containers and/or top-up containers. Recommended action to be taken by technicians must be that any dust that is in the oil will be pumped through the oil filter before entering the bearings. Therefore, the first step is to examine the oil filter looking for dust contamination or bearing material. If excessive dust is found, thoroughly check all seals and breathers, etc. Check the oil storage containers and top-up containers for finding the source of contamination.

In the syntax of implemented prototype tool, the rules are:

```
CONTS1 SI NOT NORMAL END
WEAR_1 IF
 FE NOT NORMAL or CR NOT NORMAL
 or AL NOT NORMAL
WEAR_2 IF
 PB NOT NORMAL or SN NOT NORMAL
  or CU NOT NORMAL

 CONTS2 IF CONTS1 and  WEAR_1
 CONTS3 IF CONTS1 and  WEAR_2
 SILICON_CONTAMINATION IF CONTS1
 DUST_INGESTION IF CONTS2 or CONTS3
```

As another example, water problems can be divided into two different sources: an external water contamination or refrigerant leakage, in each case a different behaviour is presented. Additionally, water can evaporated and not to be present in oil. For this case, other fingerprints, that remain in oil when water evaporated must be found, such as: sodium (NA), boron (BO), its ratio (NABO) or glycol (GLIC) (its absence would fire rule CONTW3 at most 70%).

Finally, to take into account a specific situation such as a refrigerant leakage with greats amounts of copper from tube wear caused by water surface corrosion, an specific rule is defined too (CONTW4). In the syntax of the implemented tool, the rules are:

```
CONTW1  WATER NOT NORMAL END
CONTW2  GLIC NOT NORMAL END
CONTW4 CU VERYHIGH END

CONTW3
  NA NOT NORMAL
  BO NOT NORMAL
  NABO ABNORMAL
  GLYC NOT NORMAL 0.7
END

WATERFAULT IF
 CONTW1 or CONTW2 or CONTW3 or CONTW4

DISCARD WEAR IF CONTW4
```

So, based on the ideas from the above rules, a full rulebase is being built at this moment.

4 CONCLUSIONS

This paper presents a prototype fuzzy expert system for oil diagnosis. The flexibility of this system is greater than those of binary rules, allowing for gradation of diagnosis. Also, refinements such as interval-valued memberships, membership transformations, exceptions (discarding), unions and alternatives are included. In this way, the diagnostic capabilities and the readability of the rule base improve substantially.

The oil analysis application in consideration provides a quite complete set of measurements from which expert rules can be asserted with a reasonable reliability. However, for suitable diagnosis on a particular engine, a pre-processing module is essential: this module incorporates records of similar engines (same brand, model, history of the one being monitored, fresh oil characteristics, analytical calculations, *etc.*) so that the fuzzy set definitions are adapted for each case.

The full system is, at this moment, in development and prototype testing stage (comparing with human experts' conclusions and those from preexisting *ad hoc* software based on binary logic), but its preliminary results are promising.

ACKNOWLEDGEMENTS

The research work presented in this paper is financed by interdisciplinary coordinated projects FEDER/MEC DPI-2004-07332-C02-01 and DPI-2004-07332-C02-02 (Spanish goverment and European Union funds).

REFERENCES

Atanasov, K. (1986). Intuitionistic fuzzy sets. *Fuzzy Sets and Systems*, 20:87–96.

Carrasco, E. and *et. al.*, J. R. (2004). Diagnosis of acidification states in an anaerobic wastewater treatment plant using a fuzzy-based expert system. *Control Engineering Practice*, 12(1):59–64.

Cayrac, D., Dubois, D., and Prade, H. (1996). Handling uncertainty with possibility theory and fuzzy sets in a satellite fault diagnosis application. *IEEE Trans. on Fuzzy Systems*, 4(3):251–269.

Chang, S.-Y. and Chang, C.-T. (2003). A fuzzy-logic based fault diagnosis strategy for process control loops. *Chemical Engineering Science*, 58(15):3395–3411.

Chen, H., Fuller, S., C., F., and Hersh, W., editors (2005). *Medical Informatics : Knowledge Management and Data Mining in Biomedicine*. Springer.

Chiang, L., Russell, E., and Braatz, R. (2001). *Fault Detection and Diagnosis in Industrial Systems*. Springer-Verlag.

Dubois, D. and Prade, H. (1997). The three semantics of fuzzy sets. *Fuzzy Sets and Systems*, 90(2):142–150.

Entemann, C. (2000). A fuzzy logic with interval truth values. *Fuzzy Sets and Systems*, 113:161–183.

Isermann, R. and Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5):709–719.

Lindley, D. (1987). The probability approach to the treatment of uncertainty in artificial intelligence and expert systems. *Statistical Science*, 2(1):17–24.

Macián, V., Lerma, M., and Tormos, B. (1999). Oil analysis evaluation for an engines fault diagnosis system. *SAE Papers*, 1999-01-1515.

Macián, V., Tormos, B., and Lerma, M. (2000). Knowledge based systems for predictive maintenance of diesel engines. In *Proc. Euromaintenance Conf.*, volume 1, pages 49–54. Swedish Maintenance Society-ENFMS.

Russell, E., Chiang, L., and Braatz, R. (2000). *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes*. Springer.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence: a modern approach*. Prentice-Hall, 2nd edition.

Sala, A. and Albertos, P. (2001). Inference error minimisation: Fuzzy modelling of ambiguous functions. *Fuzzy Sets and Systems*, 121(1):95–111.

Szmidt, E. and Kacprzyk, J. (2003). An intuitionistic fuzzy set based approach to intelligent data analysis: an application to medical diagnosis. In Ajit, A., Jain, L., and Kacprzyk, J., editors, *Recent Advances in Intelligent Paradigms and Applications*, chapter 3, pages 57–70. Physica-Verlag (Springer), Heidelberg.

Wang, K., editor (2003). *Intelligent Condition Monitoring and Diagnosis System*. IOS Press, Inc.

Weber, S. (1983). A general concept of fuzzy connectives, negations and implications based on T-norms and T-conorms. *Fuzzy Sets and Systems*, 11:115–134.

DERIVING BEHAVIOR FROM GOAL STRUCTURE FOR THE INTELLIGENT CONTROL OF PHYSICAL SYSTEMS

Richard Dapoigny, Patrick Barlatier, Eric Benoit, Laurent Foulloy
LISTIC/ESIA - University of Savoie (France)
BP 806 74016 Annecy cedex
Email: {richard.dapoigny,pbarlatier,eric.benoit,laurent.foulloy}@univ-savoie.fr

Keywords: Knowledge-based systems, teleological model, Formal Concept Analysis, Event Calculus.

Abstract: Given a physical system described by a structural decomposition together with additional constraints, a major task in Artificial Intelligence concerns the automatic identification of the system behavior. We will show in the present paper how concepts and techniques from different AI disciplines help solve this task in the case of the intelligent control of engineering systems. Following generative approaches grounded in Qualitative Physics, we derive behavioral specifications from structural and equational information input by the user in the context of the intelligent control of physical systems. The behavioral specifications stem from a teleological representation based on goal structures which are composed of three primitive concepts, i.e. physical entities, physical roles and actions. An ontological representation of goals extracted from user inputs facilitates both local and distributed reasoning. The causal reasoning process generates inferences of possible behaviors from the ontological representation of intended goals. This process relies on an Event Calculus approach. An application example focussing on the control of an irrigation channel illustrates the behavioral identification process.

1 INTRODUCTION

One of the most interesting and challenging tasks of Artificial Intelligence is to derive the behavior of a system from its components and additional information or constraints. Reasoning about physical systems constitutes an important and active area of research of Artificial Intelligence, also known as Qualitative Reasoning (QR). In QR, most works have focussed on the representation and composition of models to describe physical systems either with a component-based approach (de Kleer and Brown, 1984) or with a process-based approach (Forbus, 1984; Falkenhainer and Forbus, 1991). Major areas of investigation are i) the simulation of physical systems to predict their behavior ii) given a domain theory, a structural description of the system and a query about the system's behavior, the composition of a model answering the query.

Another major domain of reasoning which involves structural modelling and behavioral analysis is the Software Engineering (SE). A significant part of work in software engineering is dedicated to temporal logics (McDermott, 1982; Manna and Pnueli, 1992; Ma and Knight, 1996; Galton, 1987; Freksa, 1992) with extensions for specifying concurrent systems (Barringer, 1986; Chen and de Giacomo, 1999). These logics form the basis of behavior analysis relying on concepts such as goals, actions and event structures.

In this paper, we are concerned with the control of physical activity by means of software engineering mechanisms. Let us consider Intelligent Systems interacting with a physical system. It requires at least AI domains such as QR, for the abstraction of physical mechanisms and SE, for behavioral analysis of software components. This analysis has a great impact both on the processing of variables related to physical quantities and on their control. We introduce the notion of Intelligent Control System (ICS) composed of a computing unit (e.g., PC, workstation, microcontroller card, DSP-based system, ...) and sensor(s) and/or actuator(s) unit(s). Distributed ICS exchange information through networks using appropriate protocols (e.g., TCP/IP/Ethernet or dedicated field buses such as CAN, LonWorks, ...). Inside an ICS, two information flows co-exist, information from/to other ICS via network ports and information from/to the physical system via I/O ports. The control of physical systems with ICS requires reasoning capabilities extracted both from QR techniques and SE concepts such as events and actions. From the outside, an ICS

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 51–58.

can be seen as an intelligent device offering a set of services. Each of theses services are designed to achieve a given goal, provided that some sequence of atomic goals is achieved. More precisely, we tackle the following problem.

Given:

- A scenario description including a physical hierarchical structure together with a set of physical equations relating physical variables.
- A modelling theory (derived from the General System Theory) whose instantiation on the given domain together with a set of rules will produce a local domain theory.
- A goal (i.e., a service) request concerning the local domain.

Produce:

- during the design step, a goal hierarchy.
- during the design step, an action hierarchy which traduces the way of achievement of each goal.
- at run-time, the most relevant behavior depending upon constraints.

This problem concerns major applications such as the control of industrial processes, automotive systems, automatic planning for control of physical systems and measurements, robotics, ... Notice that in the present model, the structural description of the physical system may be replaced in unknown environments by a learning phase based on classical techniques such as neural networks, genetic algorithms, fuzzy logic, etc.

2 FOUNDATIONS FOR THE MODELLING OF CONTROL SYSTEMS RELATED TO ENGINEERING PROCESSES

2.1 The Structural Model

When designing or analyzing a system, the particular model formalisms that are used depend on the objectives of the modelling. In the engineering domain, the formalisms commonly adopted are functional, behavioral and structural (Dooley et al., 1998). The structural representation is an essential component of the model involving physical systems. Most variables of the control process are physical variables, that is, they are an abstraction of the physical mechanism which is related with each ICS. We consider the semantic representation of control variables as a tuple including the physical role and the physical (i.e., spatial) entity in which the physical role is evaluated. These tuples will be referred to as Physical Contexts in the following. Physical variables are a subset of control variables and their physical role is in fact the so-called physical quantity defined in standard ontologies (Gruber and Olsen, 1994). In a first step, a part-of hierarchy of physical entities can be easily sketched. In a second step, the physical behavior of physical entities is described by expressing the way these entities interact. The physical interactions are the result of energetic physical processes that occur in physical entities. Whatever two entities are able to exchange energy, they are said to be connected. Therefore, the mereology is extended with a topology where connections highlight the energy paths between physical entities. This approach extracts in a local database, energy paths stretching between ICS in the physical environment.

2.2 The Teleological Model of Actions: The Goal Structure

In the teleological reasoning, the structure and behavior of a physical system are related to its goals. In other words, purposes are ascribed to each component of the system and to achieve a global goal, one must describe how each function of the systems' parts can be connected. Moreover, since diagnosis is an essential part of models describing physical processes, most works relative to functional reasoning in the last decade have incorporated teleological knowledge in their model (Lind, 1994; Larsson, 1996; Chittaro et al., 1993). Finally, Qualitative Reasoning based on a teleological approach appears to be a useful component for planning systems involving the physical world (de Coste, 1994). Therefore, we adopt the teleological model where goals describe the purposes of the system, and function(s)[1] represent the way of achievement of an intended goal. This approach is similar to that of some authors (Kitamura et al., 2002) which claim that base-functions represent function types from the view point of their goals' achievement.

The concept of goal is central for behavior analysis in the control of physical systems. For example, in failure analysis, when a behavioral change affects one of the system goals, it means that a failure occurred (the effect is expressed in terms of the goals that have not been achieved). Basically, the goal representation must facilitate the construction of knowledge databases and allows to classify goals and subgoals relatively to the designers' intents. The goal modelling requires i) to describe goal representation (i.e., data structures), ii) to define how these concepts are related.

[1] i.e., computing function

A goal structure must incorporate some possible actions (at least one) in order to fulfill the intended goal (Hertzberg and Thiebaux, 1994; Lifschitz, 1993). Representation of intended goals as "*to do action*" has been proposed by several researchers (Lind, 1994; Umeda and al., 1996; Kmenta et al., 1999) but neither proposes a formal structure on which reasoning can be based. Therefore, we extend that textual definition by introducing a goal structure with information relative to the physical system. Two types of atomic goals are defined, a goal type (universal) which relate an action verb, a physical quantity[2] with its arity and an entity type, and a goal token (particular) by particularizing the physical entity type item of the goal type. In such a way, the goal modelling defines the terms that correspond to actions expressing the intension with the terms that are objects of the actions. As it incorporates an action verb, the basic goal definition proposed here can be seen as a generalization of the action concept closed to the action (or event) types defined in (Galton and Augusto, 2000).

Unlike general framework where goals cannot be formalized and relationships among them cannot be semantically captured, the present framework restricted to engineering physical entities makes it possible to describe a hierarchical structure (i.e., a mereology) of goals where the bottom level is composed of atomic goals. Complex goals can be expressed as a mereological fusion of atomic sub-goals. One of the major benefits of mereological framework is that it allows for different abstraction levels to appear in the same model.

2.3 The Behavioral Model

Behavior models play a central role in systems specifications and control. In order to specify the behavior of a system, different approaches are possible. From the software engineering point of view, a reference model specifying the behavior of distributed systems (ISO/IEC, 1996) introduces a minimal set of basic modelling concepts which are behavior, action, time constraints and states. The popular approaches to behavioral specification languages are based on either states or actions (Abadi and Lamport, 1993). In the state-based approach, the behavior of a system is viewed as a sequence of states, where each state is an assignment of values to some set of components. Alternatively, an action-based approach views a behavior as a sequence of actions. Selecting behavior, goals (i.e., extended actions) time constraints and time-variant properties (i.e., fluents), we adopt the logical formalism of the Event Calculus (EC). This formalism presents some advantages well-suited to our purposes, in particular its ability to represent actions with duration (which is required to describe compound actions) and to assimilate narrative description of events with an adjustment of the actions' effects in a dynamic way.

Definition 1 *A behavior is defined as a collection of extended actions occurring according to a set of constraints, known as pre-conditions.*

Therefore, two concepts are highlighted, action types (goals) and constraints. As the physical system include artifacts, interaction with the physical system involves actions with these artifacts. This assertion reinforces the choice of a goal structure relating an action with its physical entity. As a consequence, behaviors are the result of teleological interpretation of causal relations among atomic goals.

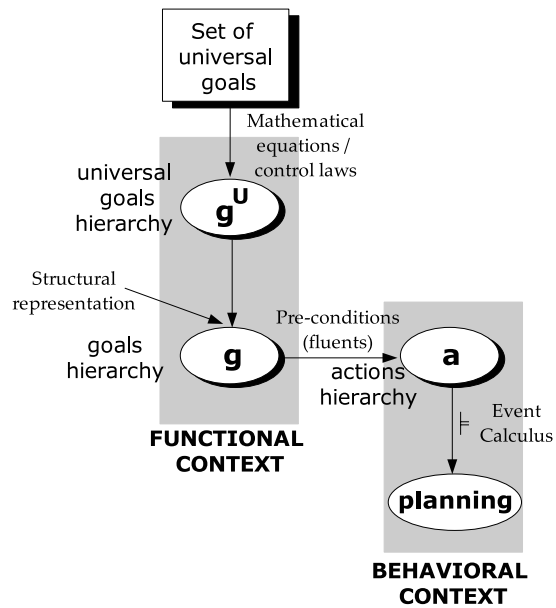


Figure 1: The local model for Intelligent control.

As suggested in (Barwise and Seligman, 1997), distribution of knowledge presupposes a system of classification. Alternatively, conception and analysis of knowledge relies on powerful techniques such as Formal Concept Analysis. The unification of these two related theories seems a promising candidate to build the foundations of a theory of distributed conceptual structures (Kent, 2003). As a consequence each of the previous sub-models can be related to classifications through formal contexts as described in Figure 1. The first part produces a goal hierarchy according to a spatial classification where types are goal types and tokens are spatial instances of goals, i.e., goals related to a spatial localization. Constraints on goal types are given through physical equations or/and control

[2] we generalize this definition with "physical role" concerning sorts which are not physical

laws relating physical roles. Then, the goal hierarchy is mapped onto programming functions through fluents constraints. These constraints correspond to the well-known pre-conditions in STRIPS-like planning. If several preconditions are defined, then several ways of achievement exist for a given goal, each of them corresponding to an event (or action) type. Therefore, a second classification is defined with events as types and events occurrences as tokens. This classification is a temporal one where the constraints are given by the Event Calculus formalism through fluents, axioms and a set of rules. The whole design process begins with the introduction of a set of universal goals and produces through a refinement process, a planning for a given control system in a given environment, i.e. spatial and temporal instances of general information.

3 THE TARGET APPLICATION

The real-world example concerns an open-channel hydraulic system which is controlled with (at least) two ICS, as shown in Figure 2. The control nodes are connected with a fieldbus (CAN network). Each active ICS_i, in the open-channel irrigation channel is located near a water gate and performs two pressure measurements from a Pitot tube (resp. in $SFArea_i$ and $DFArea_i$). In addition, it is able to react accordingly and to modify the gate position with the help of a brushless motor. Pairs of goal-program functions are the basic elements on which knowledge representation is built. While the basic functions are extracted from libraries, the goal/subgoal representation requires a particular attention. To each subgoal, one or several dedicated software functions can be either extracted from libraries or defined by the user. Goals and functioning modes are user-defined. All functions handle variables whose semantic contents is extracted from the structural mereology.

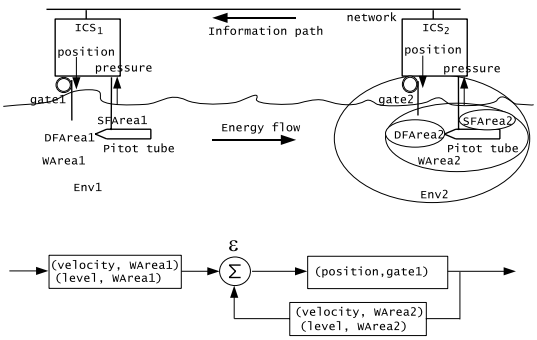


Figure 2: The hydraulic control system with two Intelligent control nodes.

4 THE CONCEPTUAL GOAL HIERARCHY

The Formal Concept Analysis produces a conceptual hierarchy of the domain by exploring all possible formal concepts for which relationships between properties and objects hold. The resulting concept lattice, also known as Galois Lattice, can be considered as a semantic net providing both a conceptual hierarchy of objects and a representation of possible implications between properties. A formal context C is described by the triple $C = (O, A, I)$, where O is a nonempty finite set of objects, A is a nonempty finite set of attributes and $I \subseteq O \times A$ is a binary relation which holds between objects and attributes. A formal concept (X, Y) is a pair which belongs to the formal context C if $X \subseteq O$, $Y \subseteq A$, $X = Y^I$ and $Y = X^I$. X and Y are respectively called the extent and the intent of the formal concept (X, Y). The ordered set $(\mathcal{B}(C), \leq)$ is a complete lattice called the concept lattice of the formal context (C).

Definition 2 *Given R, a finite set of physical roles and ϕ, the finite set of physical entities, a Physical Context (PC) is a tuple: $\theta = (r, \mu(r), \varphi_1, \varphi_2, ...\varphi_{\mu(r)})$, where $r \in R$, denotes its physical role (e.g., a physical quantity), $\mu : R \rightarrow Nat$, a function assigning to each role its arity (i.e., the number of physical entities related to a given role) and $\{\varphi_1, ...\varphi_{\mu(r)}\} \subseteq \phi$, a set of entities describing the spatial locations where the role has to be taken.*

Definition 3 *Given Φ the finite set of physical entities types, a goal type is a pair (A, Ξ), where A is an action symbol and Ξ a non-empty set of tuples $\xi = (r, \mu(r), \phi_1, \phi_2, ...\phi_{\mu(r)})$ where $\{\phi_1, ...\phi_{\mu(r)}\} \subseteq \Phi$, a set of entities types describing the spatial locations where the role r has to be taken.*

$$\gamma \stackrel{def}{=} (a, \Xi) \tag{1}$$

Definition 4 *A goal token is a pair (A, Θ), where A is an action symbol and Θ a non-empty set of tuples $\theta = (r, \mu(r), \varphi_1, \varphi_2, ...\varphi_{\mu(r)})$.*

$$g \stackrel{def}{=} (a, \Theta) \tag{2}$$

The hydraulic control system requires the following list of basic goals[3] :

$$\begin{aligned}
g_1 &= (to_acquire, \{(pressure, 1, SFArea1)\}) \\
g_2 &= (to_acquire, \{(pressure, 1, DFArea1)\}) \\
g_3 &= (to_compute, \{(velocity, 1, WaterArea1)\}) \\
g_4 &= (to_compute, \{(level, 1, WaterArea1)\}) \\
g_5 &= (to_send, \{(velocity, 1, WaterArea1), \\
&\quad (level, 1, WaterArea1)\}) \\
g_6 &= (to_receive, \{(velocity, 1, ExtEntity), \\
&\quad (level, 1, ExtEntity)\})
\end{aligned}$$

[3] these goals are not concepts

$g_7 = (to\_compute, \{(level, 2, WaterArea1, ExtEntity)$
$g_8 = (to\_compute, \{(offset, 1, Gate1)\})$
$g_9 = (to\_receive, \{(offset, 1, Gate1)\})$
$g_{10} = (to\_move, \{(position, 1, Gate1)\})$

A close connection between FCA and mereology can be established by focusing on their basic topics, i.e., concept decomposition-aggregation and concept relationships. FCA helps to build ontologies as a learning technique (Cimiano et al., 2004) and we extend this work by specifying the ontology with a part-of hierarchy. The goal hierarchy is derived from the subsumption hierarchy of conceptual scales where the many-level architecture of conceptual scales (Stumme, 1999) is extended taking into consideration the mereological nature of the extents. Higher level scales which relates scales on a higher level of abstraction provide information about hierarchy. Considering the atomic goals, the compound goals corresponding to the user intents, the ontological nature of the extents (i.e., the physical entities) and some basic assumptions, one can automatically produce the relevant instrument functional context. This context is required to produce the final concept lattice from which the functional mereology will be extracted.

As suggested in (Stumme, 1999), the set of subgoals is extended with hierarchical conceptual scales such as the intent includes atomic and compound goals and the ICS scale (highest level). Higher level scales define a partially ordered set. The formal context is filled in a two-stages process. Then, we derive some rules from the structural mereology $\mathcal{S}$ which concerns the physical entities. To overcome difficulties about the conceptual equivalence between sets and mereological individuals, we make the assumption that a mereological structure can be reproduced within sets provided we exclude the empty set. Therefore, a set can be seen as an abstract individual which represents a class[4]. The part-of relation can be described as a conceptual scale which holds between the objects (i.e., extensions) related to the mereological individuals. The context considers goal achievement predicates as formal objects, goals and compound goals as formal attributes. First, the sub-context between basic goals is derived from qualitative reasoning on PC tuples within each physical (or control) equation. Then this sub-context is extended with conceptual scales corresponding to goal requests specified by the user (see table 1). For the hydraulic system for example, we plan three services with the respective goals:

$G_1 = (to\_measure, \{(speed, 1, WaterArea1), (level, 1, WaterArea1)\})$
$G_2 = (to\_control, \{(speed, 1, WaterArea1)\})$
$G_3 = (to\_manuallyMove, \{(position, 1, Gate1)\})$

Then, the concept lattice is transformed in a partial order by some elementary rules. First, for each node the concept is labelled with the intent of the lattice node (Cimiano et al., 2003). In a second step, overlaps are highlighted and the previous ordering is reduced based on simplification rules (Dapoigny et al., 2005). In a third step, we reduce the labelling (Ganter and Wille, 1999), providing that each intent is entered once in the lattice. Finally, the bottom element is removed. These rules applied on the raw lattice (Figure 3) result in the goal hierarchy of Figure 4.

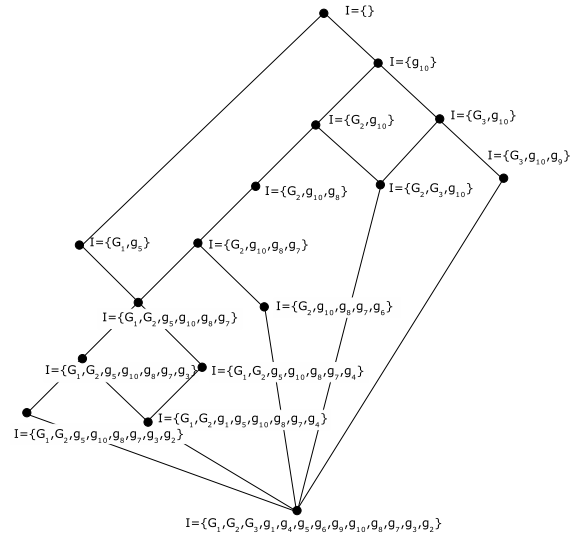


Figure 3: The goal lattice.

5 BEHAVIOR REPRESENTATION

While the EC supports deductive, inductive and abductive reasoning, the latter is of particular interest for our purpose. Given an ontological description based on possible causal behaviors, dynamical constraints can be translated in EC axioms. The EC axioms provide a partial temporal order from ontologies inferred with mereological logic from user-defined goals and SP pairs. Moreover, the abductive implementation of the EC is proved to be sound and complete (Russo et al., 2001). To solve the frame problem, formulae are derived from the circumscription of the EC representation.

Definition 5 *Let G be a goal, let Σ be a domain description, let Δ_0 be an initial situation, let Ω be a*

[4] A class is simply one or more individuals

Table 1: Functional context for the open-channel irrigation canal.

$\mathcal{F}$	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	G_1	G_2	G_3
$Achieved(g_1)$	x		x	x	x		x	x		x	x	x	
$Achieved(g_2)$		x	x		x		x	x		x	x	x	
$Achieved(g_3)$			x		x		x	x		x	x	x	
$Achieved(g_4)$				x	x		x	x		x	x	x	
$Achieved(g_5)$					x						x		
$Achieved(g_6)$						x	x	x		x		x	
$Achieved(g_7)$							x	x		x		x	
$Achieved(g_8)$								x		x		x	
$Achieved(g_9)$									x	x			x
$Achieved(g_{10})$										x		x	x

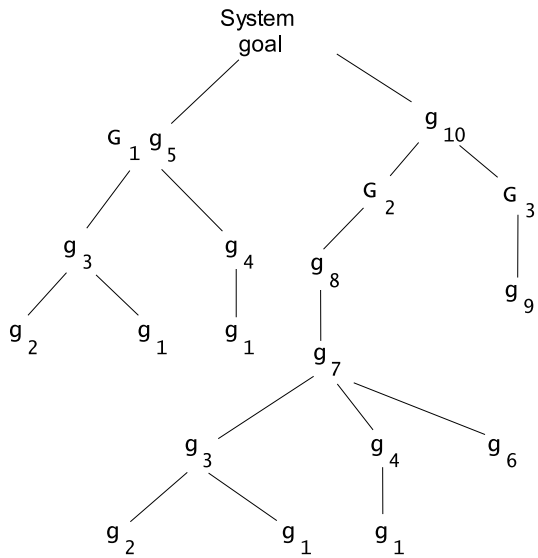


Figure 4: The goal part-of hierarchy.

conjunction of a pair of uniqueness-of-names axioms for the actions (goal) and fluents (Achieved(goal)) mentioned in Σ, and let Ψ be a finite conjunction of state constraints. A plan for G is a narrative Δ such that,

$$CIRC[\Sigma; Initiates, Terminates, Releases] \wedge CIRC[\Delta_0 \wedge \Delta; Happens] \wedge \Psi \wedge EC \wedge \Omega \models G \quad (3)$$

In this work, we use the version presented in (Shanahan, 1997) which consists in a set of time points, a set of time-variant properties (i.e., fluents) and a set of event types. Each event type is in fact an action type which at least requires an action verb, therefore we associate the extended actions (EA) to each operational goals (i.e., the triple action verb, physical role and physical entity). Under the assumption where a unique computing function is related to a single goal, domain equations are simple. For more complex situations, several computing functions (the actions of EC) can be related to a single goal provided that a set of fluents (pre-conditions) selects the relevant association in a given situation. The circumscriptive condition is consistent if the domain description does not allow a fluent to be both initiated and terminated at the same time. EA in the event calculus are considered as first class objects which can appear as predicates arguments. The conjunction of $Initiate$, $Terminates$ and $Releases$ formulae describe the effects of EA and correspond to the domain description. The finite conjunction of state constraints Ψ expresses indirect effects of potential actions. These constraints are available implicitly through the goal mereology since its description is deduced from qualitative equations where a complex goal achievement requires physical constraints to be satisfied. Mereological individuals from a given level and their adjacent lower ones give rise to a morphism between $Part-of$ relations and state constraints. From this assumption, state equations expressing physical and computational causality can be derived in event calculus (constraints on what combinations of fluents may hold in the same time). The uniqueness of EA names, i.e. $to\_Achieve(goal)$ and fluents names, i.e. $Achieved(goal)$ is a logical consequence of the uniqueness of goal description in the mereological model. Taking the example of the complex goal G_2, state equations are defined as follows:

$HoldsAt(g10, T) \leftarrow HoldsAt(g8, T).$
$HoldsAt(g8, T) \leftarrow HoldsAt(g7, T).$
$HoldsAt(g7, T) \leftarrow HoldsAt(g3, T), HoldsAt(g4, T), HoldsAt(g6, T).$
$HoldsAt(g4, T) \leftarrow HoldsAt(g1, T).$

$HoldsAt(g3,T) \leftarrow HoldsAt(g2,T)$, $HoldsAt(g1,T)$.

together with domain equations:

$Initiates(achieved(g8),g8,T) \leftarrow HoldsAt(g7,T)$.
$Initiates(achieved(g7),g7,T) \leftarrow HoldsAt(g3,T)$, $HoldsAt(g4,T), HoldsAt(g6,T)$.
$Initiates(achieved(g4),g4,T) \leftarrow HoldsAt(g1,T)]$.
$Initiates(achieved(g3),g3,T) \leftarrow HoldsAt(g2,T)$, $HoldsAt(g1,T)]$.
$Initiates(achieved(g2),g2,T)$.
$Initiates(achieved(g1),g1,T)$.
$Initiates(achieved(g6),g6,T)$.

Applying abductive logic for theorem proving, we get the plan Δ:

$Happens(achieved(g6),t7,t7)$,
$Happens(achieved(g4),t6,t6)$,
$Happens(achieved(g1),t5,t5)$,
$Happens(achieved(g2),t4,t4)$,
$Happens(achieved(g3),t3,t3)$,
$Happens(achieved(g7),t2,t2)$,
$Happens(achieved(g8),t1,t1)$.
$before(t7,t2), before(t5,t6), before(t6,t2)$,
$before(t5,t3), before(t4,t3), before(t3,t2)$,
$before(t2,t1), before(t1,t)$.

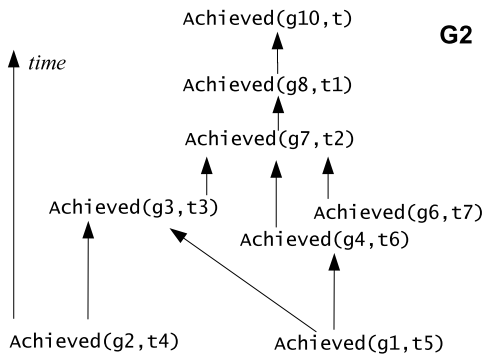


Figure 5: The temporal hierarchy.

This plan is sketched at figure 5. Initially an empty plan is presented with a goal G in the form of a $HoldAt$ formulae. The resulting plan must respect the causal hierarchy obtained in section 2.

6 RELATED WORK

Goal modelling is obviously investigated in requirements engineering. Modelling goals for engineering processes is a complex task. In (Rolland et al., 1998) goals are represented by verbs with parameters, each of them playing a special role such as target entities affected by the goal, resources needed for the goal achievement, etc. Centered on the KAOS method, (El-Maddah and Maibaum, 2003) used conditional assignments based on the application's variables in goal-oriented process control systems design with the B method. A tool translates the goal model into B specifications where the behavior is state-based. In this method no reasoning is performed at the system level due to the lack of semantic content for variables. For more general frameworks, (Giorgini et al., 2002) describes a logic of goals based on their relationship types, but goals are only represented with a label, and the reasoning is elicited from their relations only.

7 CONCLUSIONS

This work is part of an ongoing attempt to develop an automaton able to derive executable program for the intelligent control of physical systems. From a user-defined description of the context (i.e., the structural description of the physical system together with the control system which operates on it) and an initial goal request, the system architecture consists of two layered control modules to provide a response to this request. The first layer extracts a hierarchical functional model centered on the goal concept. This model is mapped through fluents on the hierarchy of action types. The second layer constructs a partial-order temporal hierarchy relating grounded actions. Unlike classical AND/OR goal decomposition which does not clearly distinguish dependency types between different abstraction levels, the part-of hierarchy is able to extract potential relevant dependencies (such as goal g_{10} in 4). The major benefits of Artificial Intelligence in this context turns out to reduce the design process, to generate automatic sound planning and to allow dynamic control at runtime through a reasoning process between distributed intelligent nodes. This last topic (under development) is centered on information flow methodology and conceptual structures.

REFERENCES

Abadi, M. and Lamport, L. (1993). Composing specifications. *ACM Transactions on Programming Languages and Systems*, 15(1):73–132.

Barringer, H. (1986). Using temporal logic in the compositional specification of concurrent systems. Technical Report UMCS-86-10-1, Dpt of Computer Science Univ. of Manchester.

Barwise, J. and Seligman, J. (1997). *Information flow - The logic of distributed systems*, volume 44 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press.

Chen, X. and de Giacomo, G. (1999). Reasoning about nondeterministic and concurrent actions: a process algebra approach. *Artificial Intelligence*, 107:63–98.

Chittaro, L., Guida, G., Tasso, C., and Toppano, E. (1993). Functional and teleological knowledge in the multimodelling approach for reasoning about physical sytems: a case study in diagnosis. *IEEE Transactions on Systems Man and Cybernetics*, 23(6):1718–1751.

Cimiano, P., Hotho, A., Stumme, G., and Tane, J. (2004). Conceptual knowledge processing with formal concept analysis and ontologies. In *ICFCA*, number 2961 in LNAI, pages 189–207. Springer.

Cimiano, P., Staab, S., and Tane, J. (2003). Deriving concept hierarchies from text by smooth formal concept analysis. In *Procs. of the GI Workshop LLWA*.

Dapoigny, R., Barlatier, P., Benoit, E., and Foulloy, L. (2005). Formal goal generation for intelligent control systems. In *18th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems*, number 3533 in LNCS. Springer.

de Coste, D. (1994). Goal-directed qualitative reasoning with partial states. Technical Report 57, Institute for the Learning Sciences, Northwestern University (Evanston).

de Kleer, J. and Brown, J. (1984). A qualitative physics based on confluences. *Artificial Intelligence*, 24:7–83.

Dooley, K., Skilton, P., and Anderson, J. (1998). Process knowledge bases: Facilitating reasoning through cause and effect thinking. *Human Systems Management*, 17(4):281–298.

El-Maddah, I. and Maibaum, T. (2003). Goal-oriented requirements analysis for process control systems design. In *Procs. of the International Conference on formal methods and models for co-design*, pages 45–46. IEEE Comp. Society Press.

Falkenhainer, B. and Forbus, K. (1991). Compositional modeling: finding the right model for the job. *Artificial Intelligence*, 51:95–143.

Forbus, K. (1984). Qualitative process theory. *Artificial Intelligence*, 24:85–168.

Freksa, C. (1992). Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54:199–227.

Galton, A. (1987). *Temporal logics and their applications*. Academic Press.

Galton, A. and Augusto, J. (2000). Two approaches to event definition. In *Springer*, number 2453 in LNCS, pages 547–556.

Ganter, B. and Wille, R. (1999). *Formal concept analysis - mathematical foundations*. Springer.

Giorgini, P., Nicchiarelli, E., Mylopoulos, J., and Sebastiani, R. (2002). Reasoning with goal models. In *Procs. of the int. conf. on Conceptual Modeling*, number 2503 in LNCS, pages 167–181. Springer.

Gruber, G. and Olsen, G. (1994). An ontology for engineering mathematics. In Doyle, J., Torasso, P., and Sandewall, E., editors, *Fourth International Conference on Principles of Knowledge Representation and Reasoning*, pages 258–269. Morgan Kaufmann.

Hertzberg, J. and Thiebaux, S. (1994). Turning an action formalism into a planner: a case study. *Journal of Logic and Computation*, 4:617–654.

ISO/IEC (1996). *Open Distributed Processing - Basic Reference model - Part 2: Foundations*. ISO/IEC and ITU-T.

Kent, R. (2003). Distributed conceptual structures. In *in Procs. of the 6th Int. Workshop on Relational Methods in Computer Science*, number 2561 in LNCS. Springer.

Kitamura, Y., Sano, T., Namba, K., and Mizoguchi, R. (2002). A functional concept ontology and its application to automatic identification of functional structures. *Artificial Intelligence in Engineering*, 16(2):145–163.

Kmenta, S., Fitch, P., and Ishii, K. (1999). Advanced failure modes and effects anlysis of complex processes. In *Procs. of the ASME Design Engineering Technical Conferences*, Las Vegas (NE).

Larsson, J. (1996). Diagnosis based on explicit means-end models. *Artificial Intelligence*, 80:29–93.

Lifschitz, V. (1993). A theory of actions. In *Procs. of the tenth International Joint Conference on Artificial Intelligence*, pages 432–437. Morgan Kaufmann.

Lind, M. (1994). Modeling goals and functions of complex industrial plant. *Journal of Applied Artificial Intelligence*, 8:259–283.

Ma, J. and Knight, B. (1996). A reified temporal logic. *The Computer Journal*, 39(9):800–807.

Manna, Z. and Pnueli, A. (1992). *The temporal logic of reactive and concurrent systems (Specification)*. Springer-Verlag.

McDermott, D. (1982). A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6: 101–155.

Rolland, C., Souveyet, C., and Achour, C. B. (1998). Guiding goal modelling using scenarios. *IEEE Transactions on software engineering*, pages 1055–1071.

Russo, A., Miller, R., Nuseibeh, B., and Kramer, J. (2001). An abductive approach for analysing event-based requirements specifications. Technical Report DoC-2001/7, Dpt of Computing Imperial College, London.

Shanahan, M. (1997). *Solving the frame problem: A mathematical investigation of the common sense law of inertia*. MIT Press.

Stumme, G. (1999). Hierarchies of conceptual scales. In *Procs. of the Workshop on Knowledge Acquisition, Modeling and Managenent (KAW'99)*, volume 2, pages 78–95.

Umeda, Y. and al. (1996). Supporting conceptual design based on the function-behavior-state modeler. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 10(4):275–288.

EVOLUTIONARY COMPUTATION FOR DISCRETE AND CONTINUOUS TIME OPTIMAL CONTROL PROBLEMS

Yechiel Crispin
Department of Aerospace Engineering
Embry-Riddle University
Daytona Beach, FL 32114
crispinj@erau.edu

Keywords: Optimal Control, Rocket Dynamics, Goddard's Problem, Evolutionary Computation, Genetic Algorithms.

Abstract: Nonlinear discrete time and continuous time optimal control problems with terminal constraints are solved using a new evolutionary approach which seeks the control history directly by evolutionary computation. Unlike methods that use the first order necessary conditions to determine the optimum, the main advantage of the present method is that it does not require the development of a Hamiltonian formulation and consequently, it eliminates the requirement to solve the adjoint problem which usually leads to a difficult two-point boundary value problem. The method is verified on two benchmark problems. The first problem is the discrete time velocity direction programming problem with the effects of gravity, thrust and drag and a terminal constraint on the final vertical position. The second problem is a continuous time optimal control problem in rocket dynamics, the Goddard's problem. The solutions of both problems compared favorably with published results based on gradient methods.

1 INTRODUCTION

An optimal control problem consists of finding the time histories of the controls and the state variables such as to maximize an integral performance index over a finite period of time, subject to dynamical constraints in the form of a system of ordinary differential equations (Bryson, 1975). In a discrete-time optimal control problem, the time period is divided into a finite number of time intervals of equal duration ΔT. The controls are kept constant over each time interval. This results in a considerable simplification of the continuous time problem, since the ordinary differential equations can be reduced to difference equations and the integral performance index can be reduced to a finite sum over the discrete time counter (Bryson, 1999). In some problems, additional constraints may be prescribed on the final states of the system.

Modern methods for solving the optimal control problem are extensions of the classical methods of the calculus of variations (Fox, 1950). These methods are known as indirect methods and are based on the maximum principle of Pontryagin, which is a statement of the first order necessary conditions for optimality, and results in a two-point boundaryvalue problem (TPBVP) for the state and adjoint variables (Pontryagin et al., 1962).

It has been known, however, that the TPBVP is much more difficult to solve than the initial value problem (IVP). As a consequence, a second class of solutions, known as the direct method has evolved. For example, attempts have been made to recast the original dynamic optimization problem as a static optimization problem by direct transcription (Betts, 2001) or some other discretisation method, eventually reformulating the original problem as a nonlinear programming (NLP) problem. This is often achieved by parameterisation of the state variables or the controls, or both. The original differential equations or difference equations are reduced to algebraic equality constraints. A significant advantage of this method is that the Hamiltonian formulation is completely avoided, which can be advantageous to practicing engineers who have not been exposed to the theoretical framework of optimal control. However, there are some problems with this approach. First, it might result in a large scale NLP problem which might suffer from numerical stability and convergence problems and might require excessive computing time. Also, the parameterisation might introduce spurious local minima which are not present in the original problem.

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 59–69.

With the advent of computing power and the progress made in methods that are based on optimization analogies from nature, it became possible to achieve a remedy to some of the above mentioned disadvantages through the use of global methods of optimization. These include stochastic methods, such as simulated annealing (Laarhoven and Aarts, 1989), (Kirkpatrick and Vecchi, 1983) and evolutionary computation methods (Fogel, 1998), (Schwefel, 1995) such as genetic algorithms (GAs) (Michalewicz, 1992), see also (Michalewicz et al., 1992) for an interesting treatment of the linear discrete-time problem.

Genetic algorithms provide a powerful mechanism towards a global search for the optimum, but in many cases, the convergence is very slow. However, as will be shown in this paper, if the GA is supplemented by problem specific heuristics, the convergence can be accelerated significantly. It is well known that GAs are based on a guided random search through the genetic operators and evolution by artificial selection. This process is inherently very slow, because the search space is very large and evolution progresses step by step, exploring many regions with solutions of low fitness. However, it is often possible to guide the search further, by incorporating qualitative knowledge about potential good solutions. In many problems, this might involve simple heuristics, which when combined with the genetic search, provide a powerful tool for finding the optimum very quickly.

The purpose of the present work is to incorporate problem specific heuristic arguments, which when combined with a modified hybrid GA, can solve the discrete-time optimal control problem very easily. There are significant advantages to this approach. First, the need to solve a difficult two-point boundary value problem (TPBVP) is completely avoided. Instead, only initial value problems (IVP) need to be solved. Second, after finding an optimal solution, we verify that it approximately satisfies the first-order necessary conditions for a stationary solution, so the mathematical soundness of the traditional necessary conditions is retained. Furthermore, after obtaining a solution by direct genetic search, the static and dynamic Lagrange multipliers, i.e., the adjoint variables, can be computed and compared with the results from a gradient method. All this is achieved without directly solving the TPBVP. There is a price to be paid, however, since, in the process, we are solving many initial value problems (IVPs). This might present a challenge in more advanced and difficult problems, where the dynamics are described by higher order systems of ordinary differential equations, or when the equations are difficult to integrate over the required time interval and special methods of numerical integration are required. On the other hand, if the system is described by discrete-time difference equations that are relatively well behaved and easy to iterate, the need to solve the initial value problem many times does not represent a serious problem. For instance, the example problem presented here , the discrete velocity programming problem (DVDP) with the combined effects of gravity, thrust and drag, together with a terminal constraint (Bryson, 1999), runs on a 1.6 GHz pentium 4 processor in less than one minute CPU time.

In the next section, a mathematical formulation of the discrete time optimal control problem is given. This formulation is used to study a specific example of a discrete time problem, namely the velocity direction programming of a body moving in a viscous fluid. Details of this problem are given in Section 3. The evolutionary computation approach to the solution is then described in Section 4 where results are presented and compared with the results of an indirect gradient method developed by Bryson (Bryson, 1999). In Section 5, a mathematical formulation of the continuous time optimal control problem for nonlinear dynamical systems is presented. A specific illustrative example of a continuous time optimal control problem is described in Section 6, where we study the Goddard's problem of rocket dynamics using the proposed evolutionary computation method. Finally conclusions are summarized in Section 7.

2 OPTIMAL CONTROL OF DISCRETE TIME NONLINEAR SYSTEMS

In this section, a formulation is developed for the nonlinear discrete-time optimal control problem subject to terminal constraints. Consider the nonlinear discrete-time dynamical system described by difference equations with initial conditions

$$\boldsymbol{x}(i+1) = \boldsymbol{f}[\boldsymbol{x}(i), \boldsymbol{u}(i), i] \tag{2.1}$$

$$\boldsymbol{x}(0) = \boldsymbol{x_0} \tag{2.2}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of state variables, $\boldsymbol{u} \in \mathbb{R}^p$, $p < n$ is the vector of control variables and $i \in [0, N-1]$ is a discrete time counter. The function $\boldsymbol{f}$ is a nonlinear function of the state vector, the control vector and the discrete time i, i.e., $\boldsymbol{f} : \mathbb{R}^n$ x $\mathbb{R}^p$ x $\mathbb{R} \mapsto \mathbb{R}^n$. Next, define a performance index

$$J[\boldsymbol{x}(i), \boldsymbol{u}(i), i, N] = \phi[\boldsymbol{x}(N)] + \Sigma_{i=0}^{M} L[\boldsymbol{x}(i), \boldsymbol{u}(i), i] \tag{2.3}$$

where

$$M = N-1, \;\; \phi : \mathbb{R}^n \mapsto \mathbb{R}, \;\; L : \mathbb{R}^n \mathrm{x} \mathbb{R}^p \mathrm{x} \mathbb{R} \mapsto \mathbb{R}$$

Here L is the Lagrangian function and $\phi[\boldsymbol{x}(N)]$ is a function of the terminal value of the state vector $\boldsymbol{x}(N)$. In some problems, additional terminal constraints can be prescribed through the use of functions ψ of the state variables $\boldsymbol{x}(N)$

$$\psi[\boldsymbol{x}(N)] = 0, \qquad \psi : \mathbb{R}^n \mapsto \mathbb{R}^k \quad k \leq n \tag{2.4}$$

The optimal control problem consists of finding the control sequence $\boldsymbol{u}(i)$ such as to maximize (or minimize) the performance index defined by (2.3), subject to the dynamical equations (2.1) with initial conditions (2.2) and terminal constraints (2.4). This formulation is known as the Bolza problem in the calculus of variations. In an alternative formulation, due to Mayer, the state vector $x_j,\ j \in [1, n]$ is augmented by an additional state variable x_{n+1} which satisfies the initial value problem:

$$x_{n+1}(i+1) = x_{n+1}(i) + L[\boldsymbol{x}(i), \boldsymbol{u}(i), i] \tag{2.5}$$

$$x_{n+1}(0) = 0 \tag{2.6}$$

The performance index can then be written in the following form

$$J(N) = \phi[\boldsymbol{x}(N)] + x_{n+1}(N) \equiv \phi_a[\boldsymbol{x_a}(N)] \tag{2.7}$$

where $\boldsymbol{x_a} = [\boldsymbol{x} \ \ x_{n+1}]^T$ is the augmented state vector and ϕ_a the augmented performance index. In this paper, the Meyer formulation is used.

We next define an augmented performance index with adjoint constraints ψ and adjoint dynamical constraints $\boldsymbol{f}[\boldsymbol{x}(i), \boldsymbol{u}(i), i] - \boldsymbol{x}(i+1) = 0$, with static and dynamical Lagrange multipliers ν and λ, respectively, in the following form:

$$J_a = \phi + \nu^T \psi + \lambda^T(0)[\boldsymbol{x_0} - \boldsymbol{x}(0)] + \\ + \Sigma_{i=0}^{M} \lambda^T(i+1)\{\boldsymbol{f}[\boldsymbol{x}(i), \boldsymbol{u}(i), i] - \boldsymbol{x}(i+1)\} \tag{2.8}$$

Define a Hamiltonian function as

$$H(i) = \lambda^T(i+1)\boldsymbol{f}[\boldsymbol{x}(i), \boldsymbol{u}(i), i] \tag{2.9}$$

Rewriting the augmented performance index in terms of the Hamiltonian function, we get

$$J_a = \phi + \nu^T \psi - \lambda^T(N)\boldsymbol{x}(N) + \lambda^T(0)\boldsymbol{x_0} + \\ + \Sigma_{i=0}^{M}[H(i) - \lambda^T(i)\boldsymbol{x}(i)] \tag{2.10}$$

A first order necessary condition for J_a to reach a stationary solution is given by the discrete version of the Euler-Lagrange equations

$$\lambda^T(i) = H_{\boldsymbol{x}}(i) = \lambda^T(i+1)\boldsymbol{f}_{\boldsymbol{x}}[\boldsymbol{x}(i), \boldsymbol{u}(i), i] \tag{2.11}$$

with final conditions

$$\lambda^T(N) = \phi_x + \nu^T \psi_{\boldsymbol{x}} \tag{2.12}$$

The control $\boldsymbol{u}(i)$ satisfies the optimality condition:

$$H_{\boldsymbol{u}}(i) = \lambda^T(i+1)\boldsymbol{f}_{\boldsymbol{u}}[\boldsymbol{x}(i), \boldsymbol{u}(i), i] = 0 \tag{2.13}$$

If we define an augmented function Φ as

$$\Phi = \phi + \nu^T \psi \tag{2.14}$$

then the final conditions can be written in terms of the augmented function Φ in a similar way to the problem without terminal constraints

$$\lambda^T(N) = \Phi_{\boldsymbol{x}} = \phi_{\boldsymbol{x}} + \nu^T \psi_{\boldsymbol{x}} \tag{2.15}$$

The indirect approach to optimal control uses the necessary conditions for an optimum to obtain a solution. In this approach, the state equations (2.1) with initial conditions (2.2) need to be solved together with the adjoint equations (2.11) and the final conditions (2.15), where the control sequence $\boldsymbol{u}(i)$ is to be determined from the optimality condition (2.13). This represents a coupled system of nonlinear difference equations with part of the boundary conditions specified at the initial time $i = 0$ and the rest of the boundary conditions specified at the final time $i = N$. This is a nonlinear two-point boundary value problem (TPBVP) in difference equations. Except for some special simplified cases, it is usually very difficult to obtain solutions for such nonlinear TPBVPs in closed form. Therefore, many numerical methods have been developed to tackle this problem.

Several gradient based methods have been proposed for solving the discrete-time optimal control problem (Mayne, 1966). For example, Murray and Yakowitz (Murray and Yakowitz, 1984) and (Yakowitz and Rutherford, 1984) developed a differential dynamic programming and Newton's method for the solution of discrete optimal control problems, see also the book of Jacobson and Mayne (Jacobson and Mayne, 1970), (Ohno, 1978), (Pantoja, 1988) and (Dunn and Bertsekas, 1989). Similar methods have been further developed by Liao and Shoemaker (Liao and Shoemaker, 1991). Another method, the trust region method, was proposed by Coleman and Liao (Coleman and Liao, 1995) for the solution of unconstrained discrete-time optimal control problems. Although confined to the unconstrained problem, this method works for large scale minimization problems.

In contrast to the indirect approach, in the present proposed approach, the optimality condition (2.13) and the adjoint equations (2.11) together with their

final conditions (2.15) are not used in order to obtain the optimal solution. Instead, the optimal values of the control sequence $\boldsymbol{u}(i)$ are found by a modified genetic search method starting with an initial population of solutions with values of $\boldsymbol{u}(i)$ randomly distributed within a given domain of admissible controls. During the search, approximate, not necessarily optimal values of the solutions $\boldsymbol{u}(i)$ are found for each generation. With these approximate values known, the state equations (2.1) together with their initial conditions (2.2) are very easy to solve as an initial value problem, by a straightforward iteration of the difference equations from $i = 0$ to $i = N - 1$. At the end of this iterative process, the final values $\boldsymbol{x}(N)$ are obtained, and the fitness function can be determined. The search than seeks to maximize the fitness function F such as to fulfill the goal of the evolution, which is to maximize $J(N)$, as given by the following Eq.(2.16), subject to the terminal constraints as defined by Eq.(2.17).

$$\text{maximize } J(N) = \phi[\boldsymbol{x}(N)] \tag{2.16}$$

subject to the dynamical equality constraints, Eqs. (2.1-2.2) and to the terminal constraints (2.4), which are repeated here for convenience as Eq.(2.17)

$$\begin{gathered}\psi[\boldsymbol{x}(N)] = 0 \\ \psi : \mathbb{R}^n \mapsto \mathbb{R}^k \quad k \leq n\end{gathered} \tag{2.17}$$

Since we are using a direct search method, condition (2.17) can also be stated as a search for a maximum, namely we can set a goal which is equivalent to (2.17) in the form

$$\text{maximize } J_1(N) = -\psi^{\mathrm{T}}[\boldsymbol{x}(N)]\psi[\boldsymbol{x}(N)] \tag{2.18}$$

The fitness function F can now be defined by

$$\begin{gathered}F(N) = \alpha J(N) + (1-\alpha)J_1(N) = \\ = \alpha\phi[\boldsymbol{x}(N)] - (1-\alpha)\psi^{\mathrm{T}}[\boldsymbol{x}(N)]\psi[\boldsymbol{x}(N)]\end{gathered} \tag{2.19}$$

with $\alpha \in [0,1]$ and $\boldsymbol{x}(N)$ determined from a solution of the original initial value problem for the state variables:

$$\boldsymbol{x}(i+1) = \boldsymbol{f}[\boldsymbol{x}(i), \boldsymbol{u}(i), i], \quad i \in [0, N-1] \tag{2.20}$$

$$\boldsymbol{x}(0) = \boldsymbol{x_0} \tag{2.21}$$

3 VELOCITY DIRECTION CONTROL OF A BODY IN A VISCOUS FLUID

In this section, we treat the case of controlling the motion of a particle moving in a viscous fluid medium by varying the direction of a thrust vector of constant magnitude. We describe the motion in a cartesian system of coordinates in which x is pointing to the right and y is positive downward. The constant thrust force $\boldsymbol{F}$ is acting along the path, i.e. in the direction of the velocity vector $\boldsymbol{V}$ with magnitude $F = amg$. The acceleration of gravity g is acting downward in the positive y direction. The drag force is proportional to the square of the speed and acts in a direction opposite to the velocity vector $\boldsymbol{V}$. The motion is controlled by varying the angle γ, which is positive downward from the horizontal. The velocity direction γ is to be programmed such as to achieve maximum range and fulfill a prescribed terminal constraint on the vertical final location y_f. Newton's second law of motion for a particle of mass m can be written as

$$mdV/dt = mg(a + sin\gamma) - \frac{1}{2}\rho V^2 C_D S \tag{3.1}$$

where ρ is the fluid density, C_D is the coefficient of drag and S is a typical cross-section area of the body. For example, if the motion of the center of gravity of a spherical submarine vehicle is considered, then S is the maximum cross-section area of the vehicle and C_D would depend on the Reynolds number $Re = \rho V d/\mu$, where μ is the fluid viscosity and d the diameter of the vehicle. Dividing (3.1) by the mass m, we obtain

$$dV/dt = g(a + sin\gamma) - V^2/L_c \tag{3.2}$$

The length $L_c = 2m/(\rho S C_D)$ is a typical hydrodynamic length. The other equations of motion are:

$$dx/dt = V cos\gamma \tag{3.3}$$

$$dy/dt = V sin\gamma \tag{3.4}$$

with initial conditions and final constraint

$$V(0) = 0, \quad x(0) = 0, \quad y(0) = 0 \tag{3.5}$$

$$y(t_f) = y_f \tag{3.6}$$

In order to rewrite the equations in nondimensional form, we introduce the following nondimensional variables, denoted by primes:

$$t = (L_c/g)^{1/2} t\prime, \quad V = (gL_c)^{1/2} V\prime$$

$$x = L_c x\prime, \;\; y = L_c y\prime \tag{3.7}$$

where we have chosen L_c as the characteristic length. Substituting the nondimensional variables (3.7) in the equations of motion (3.2-3.4) and omitting the prime notation, we obtain the nondimensional state equations

$$dV/dt = a + sin\gamma - V^2 \tag{3.8}$$

$$dx/dt = V cos\gamma \tag{3.9}$$

$$dy/dt = V sin\gamma \tag{3.10}$$

In order to formulate a discrete time version of this problem, we first rewrite (3.8) in separated variables form as $dV/(a + sin\gamma - V^2) = dt$. Integrating and using the condition $V(0) = 0$, we get

$$(1/b) argtanh(V/b) = t \tag{3.11}$$

Solving for the speed, we obtain

$$V = btanh(bt) \tag{3.12}$$

$$b = (a + sin\gamma)^{1/2} \tag{3.13}$$

We now develop a discrete time model by dividing the trajectory into a finite number N of straight line segments of fixed duration $\Delta T = t_f/N$ along which the control γ is kept constant. The time at the end of each segment is given by $t(i) = i\Delta T$, with i a time step counter at point i. The time is normalized by $(L_c/g)^{1/2}$, so the nondimensional final time is $t\prime_f = t_f/(L_c/g)^{1/2}$ and the nondimensional time at step i is $t\prime(i) = it\prime_f/N$. The nondimensional time interval is $(\Delta T)\prime = t\prime_f/N$. Writing (3.12) at $t(i+1) = (i+1)\Delta T$, we obtain the velocity at the point $(i+1)$ along the trajectory

$$V(i+1) = b(i) tanh[b(i)(i+1)t\prime_f/N] \tag{3.14}$$

$$b(i) = (a + sin\gamma(i))^{1/2} \tag{3.15}$$

Similarly, substituting the time $t\prime(i) = it\prime_f/N$ in (3.11), the following expression is obtained, which we define as the function $G_0(i)$.

$$ib(i)t\prime_f/N = argtanh[V(i)/b(i)] \equiv G_0(i) \tag{3.16}$$

Introducing a second function $G_1(i)$ defined by

$$G_1(i) = G_0(i) + b(i)t\prime_f/N \tag{3.17}$$

Eq.(3.14) can be written as

$$V(i+1) = b(i) tanh[G_1(i)] \tag{3.18}$$

We now determine the coordinates x and y as a function of time. Using the state equation (3.9) together with the result (3.12) and defining $\theta = bt$, we obtain

$$\begin{aligned} dx &= V cos\gamma dt = b cos\gamma tanh(bt) dt = \\ &= cos\gamma tanh\theta d\theta \end{aligned} \tag{3.19}$$

Integrating along a straight line segment between points i and $i+1$, we get

$$x(i+1) = x(i) +$$

$$+cos\gamma(i)[logcosh\theta(i+1) - logcosh\theta(i)] \tag{3.20}$$

$$\theta(i) = ib(i)t\prime_f/N = G_0(i) \tag{3.21}$$

$$\theta(i+1) = b(i)(i+1)t\prime_f/N = ib(i)t\prime_f/N +$$

$$+b(i)t\prime_f/N = G_0(i) + b(i)t\prime_f/N = G_1(i) \tag{3.22}$$

Substituting (3.21-3.22) in (3.20), we obtain the following discrete-time state equation (3.24) for the location $x(i+1)$. The equation for the coordinate $y(i+1)$ can be developed in a similar way to $x(i+1)$, with $cos\gamma(i)$ replaced by $sin\gamma(i)$. Adding the state equation (3.18) for the velocity $V(i+1)$, which is repeated here as Eq.(3.23), the state equations become:

$$V(i+1) = b(i) tanh[G_1(i)] \tag{3.23}$$

$$x(i+1) = x(i) +$$

$$+cos\gamma(i)\, log[coshG_1(i)/coshG_0(i)] \tag{3.24}$$

$$y(i+1) = y(i) +$$

$$+sin\gamma(i) log[coshG_1(i)/coshG_0(i)] \tag{3.25}$$

with initial conditions and terminal constraint

$$V(0) = 0, \quad x(0) = 0, \quad y(0) = 0 \tag{3.26}$$

$$y(N) = y\prime_f = y_f/L_c \tag{3.27}$$

The optimal control problem now consists of finding the sequence $\gamma(i)$ for $i \in [0, N-1]$ such as to maximize the range $x(N)$, subject to the state equations (3.23-3.25), the initial conditions (3.26) and the terminal constraint (3.27), where $y\prime_f$ is in units of L_c and the final time $t\prime_f$ in units of $(L_c/g)^{1/2}$.

4 EVOLUTIONARY APPROACH TO OPTIMAL CONTROL

We now describe the proposed direct approach which is based on a genetic search method. As was previously mentioned, an important advantage of this approach is that there is no need to solve the two-point boundary value problem described by the state equations (2.1) and the adjoint equations (2.11), together with the initial conditions (2.2), the final conditions (2.15), the terminal constraints (2.4) and the optimality condition (2.13) for the optimal control $u(i)$.

Instead, the direct evolutionary computation method allows us to evolve a population of solutions such as to maximize the objective function or fitness function $F(N)$. The initial population is built by generating a random population of solutions $\gamma(i)$, $i \in [0, N-1]$, uniformly distributed within a domain $\gamma \in [\gamma_{\min}, \gamma_{\max}]$. Typical values are $\gamma_{\max} = \pi/2$ and either $\gamma_{\min} = -\pi/2$ or $\gamma_{\min} = 0$ depending on the problem. The genetic algorithm evolves this initial population using the operations of selection, mutation and crossover over many generations such as to maximize the fitness function:

$$F(N) = \alpha J(N) + (1-\alpha) J_1(N) = \\ = \alpha \phi[\xi(N)] - (1-\alpha)\psi^{\mathrm{T}}[\xi(N)]\psi[\xi(N)] \quad (4.1)$$

with $\alpha \in [0,1]$ and $J(N)$ and $J_1(N)$ given by:

$$J(N) = \phi[\xi(N)] = x(N) \quad (4.2)$$

$$J_1(N) = \psi^2[\xi(N)] = (y(N) - y_f)^2 \quad (4.3)$$

For each member in the population of solutions, the fitness function depends on the final values $x(N)$ and $y(N)$, which are determined by solving the initial value problem defined by the state equations (3.23-3.25) together with the initial conditions (3.26). This process is repeated over many generations. Here, we run the genetic algorithm for a predetermined number of generations and then we check if the terminal constraint (3.27) is fulfilled. If the constraint is not fulfilled, we can either increase the number of generations or readjust the weight $\alpha \in [0,1]$.

Each member in the population consists of a sequence of values of the control history at the discrete time intervals, e.g., $u(i)$ or $\gamma(i)$. These are the variables to be searched by the genetic algorithm. Since the discrete time control problem is an approximation of the continuous time problem, the discrete control sequence is an approximation of either a continuous function of time or, possibly a piecewise continuous function with a finite number of switching points. We have not treated any problems with a discontinuous control history, such as in bang-bang control. We believe the present method can be extended to treat such problems, but it is not clear at the present time how to approach the problem in order to obtain the switching points.

If we restrict our discussion to problems with continuous controls, then there is a question of how to obtain a smooth control history from the discrete values obtained through the genetic search. We have tried smoothing by polynomial approximation of the discrete control sequence, which works well for many problems. There is also a possibility of parameterisation of the discrete control sequence by series of orthogonal polynomials, such as in approximation theory. In this case the genetic algorithm can search the coefficients of the polynomials directly.

We now present results obtained by solving this problem using the proposed approach. We first treat the case where $x(N)$is maximized with no constraint placed on y_f. We solve an example where the value of the thrust is $a = 0.05$ and the final time is $t_f = 5$.

The evolution of the solution over 50 generations is shown in Figure 1. The control sequence $\gamma(i)$, the optimal trajectory and the velocity $V^2(i)$ are displayed in Figure 2. The sign of y is reversed for plotting. It can be seen from the plots that the angle varies at the beginning and at the end of the motion, but remains constant in the middle of the maneuver, resulting in a dive along a straight line along a considerable portion of the trajectory. This finding agrees well with the results obtained by Bryson (Bryson, 1999) using a gradient method.

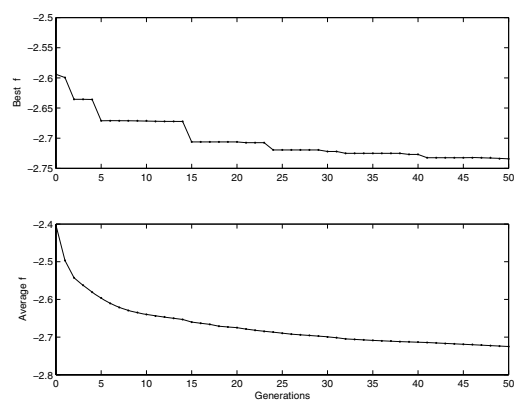


Figure 1: Convergence of the DVDP solution with gravity, thrust a=0.05 and drag, with no terminal constraint on y_f. $x(N)$ is maximized. With final time $t_f = 5$.

Another case that was studied is the case where a control sequence is sought such as to bring the mass m to a required vertical location y_f without maximizing the horizontal distance $x(N)$. In order to fulfill this terminal constraint, we minimize $(y(N) - y_f)^2$. The results are shown in Figure 3. The required

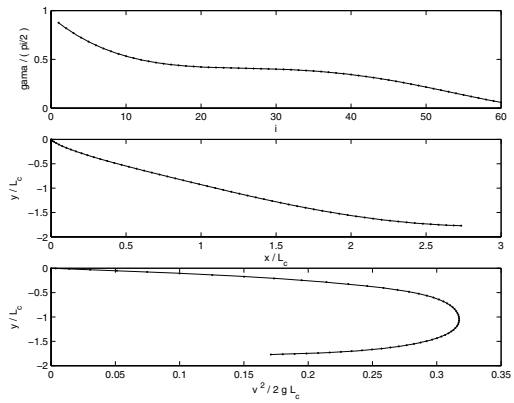


Figure 2: The control sequence, the optimal trajectory and the velocity $V^2(i)$ for the DVDP problem with gravity, thrust a=0.05 and drag. No terminal constraint on y_f. Final time $t_f = 5$. The sign of y is reversed for plotting.

control sequence, the shape of the trajectory and the velocity squared are displayed in the figure. It can be seen that the terminal constraint $y(N) = y_f = 1$ is fulfilled.

5 NONLINEAR CONTINUOUS TIME OPTIMAL CONTROL

In this section, a formulation is developed for the nonlinear continuous time optimal control problem subject to terminal constraints. Consider the continuous time nonlinear problem described by a system of ordinary differential equations with initial conditions

$$d\boldsymbol{x}/dt = \boldsymbol{f}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \tag{5.1}$$

$$\boldsymbol{x}(0) = \boldsymbol{x_0} \tag{5.2}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of state variables, $\boldsymbol{u} \in \mathbb{R}^p$, $p < n$ is the vector of control variables and $t \in [0, t_f]$ is the continuous time. The function $\boldsymbol{f}$ is a nonlinear function of the state vector, the control vector and the time t, i.e., $\boldsymbol{f} : \mathbb{R}^n$ x $\mathbb{R}^p$ x $\mathbb{R} \mapsto \mathbb{R}^n$. Next, define a performance index

$$J[\boldsymbol{x}(t), \boldsymbol{u}(t), t_f] = \phi[\boldsymbol{x}(t_f)] + \int_0^{t_f} L[\boldsymbol{x}(t), \boldsymbol{u}(t), t]dt \tag{5.3}$$

$$\phi : \mathbb{R}^n \mapsto \mathbb{R}, \quad L : \mathbb{R}^n \text{x} \mathbb{R}^p \text{x} \mathbb{R} \mapsto \mathbb{R}$$

Here L is the Lagrangian function and $\phi[\boldsymbol{x}(t_f)]$ is a function of the terminal value of the state vector $\boldsymbol{x}(t_f)$. In some problems, additional terminal constraints can be prescribed through the use of functions

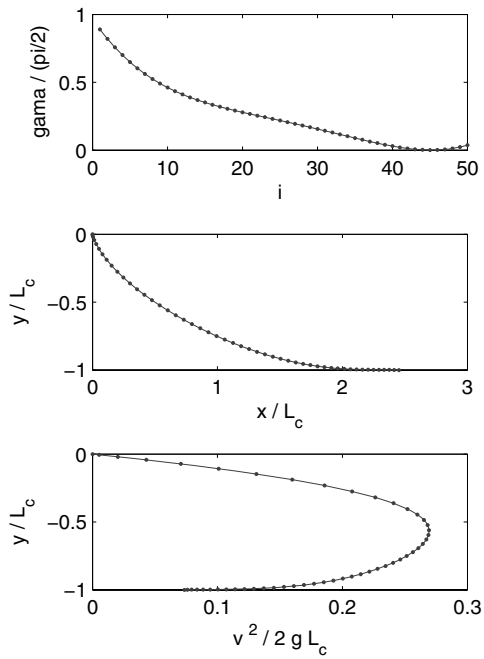


Figure 3: The control sequence, the optimal trajectory and the velocity $V^2(i)$ for the DVDP problem with gravity, thrust a=0.05 and drag and terminal constraint on $y_f = 1$. Final time $t_f = 5$. $(y(N) - y_f)^2$ is minimized. Here $x(N)$ is not maximized. The sign of y is reversed for plotting.

ψ of the state variables $\boldsymbol{x}(t_f)$

$$\psi[\boldsymbol{x}(t_f)] = 0, \qquad \psi : \mathbb{R}^n \mapsto \mathbb{R}^k, \quad k \leq n \tag{5.4}$$

The formulation of the optimal control problem according to Bolza consists of finding the control $\boldsymbol{u}(t)$ such as to maximize the performance index defined by (5.3), subject to the state equations (5.1) with initial conditions (5.2) and terminal constraints (5.4). In the alternative formulation, due to Mayer, the state vector x_j, $j \in [1, n]$ is augmented by an additional state variable x_{n+1} which satisfies the following initial value problem:

$$dx_{n+1}/dt = L[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \tag{5.5}$$

$$x_{n+1}(0) = 0 \tag{5.6}$$

The performance index can then be written as

$$J(t_f) = \phi[\boldsymbol{x}(t_f)] + x_{n+1}(t_f) \equiv \phi_a[\boldsymbol{x_a}(t_f)] \tag{5.7}$$

where $\boldsymbol{x_a} = [\boldsymbol{x} \;\; x_{n+1}]^T$ is the augmented state vector and ϕ_a the augmented performance index. We

next define an augmented performance index with adjoint constraints ψ and adjoint dynamical constraints $\boldsymbol{f}\,[\boldsymbol{x}(t), \boldsymbol{u}(t), t\,] - d\boldsymbol{x}/dt = 0$, with static and dynamical Lagrange multipliers ν and λ as:

$$J_a(t_f) = \phi[\boldsymbol{x}(t_f)] + \nu^T \psi[\boldsymbol{x}(t_f)] + \lambda^T(0)[\boldsymbol{x_0} - \boldsymbol{x}(0)]$$

$$+ \int_0^{t_f} \lambda^T(t)[\boldsymbol{f}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] - d\boldsymbol{x}/dt]dt \qquad (5.8)$$

Introducing a Hamiltonian function

$$H(t) = \lambda^T(t)\boldsymbol{f}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \qquad (5.9)$$

and rewriting the augmented performance index in terms of the Hamiltonian, we get

$$J_a(t_f) = \phi + \nu^T \psi - \lambda^T(t_f)\boldsymbol{x}(t_f) + \lambda^T(0)\boldsymbol{x_0} +$$

$$+ \int_0^{t_f} [H(t) - \lambda^T(t)\boldsymbol{x}(t)]dt \qquad (5.10)$$

A first order necessary condition for J_a to reach a stationary solution is given by the Euler-Lagrange equations

$$d\lambda^T/dt = -H_{\boldsymbol{x}}(t) = -\lambda^T(t)\boldsymbol{f}_{\boldsymbol{x}}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \qquad (5.11)$$

with final conditions

$$\lambda^T(t_f) = \phi_x[\boldsymbol{x}(t_f)] + \nu^T\, \psi_{\boldsymbol{x}}[\boldsymbol{x}(t_f)] \qquad (5.12)$$

The control $\boldsymbol{u}(t)$ satisfies the optimality condition:

$$H_{\boldsymbol{u}}(t) = \lambda^T(t)\boldsymbol{f}_{\boldsymbol{u}}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] = 0 \qquad (5.13)$$

If we define an augmented function $\Phi[\boldsymbol{x}(t_f)]$ as

$$\Phi[\boldsymbol{x}(t_f)] = \phi[\boldsymbol{x}(t_f)] + \nu^T\, \psi[\boldsymbol{x}(t_f)] \qquad (5.14)$$

then the final conditions can be written in terms of the augmented function Φ in a similar way to the problem without terminal constraints

$$\lambda^T(t_f) = \Phi_{\boldsymbol{x}}[\boldsymbol{x}(t_f)] = \phi_{\boldsymbol{x}}[\boldsymbol{x}(t_f)] + \nu^T\, \psi_{\boldsymbol{x}}[\boldsymbol{x}(t_f)] \qquad (5.15)$$

In the indirect approach to optimal control, the necessary conditions are used to obtain an optimal solution: the state equations (5.1) with initial conditions (5.2) have to be solved together with the adjoint equations (5.11) and the final conditions (5.15). The control history $u(t)$ is determined from the optimality condition (5.13). Consequently, this approach leads to a coupled system of nonlinear ordinary differential equations with the boundary conditions for the state variables specified at the initial time $t = 0$ and the boundary conditions for the adjoint variables specified at the final time $t = t_f$. This is a nonlinear two-point boundary value problem (TPBVP) in ordinary differential equations. Except for some special simplified cases, it is usually very difficult to obtain solutions for such nonlinear TPBVPs analytically. Many numerical methods have been developed in order to obtain approximate solutions to this problem.

6 GODDARD'S OPTIMAL CONTROL PROBLEM IN ROCKET DYNAMICS

We now illustrate the above approach with a continuous time optimal control example. We apply the optimal control formulation described in the previous section for continuous time dynamical systems to the study of the vertical climb of a single stage sounding rocket launched vertically from the ground. This is known in the literature as the Goddard's problem. The problem is to control the thrust of the rocket such as to maximize the final velocity or the final altitude. There are two versions to this problem: in the first version, the final mass of the rocket is prescribed and the final time is free. In the second version, the final time is prescribed and the final mass is free. The second version of this problem will be presented.

Let $h(t)$ denote the altitude of the rocket as measured from sea level and $v(t)$ and $m(t)$ the velocity and the mass of the rocket, respectively. Here the time t is continuous. The trajectory of the rocket is a vertical straight line. The forces acting on the rocket are the thrust $T(t)$, which is used as the control variable or control history, the aerodynamic drag force $D(h, v)$, which is a function of altitude and speed and the weight of the rocket $m(t)g$, where $m(t)$ is the mass and g is the acceleration of gravity, assumed constant. The equations of motion are:

$$dh/dt = v \qquad (6.1)$$

$$mdv/dt = T - D - mg \qquad (6.2)$$

$$dm/dt = -T/c \qquad (6.3)$$

where the drag force is given by

$$D(h, v) = D_0\, v^2\, exp(-h/h_r) \qquad (6.4)$$

Here $h_r = 23800$ ft is a characteristic altitude and D_0 is a characteristic drag force given by

$$D_0 = 0.711 T_M/c^2 \qquad (6.5)$$

where T_M is the maximum thrust developed by the rocket. The speed c is the propellant jet exhaust speed. An important parameter in rocket dynamics is the thrust to weight ratio $\tau = T_M/(m_0 g)$, where m_0 is the initial mass of the vehicle and $g = 32.2\ ft/s^2$. In this example, a ratio of 2 is chosen:

$$\tau = T_M/(m_0 g) = 2 \tag{6.6}$$

Here we take a value of $m_0 = 3$ slugs for a small experimental rocket. A typical value of the exhaust speed c and the specific impulse I_{sp} for an early rocket such as the one tested by Goddard is given by

$$c = (3.264 g h_r)^{1/2} = 1581 ft/s$$

$$I_{sp} = c/g = 49.14 sec \tag{6.7}$$

The initial conditions are

$$h(0) = 0, \quad v(0) = 0, \quad m(0) = m_0 = 3\ \ slugs \tag{6.8}$$

The optimal control problem is to find the control history $T(t)$ such as to maximize the final altitude (or the altitude at burnout) $h(t_f)$ in a given time t_f, where a value $t_f = 18$ sec was used in this example. The state equations (6.1-6.3) with the initial conditions (6.8) are to be solved in the optimization process. Before solving this problem, we first restate the problem in non-dimensional form. Choosing the characteristic speed $(g h_r)^{1/2} = 875$ ft/s and the characteristic time $(h_r/g)^{1/2} = 27.2$ sec, we introduce nondimensional variables, denoted here by primes:

$$h = h_r h\prime, \quad v = (g h_r)^{1/2} v\prime, \quad t = (h_r/g)^{1/2} t\prime$$

$$m = m_0 m\prime, \quad T = T_M T\prime, \quad D = T_M D\prime \tag{6.9}$$

Introducing the variables from Eq.(6.9) in the state equations (6.1-6.3) and simplifying, the following system of non-dimensional equations is obtained:

$$dh/dt = v \tag{6.10}$$

$$m dv/dt = \tau T - \tau \sigma^2 v^2 exp(-h) - m \tag{6.11}$$

$$dm/dt = -1.186\, \sigma \tau T \tag{6.12}$$

In this system of equations (6.10-6.12) all the variables are non-dimensional and the prime notation has been omitted. Two independent non-dimensional parameters characterizing this problem are obtained: the thrust to weight ratio τ, introduced before and a ratio of two speeds σ defined by

$$\tau = T_M/(m_0 g) = 2, \quad T_M = 193\ \ lbs$$

$$\sigma = (0.711 g h_r)^{1/2}/c = 0.467 \tag{6.13}$$

The non-dimensional initial conditions are:

$$h(0) = 0, \quad v(0) = 0, \quad m(0) = 1 \tag{6.14}$$

The optimal control problem is to find the control history $T(t)$ such as to maximize the final altitude (the altitude at burnout) $h(t_f)$ in a given normalized time

$$t\prime_f = t_f/(h_r/g)^{1/2} = 0.662$$

subject to the state equations (6.10-6.12) and the initial conditions (6.14). The results of this continuous time optimal control problem are given in Figures 4 and 5. Figure 4 shows the control history of the thrust as a function of the time in seconds. In the genetic search, the search range for the normalized thrust was between a lower bound of $(T/T_{max})_L = 0.1$ and an upper bound $(T/T_{max})_U = 1$. It can be seen that the thrust increases sharply during the first two seconds of the flight and remains closer to 1 afterwards. Figure 5 displays the state variables, the altitude, the velocity and the mass of the rocket as a function of time. The mass of the rocket decreases almost linearly during the flight due to the optimal control which requires an almost constant thrust. This is in agreement with the results of Betts (Betts, Eldersveld and Huffman, 1993) and (Bondarenko et al., 1999) who used a nonlinear programming method. See also (Dolan and More, 2000).

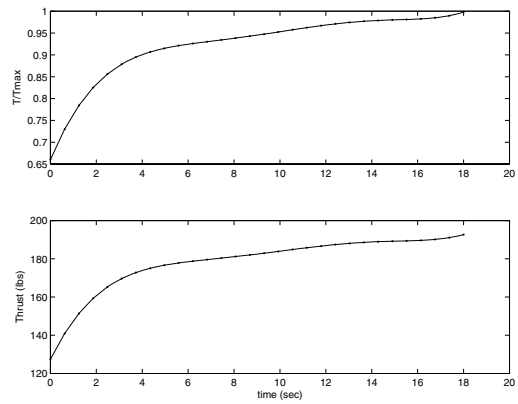


Figure 4: The thrust control history. The upper graph displays the value of the thrust normalized by maximum thrust. The lower graph shows the actual thrust in lbs.

7 CONCLUSIONS

A new method for solving both discrete time and continuous time nonlinear optimal control problems with

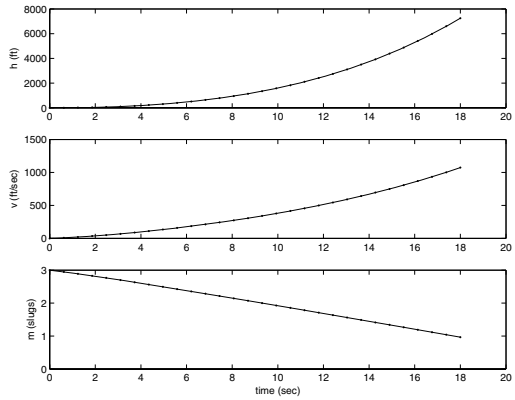


Figure 5: The state variables, the altitude, velocity and mass of the rocket as a function of time

terminal constraints has been presented. Unlike other methods that use the first-order necessary conditions to find the optimum, the present method seeks the best control history directly by a modified genetic search. As a consequence of this direct search approach, there is no need to develop a Hamiltonian formulation and therefore there is no need to solve a difficult two-point boundary value problem for the state and adjoint variables. This has a significant advantage in more advanced and higher order problems where it is difficult to solve the two point boundary value problem (TPBVP) with large systems of ordinary differential equations. There is a computational price to be paid, however, since the method involves repetitive solutions of the initial value problem (IVP) for the state variables during the evolutionary process.

The method was demonstrated by solving a discrete-time optimal control problem, namely, the discrete velocity direction programming problem (DVDP) of a body with the effects of gravity, thrust and hydrodynamic drag. Benchmark problems of this kind were pioneered by Bryson who used analytical and gradient methods. This discrete time problem was solved easily using the proposed approach and the results compared favorably with the results of Bryson.

The method was also applied to a continuous time nonlinear optimal control problem, the Goddard's problem of rocket dynamics. The results compared favorably with published results obtained by a nonlinear programming method.

REFERENCES

Betts, J. (2001). *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, PA.

Betts, J., Eldersveld, S. and Huffman, W. (1993). Sparse nonlinear programming test problems. In *Technical Report BCSTECH-93-047*. Boeing Computer Services, Seattle, Washington.

Bondarenko, A.S., Bortz, D.M. and More, J.J. (1999). Cops: Large scale nonlinearly constrained optimization problems. In *Argonne National Laboratory Technical Report ANL / MCS-TM-237*.

Bryson, A. (1999). *Dynamic Optimization*. Addison-Wesley Longman, Menlo Park, CA.

Coleman, T. and Liao, A. (1995). An efficient trust region method for unconstrained discrete-time optimal control problems. In *Computational Optimization and Applications, 4:47–66*.

Dolan, E. and More, J. (2000). Benchmarking optimization software with cops. In *Argonne National Labo-ratory Technical Report ANL / MCS-TM-246*.

Dunn, J. and Bertsekas, D. (1989). Efficient dynamic programming implementations of newton's method for unconstrained optimal control problems. In *J. of Optimization Theory and Applications,63, pp. 23–38*.

Fogel, D. (1998). *Evolutionary Computation, The Fossil Record*. IEEE Press, New York.

Fox, C. (1950). *An Introduction to the Calculus of Variations*. Oxford University Press, London.

Jacobson, D. and Mayne, D. (1970). *Differential Dynamic Programming*. Elsevier Science Publishers, Amsterdam, Netherland.

Kirkpatrick, G. and Vecchi (1983). Optimization by simulated annealing. In *Science, 220:671–680*. AAAS.

Laarhoven, P. and Aarts, E. (1989). *Simulated Annealing: Theory and Applications*. Kluwer, Amsterdam.

Liao, L. and Shoemaker, C. (1991). Convergence in unconstrained discrete-time differential dynamic programming. In *IEEE Transactions on Automatic Control, 36, pp. 692–706*. IEEE.

Mayne, D. (1966). A second-order gradient method for determining optimal trajectories of nonlinear discrete time systems. In *International Journal of Control, 3, pp. 85–95*.

Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures= Evolution Programs*. Springer-Verlag, Berlin.

Michalewicz, Z., Janikow, C.Z. and Krawczyk, J.B. (1992). A modified genetic algorithm for optimal control problems. In *Computers Math. Appl., 23(12),8394*.

Murray, D. and Yakowitz, S. (1984). Differential dynamic programming and newton's method for discrete optimal control problems. In *J. of Optimization Theory and Applications, 43:395–414*.

Ohno, K. (1978). A new approach of differential dynamic programming for discrete time systems. In *IEEE Transactions on Automatic Control, 23, pp. 37–47*. IEEE.

Pantoja, J. (1988). Differential dynamic programming and newton's method. In *International Journal of Control, 53:1539–1553*.

Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V. and Mishchenko, E.F. (1962). *The Mathematical Theory of Optimal Processes*. translated by K.N. Trirogoff, L.W. Neustadt (Ed.), Moscow, interscience, New York edition.

Schwefel, H. (1995). *Evolution and Optimum Seeking*. Wiley, New York.

Yakowitz, S. and Rutherford, B. (1984). Computational aspects of discrete-time optimal control. In *Appl. Math. Comput., 15, pp. 29–45*.

CONTRIBUTORS TO A SIGNAL FROM AN ARTIFICIAL CONTRAST

Jing Hu and George Runger
Arizona State University
Tempe, AZ
jinghu, george.runger@asu.edu

Eugene Tuv
Intel Corporation
Chandler, AZ
eugene.tuv@intel.com

Keywords: Patterns, statistical process control, supervised learning, multivariate analysis.

Abstract: Data from a process or system is often monitored in order to detect unusual events and this task is required in many disciplines. A decision rule can be learned to detect anomalies from the normal operating environment when neither the normal operations nor the anomalies to be detected are pre-specified. This is accomplished through artificial data that transforms the problem to one of supervised learning. However, when a large collection of variables are monitored, not all react to the anomaly detected by the decision rule. It is important to interrogate a signal to determine the variables that are most relevant to or most contribute to the signal in order to improve and facilitate the actions to signal. Metrics are presented that can be used determine contributors to a signal developed through an artificial contrast that are conceptually simple. The metrics are shown to be related to traditional tools for normally distributed data and their efficacy is shown on simulated and actual data.

1 INTRODUCTION

Statistical process control (SPC) is used to detect changes from standard operating conditions. In multivariate SPC a $p \times 1$ observation vector $\mathbf{x}$ is obtained at each sample time. Some statistics, such as Hotelling's statistic (Hotelling, 1947), have been developed to detect whether the observation falls in or out of the control region representing standard operating conditions. This leads to two important comments. First, the control region is defined through an analytical expression which is based on the assumption of normal distribution of the data. Second, after a signal further analysis is needed to determine the variables that contribute to the signal.

Our research is an extension of the classical methods in terms of the above two points. The results in (Hwang et al., 2004) described the design of a control region based only on training data without a distributional assumption. An artificial contrast was developed to allow the control region to be learned through supervised learning techniques. This also allowed for control of the decision errors through appropriate parameter values. The second question is to identify variables that are most relevant to or most contribute to a particular signal. We refer to these variables as *contributors* to the signal. These are the variables that receive priority for corrective action. Many industries use an out-of-control action plan (OCAP) to react to a signal from a control chart. This research enhances and extends OCAP to incorporate learned control regions and large numbers of variables.

A physical event, such as a broken pump or a clogged pipe, might generate a signal from a control policy. However, not all variables might react to this physical event. Instead, when a large collection of variables are monitored, often only a few contribute to the signal from the control policy. For example, although a large collection of variables might be monitored, potentially only the pressure drop across a pump might be sensitive to a clogged pipe. The objective of this work is to identify these contributors in order to improve and facilitate corrective actions.

It has been a challenge for even normal-theory based methods to completely solve this problem. The key issue is the interrelationships between the variables. It is not sufficient to simply explore the marginal distribution of each variable. This is made clear in our illustrations that follow. Consequently,

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 71–78.

early work (Alt, 1985; Doganaksay et al., 1991) required improvement. Subsequent work under normal theory considered joint distributions of all subsets of variables (Mason et al., 1995; Chua and Montgomery, 1992; Murphy, 1987). However, this results in a combinatorial explosion of possible subsets for even a moderate number of variables. In (Rencher, 1993) and (Runger et al., 1996) an approach based on conditional distributions was used that resulted in feasible computations, again for normally distributed data. Only one metric was calculated for each variable. Furthermore, in (Runger et al., 1996) a number of reasonable geometric approaches were defined and these were shown to result in equivalent metrics. Still, one metric was computed for each variable. This idea is summarized briefly in a following section. Although there are cases where the feasible approaches used in (Rencher, 1993) and (Runger et al., 1996) are not sufficient, they are effective in many instances, and the results indicate when further analysis is needed. This is illustrated in a following section.

The method proposed here is a simple, computationally feasible approach that can be shown to generalize the normal-theory methods in (Rencher, 1993) and (Runger et al., 1996). Consequently, it has the advantage of equivalence of a traditional solution under traditional assumptions, yet provides a computationally and conceptually simple extension. In Section 2 a summary is provided of the use of an artificial contrast with supervised learning is to generate a control region. In Section 3 the metric used for contributions is presented. The following section present illustrative examples.

2 CONTROL REGION DESIGN

Modern data collection techniques facilitate the collection of in-control data. In practice, the joint distribution of the variables for the in-control data is unknown and rarely as well-behaved as a multivariate normal distribution. If specific deviations from standard operating conditions are not a priori specified, leaning the control region is a type of unsupervised learning problem. An elegant technique can be used to transform the unsupervised learning problem to a supervised one by using an artificial reference distribution proposed by (Hwang et al., 2004). This is summarized briefly as follows.

Suppose $f(x)$ is an unknown probability density function of in-control data, and $f_0(x)$ is a specified reference density function. Combine the original data set $x_1, x_2, ..., x_N$ sampled from $f_0(x)$ and a random sample of equal size N drawn from $f_0(x)$. If we assign $y = -1$ to each sample point drawn from $f(x)$ and $y = 1$ for those drawn from $f_0(x)$, then learning control region can be considered to define a solution to a two-class classification problem. Points whose predicted y are -1 are assigned to the control region, and classified into the "standard" or "on-target" class. Points with predicted y equal to 1 are are classified into the"off-target" class.

For a given point x, the expected value of y is

$$\begin{aligned}\mu(x) = E(y|x) &= p(y=1|x) - p(y=-1|x) \\ &= 2p(y=1|x) - 1\end{aligned}$$

Then, according to Bayes' Theorem,

$$\begin{aligned}p(y=-1|x) &= \frac{p(y=-1,x)}{p(x)} \\ &= \frac{p(x|-1)p(y=-1)}{p(x|-1)p(y=-1)+p(x|1)p(y=1)} \\ &= \frac{f(x)}{f(x)+f_0(x)}\end{aligned} \quad (1)$$

where we assume $p(y=1) = p(y=-1)$ for training data, which means in estimating $E(y|x)$ we use the same sample size for each class. Therefore, an estimate of the unknown density $f(x)$ is obtained as

$$\hat{f}(x) = \frac{1-\widehat{\mu}(x)}{1+\widehat{\mu}(x)} \times f_0(x), \quad (2)$$

where $f_0(x)$ is the known reference probability density function of the random data and $\hat{\mu}(x)$is learned from the supervised algorithm. Also, the odds are

$$\frac{p(y=-1|x)}{p(y=1|x)} = \frac{f(x)}{f_0(x)} \quad (3)$$

The assignment is determined by the value of $\hat{\mu}(x)$. A data x is assigned to the class with density $f(x)$ when

$$\widehat{\mu}(x) < v,$$

and the class with density $f_0(x)$ when

$$\widehat{\mu}(x) > v.$$

where v is a parameter that can used to adjust the error rates of the procedure.

Any supervised learner is a potential candidate to build the model. In our research, a Regularized Least Square Classifier (RLSC) (Cucker and Smale, 2001) is employed as the specific classifier. Squared error loss is used with a quadratic penalty term on the coefficients (from the standardization the intercept is zero). Radial basis functions are used at each observed point with common standard deviation. That is the mean of y is estimated from

$$\begin{aligned}\mu(x) &= \beta_0 + \sum_{j=1}^{n} \beta_j \exp\left(-\frac{1}{2}\|x - x_j\|^2/\sigma^2\right) \\ &= \beta_0 + \sum_{j=1}^{n} \beta_j K_\sigma(x, x_j)\end{aligned} \quad (4)$$

Also, let $\beta = (\beta_1, \ldots, \beta_n)$. The β_j are estimated from the penalized least squares criterion

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{n} \beta_j \exp\left(-\tfrac{1}{2}\|x - x_j\|^2/\sigma^2\right)\right)^2 + \gamma\|\beta\|^2 \tag{5}$$

where n is the total number of observations in the training data set. If the x's and y are standardized to mean zero then it can be shown that $\widehat{\beta}_0 = 0$. Also, let the matrix K denote the $n \times n$ matrix with (i,j)th element equal to $K_\sigma(x_i, x_j)$. Then for a fixed σ the solution for β is

$$\widehat{\beta} = (K + n\gamma I)^{-1}\mathbf{y} \tag{6}$$

and this is used to estimate $\mu(x)$.

3 CONTRIBUTORS TO A SIGNAL

In this section, a metric is developed to identify variables that contribute to a signal from SPC based upon artificial contrasts. Suppose there are p correlated variables $(x_1, x_2, \ldots, x_p)$. Let x^* be an observed data point that results in a signal from the control scheme. Define the set

$$L_k = \{x | x_i = x_i^*, i \neq k\}$$

There are several reasonable metrics for the contribution of variable x_k to the out-of-control signal. We use

$$\eta_k(x^*) = \max_{x \in L_k} \frac{\widehat{f}(x)}{\widehat{f}(x^*)} \tag{7}$$

This measures the change from $\widehat{f}(x)/\widehat{f}(x^*)$ that can be obtained from only a change to x_k. If $\eta_k(x^*)$ is small then x_k^* is not unusual. If $\eta_k(x^*)$ is large, then a substantial change can result from a change to x_k and x_k is considered to be an important contributor to the signal.

From (2) it can be shown that $\widehat{\mu}(x)$ is a monotone function of the estimated density ratio $\widehat{f}(x)/\widehat{f}_0(x)$. Therefore, the value $x_k \in L_k$ that maximizes the estimated density ratio also maximizes $\widehat{\mu}(x)$ over this same set. In the special case that $f_0(x)$ is a uniform density the value of $x_k \in L_k$ that maximizes $\widehat{\mu}(x)$ also maximizes $\widehat{f}(x)$ over this set. Consequently, $\eta_k(x^*)$ considers the change in estimated density that can be obtained from a change to x_k.

From (3) we have that η_k is the maximum odds ratio obtained over L_k

$$\eta_k(x^*) = \max_{x \in L_k} \frac{\hat{p}(y = -1|x)/\hat{p}(y = 1|x)}{\hat{p}(y = -1|x^*)/\hat{p}(y = 1|x^*)} \tag{8}$$

To compare values of $\eta_k(x^*)$ over k the denominator in (8) can be ignored and the numerator is a monotone function of $\hat{p}(y = -1|x)$. Consequently, the value in L_k that maximizes $\eta_k(x^*)$ is the one that maximizes $\hat{p}(y = -1|x)$. Therefore, the $\eta_k(x^*)$ metric is similar to one that scores the change in estimated probability of an in-control point.

A point that is unusual simultaneously in more than one variable, but *not* in either variable individually, is not well identified by this metric. That is, if x^* is unusual in the joint distribution of $(x_1, \ldots, x_k)$ for $k \leq p$, but not in the conditional marginal distribution of $f(x_i|x_j = x_j^*, i \neq j)$ then the metric is not sensitive. This implies that the point is unusual in a marginal distribution of more than one variable. Consequently, one can consider a two-dimensional set

$$L_{jk} = \{x | x_i = x_i^*, i \neq j, k\}$$

and a new metric

$$\eta_{jk}(x^*) = \max_{x \in L_{jk}} \frac{\widehat{f}(x)}{\widehat{f}(x^*)} \tag{9}$$

to investigate such points. This two-dimensional metric would be applied if none of the one-dimensional metrics $\eta_k(x^*)$ are unusual. Similarly, higher-dimensional metrics can be defined and applied as needed. The two-dimensional metric $\eta_{jk}(x^*)$ would maximize the the estimated density over x_j and x_k. It might use a gradient-based method or others heuristics to conduct the search. The objective is only to determine the pair of variables that generate large changes in the estimated density. The exact value of the maximum density is not needed. This permits large step sizes to be used in the search space. However, the focus of the work here is to use the one-dimensional metrics $\eta_k(x^*)$'s. Because the contribution analysis is only applied to a point which generates a signal, no information for the set of one-dimensional η_k's implies that a two-dimensional (or higher) metric needs to be explored. However, the one-dimensional η_k's are effective in many cases, and they provide a starting point for all cases.

3.1 Comparison with a Multivariate Normal Distribution

In this section, we assume the variables follow a multidimensional normal distribution. Under these assumptions, we can determine the theoretical form of the metric $\eta_k(x^*)$. Given the estimate of the unknown density $\widehat{f}(x)$, define x_0 as

$$x_0 = \text{argmax}_{x \in L_k} \widehat{f}(x)$$

For a multivariate normal density with mean vector μ and covariance matrix Σ

$$x_0 = \text{argmin}_{x \in L_k} (x - \mu)' \Sigma^{-1} (x - \mu)$$

Therefore, x_0 is the point in L_k at which Hotelling's statistic is minimized. Consequently, x_0 is the same

point used in (Runger et al., 1996) to define the contribution of variable x_k in the multivariate normal case. The use of the metric in (7) generalizes this previous result from a normal distribution to an arbitrary distribution.

4 ILLUSTRATIVE EXAMPLE

4.1 Learning the In-Control Boundary

To demonstrate that our method is an extension of the traditional method, first we assume that the in-control data follow a multivariate normal distribution. In the case of two variables, we capture a smooth, closed elliptical boundary. Figure 1 shows the boundary learned through an artificial contrast and a supervised learning method along with the boundary specified by Hotelling's statistic (Hotelling, 1947) for the in-control data.

The size of in-control training data is 400 and the size of uniform data is also 400. The in-control training data are generated from the two-dimensional normal distribution $\mathbf{X} = \mathbf{C} * \mathbf{Z}$ with covariance

$$\text{Cov}(\mathbf{X}) = \mathbf{C}\mathbf{C}' = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

and $\mathbf{Z}$ following two-dimensional joint standardized normal distribution with $\rho = 0$. The smoothing parameter for the classifier is $\gamma = 4/800$. The parameter for the kernel function is $\sigma = \sqrt{8}$. The out-of-control training data are generated from the reference distribution. There are four unusual points: A $(3, 0)$, B $(3, 1)$, C $(3, 2)$, and D $(3, 3)$.

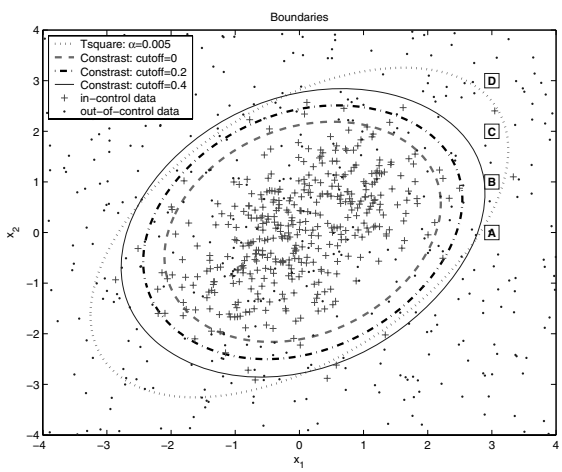


Figure 1: Learned Boundaries and Hotelling's Boundary.

Table 1: Type I error for In-control Data.

cut-off value	0	0.2	0.4
the training data	0.085	0.0325	0.015
the testing data	0.1	0.0525	0.025

Table 2: Type II error for Out-of-control Data with Different Shifted Means.

cut-off value	0	0.2	0.4
(1,0)	0.785	0.895	0.96
(1,1)	0.7275	0.8325	0.8975
(2,0)	0.4875	0.6125	0.7325
(2,2)	0.3225	0.45	0.565
(3,0)	0.1025	0.215	0.325
(3,3)	0.055	0.1025	0.185

Testing data sets are used to evaluate performance, that is, Type I error and Type II error of the classifier. They are generated from similar multivariate normal distributions with or without shifted means. Each testing data set has a sample size of 400.

Table 1 gives the Type I error for the training data and for the testing data whose mean is not shifted. It shows that the Type I error decreases when the cut-off value of the boundary increases. Table 2 gives the Type II error for the testing data with shifted mean. It shows that for a given shift, the Type II error increases when the cut-off value of the boundary increases. It also illustrates that, for a given cut-off value, the further the mean shifts from the in-control mean, the lower the Type II error.

4.2 Contribution Evaluation

The probability density function of the in-control data $f(x)$ is estimated by (2). For the normal distribution in Section 3.1 examples are provided in the cases of two-dimensions (Figure 2) and 30-dimensions (Figure 3).

For the case of two dimensions, Figure 1 shows 4 points at $(3, 0), (3, 1), (3, 2), (3, 3)$. The corresponding curves for $\widehat{f}(x)$ for each point are shown in Figure 4 through Figure 7. These figures show that the variable that would be considered to contribute to the signal for points $(3, 0)$ and $(3, 1)$ is identified by the corresponding curve. For point $(3, 2)$ the variable is not as clear and the curves are also ambiguous. For the point $(3, 3)$ both variables can be considered to the signal and this is indicated by the special case where all curve are similar. That is, no proper subset of variables is identified and this is an example where a higher-dimensional analysis (such as with $\eta_{jk}(x^*)$) is useful.

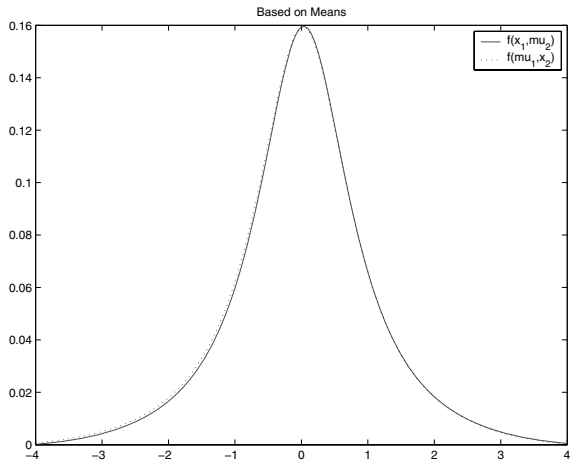


Figure 2: Density estimate for two dimensions.

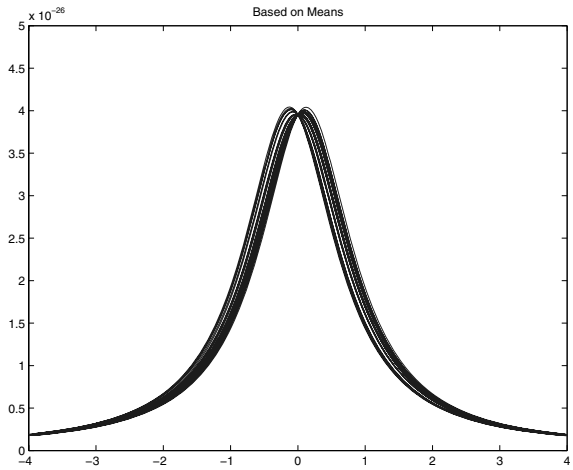


Figure 3: Density estimate for thirty dimensions.

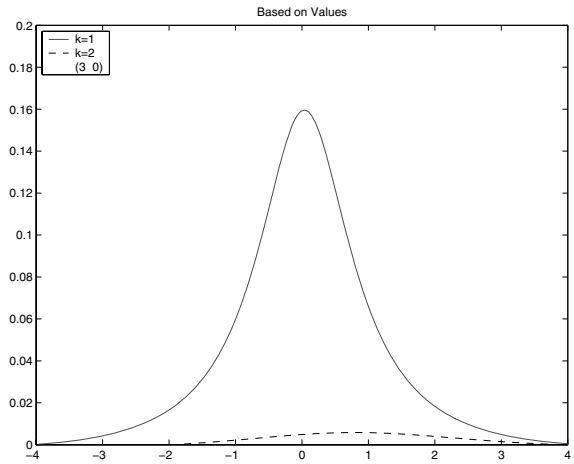


Figure 4: $f(x_1, 0)$ and $f(3, x_2)$.

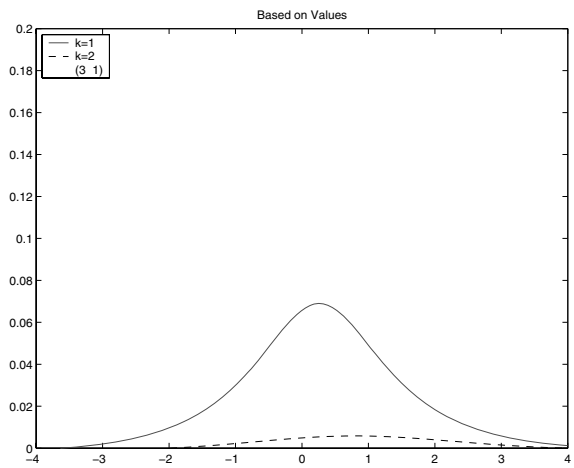


Figure 5: $f(x_1, 1)$ and $f(3, x_2)$.

4.3 Example in 30 Dimensions

For a higher dimensional example, consider $p = 30$ dimensions. Out-of-control points are generated and density curves are produced for each variable. These curves are proportional to the conditional density with all the other variables at the observed values. For $p = 30$ the size of in-control training data is 200 and the size of the uniform data is also 200. Curves for out-of-control points

$$A = (3, 0, \ldots, 0)$$

$$B = (3, -3, 0, \ldots, 0)$$

$$C = (3, 3, \ldots, 3)$$

are generated. For $p = 30$ dimensions the density curves are shown in Figure 8 through Figure 10. Note that the changes in density match the contributors to an unusual point. Note that for point C the density metric does not indicate any subset of variables as contributors. This is a special case and such a graph implies that all variables contribute to the signal from the chart because these graphs are only generated after a signal from a control has been generated. Such a special case is also distinguished from cases where only a proper subset of variables contribute to the signal.

For the particular case of $p = 30$ dimensions, values of $\eta_k(x_i)$ are calculated for these points and $k = 1, \ldots, 30$ in Figure 11 through Figure 13. The results indicate the this metric can identify variables that contribute to the signal. For point C similar comments made for the density curves apply here. The metric does not indicate any subset of variables as contributors. This is a special case and such a graph implies that all variables contribute to the signal from the chart.

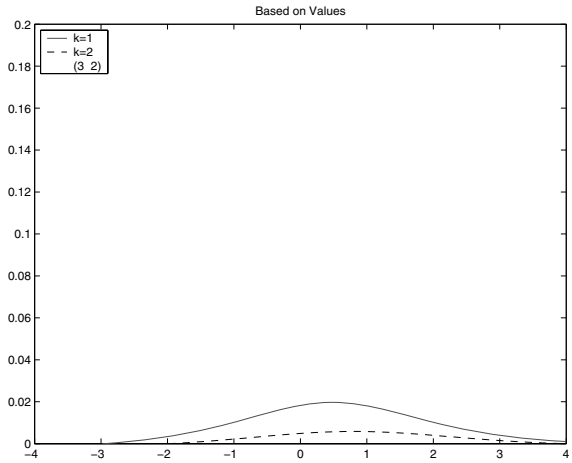


Figure 6: $f(x_1, 2)$ and $f(3, x_2)$.

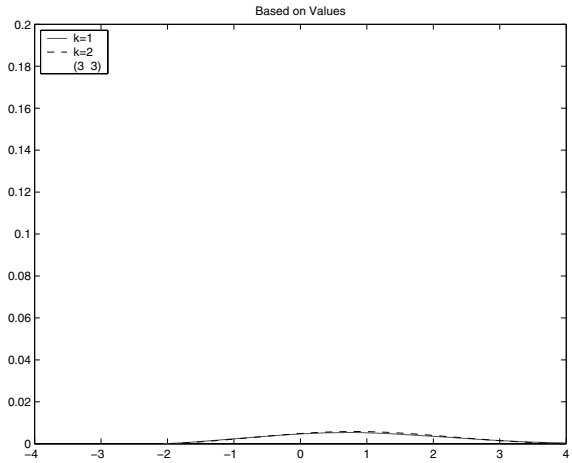


Figure 7: $f(x_1, 3)$ and $f(3, x_2)$.

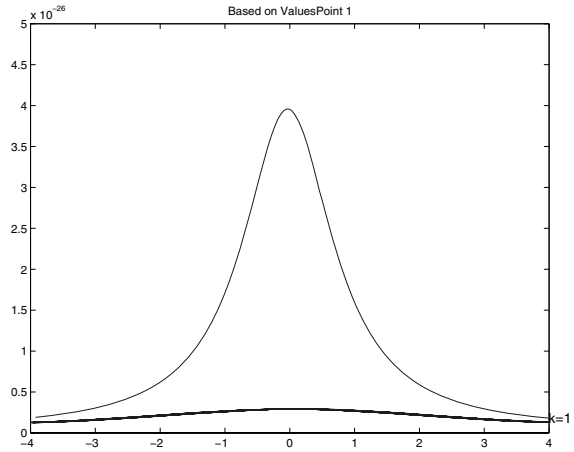


Figure 8: Density $f(x)$ as a function of x_k for $k = 1, 2, \ldots, 30$ for Point A.

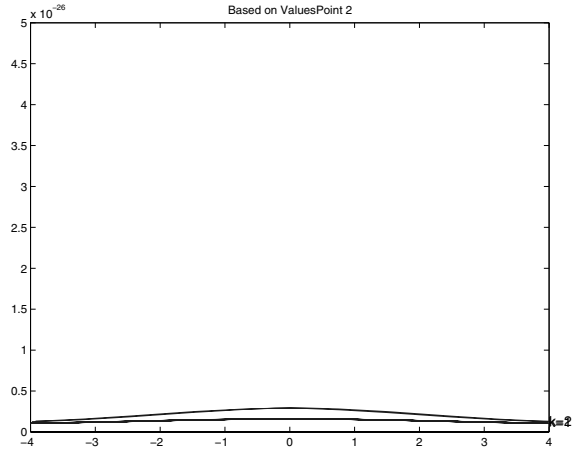


Figure 9: Density $f(x)$ as a function of x_k for $k = 1, 2, \ldots, 30$ for Point B.

5 MANUFACTURING EXAMPLE

The data set was from a real industrial process. There are 228 samples in total. To illustrate our problem, we use two variables. Here, Hotelling T^2 is employed to find out in-control data. The mean vector and covariance matrix are estimated from the whole data set and T^2 follows a χ^2 distribution with two degrees of freedom. The false alarm, α, is set as 0.05 in order to screen out unusual data. Figure 14 displays the Hotelling T^2 for each observation. From the results, we obtain 219 in-control data points that are used as the training data.

Figure 15 shows the learned boundaries with different cut-off values and the Hotelling T^2 boundary with α being 0.005. The learned boundary well captures the characteristic of the distribution of the in-control data. We select the learned boundary with cut-off $v = 0.4$ as the decision boundary and obtain three unusual points: Point 1, 2, and 4. The metric is applied to Point 2 and 4 and Table 3 and it demonstrates η values for each dimension for each point. Figure 16 and Figure 17 demonstrate $f(x_1, x_2)$ when as functions of x_1 and x_2 for Point 2 and 4, respectively. For Point 2, η_1 is significantly larger than η_2 so the first variable contributes to the out-of-control signal. For Point 4, η_1 and η_2 are close so both variables contributes to the out-of-control signal.

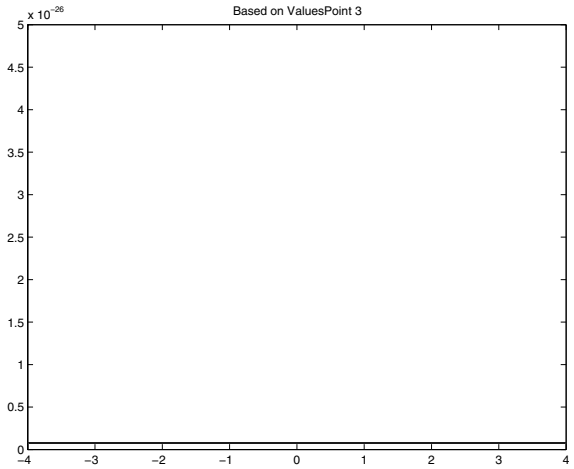


Figure 10: Density $f(x)$ as a function of x_k for $k = 1, 2, \ldots, 30$ for Point C.

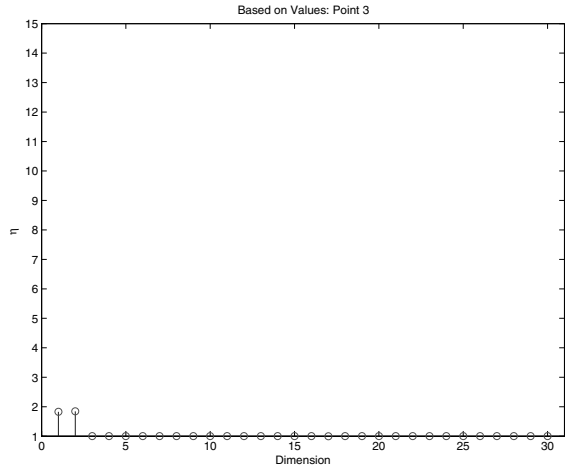


Figure 12: Contributor metric η_k for variables $k = 1, 2, \ldots, 30$ for Point B.

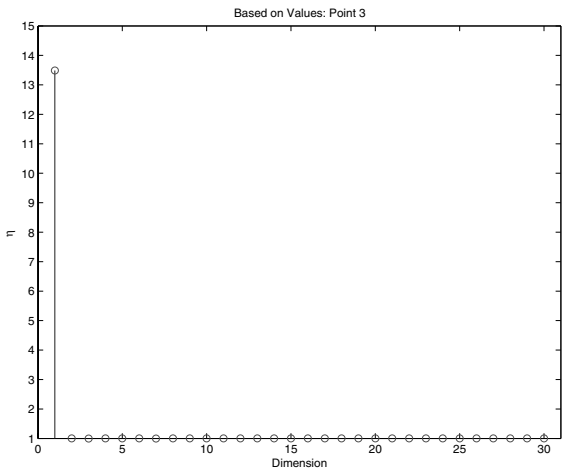


Figure 11: Contributor metric η_k for variables $k = 1, 2, \ldots, 30$ for Point A.

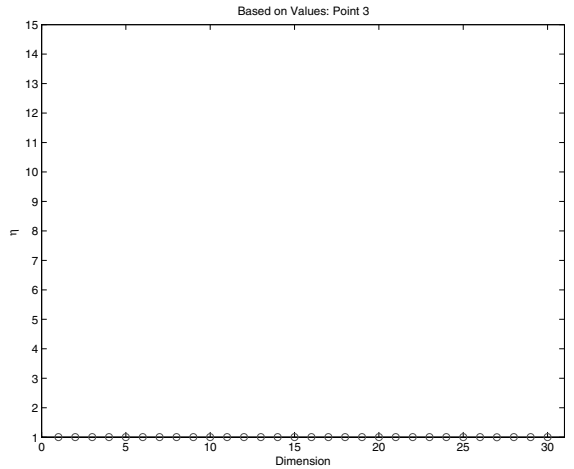


Figure 13: Contributor metric η_k for variables $k = 1, 2, \ldots, 30$ for Point C.

6 CONCLUSIONS

A supervised method to learn normal operating conditions provides a general solution to monitor systems of many types in many disciplines. In addition to the decision rule it is important to be able to interrogate a signal to determine the variables that contribute to it. This facilitates an actionable response to a signal from decision rule used to monitor the process. In this paper, contributors to a multivariate SPC signal are identified from the same function that is learned to define the decision rule. The approach is computationally and conceptually simple. It was shown that the method generalizes a traditional approach for traditional multivariate normal theory. Examples show that the method effectively reproduces solutions for known cases, yet it generalizes to a broader class of problems. The one-dimensional metric used here would always be a starting point for such a contribution analysis. Future work is planned to extend the metric to two- and higher-dimensions to better diagnose contributors for cases in which the one-dimensional solution is not adequate.

Table 3: η for Point 2 and 4.

	η_1	η_2
Point 2	16.791	1.0001
Point 4	3.6737	1.6549

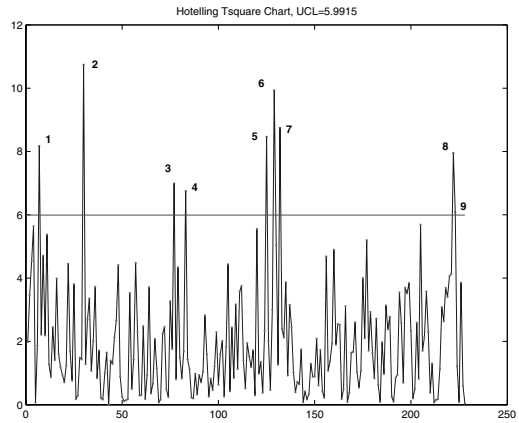


Figure 14: Hotelling T^2.

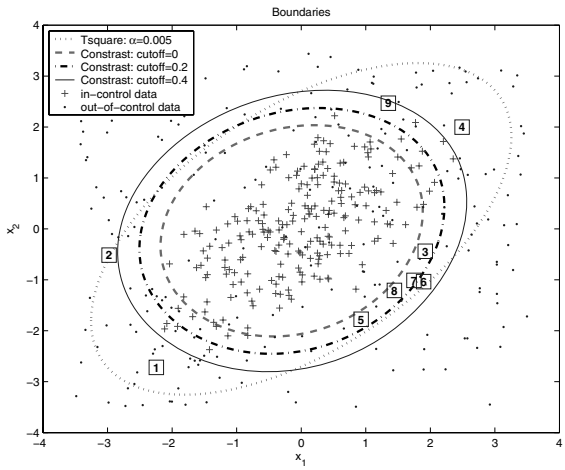


Figure 15: Learned Boundaries and Hotelling T^2.

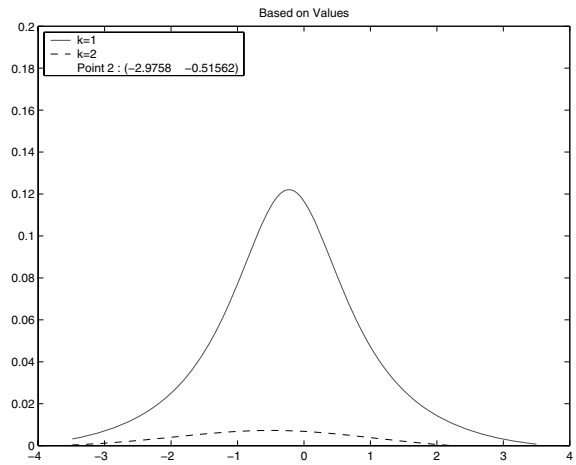


Figure 16: Density $f(x)$ as a function of x_1 and x_2 for Point 2.

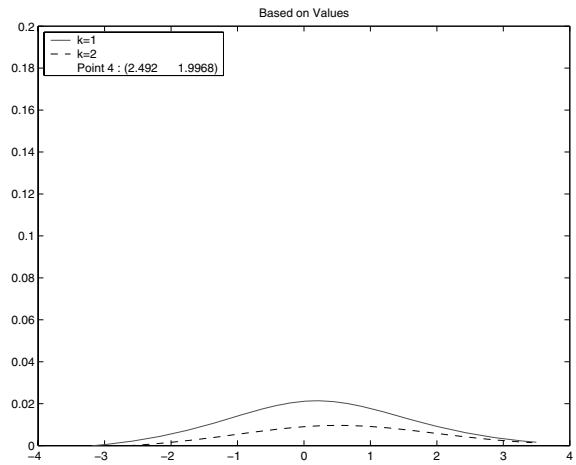


Figure 17: Density $f(x)$ as a function of x_1 and x_2 for Point 4.

REFERENCES

Alt, F. B. (1985). Multivariate quality control. In Kotz, S., Johnson, N. L., and Read, C. R., editors, *Encyclopedia of Statistical Sciences*, pages 110–122. John Wiley and Sons, New York.

Chua, M. and Montgomery, D. C. (1992). Investigation and characterization of a control scheme for multivariate quality control. *Quality and Reliability Engineering International*, 8:37–44.

Cucker, F. and Smale, S. (2001). On the mathematical foundations of learning. *Bulletin of AMS*, 39:1–49.

Doganaksay, N., Faltin, F. W., and Tucker, W. T. (1991). Identification of out-of-control quality characteristics in a multivariate manufacturing environment. *Communications in Statistics-Theory and Methods*, 20:2775–2790.

Hotelling, H. (1947). Multivariate quality control-illustrated by the air testing of sample bombsights. In Eisenhart, C., Hastay, M., and Wallis, W., editors, *Techniques of Statistical Analysis*, pages 111–184. McGraw-Hill, New York.

Hwang, W., Runger, G., and Tuv, E. (2004). Multivariate statistical process control with artificial contrasts. under review.

Mason, R. L., Tracy, N. D., and Young, J. C. (1995). Decomposition of T^2 for multivariate control chart interpretation. *Journal of Quality Technology*, 27:99–108.

Murphy, B. J. (1987). Selecting out-of-control variables with T^2 multivariate quality control procedures. *The Statistician*, 36:571–583.

Rencher (1993). The contribution of individual variables to hotelling's T^2, wilks' Λ, and R^2. *Biometrics*, 49:479–489.

Runger, G. C., Alt, F. B., and Montgomery, D. C. (1996). Contributors to a multivariate control chart signal. *Communications in Statistics - Theory and Methods*, 25:2203–2213.

REAL-TIME TIME-OPTIMAL CONTROL FOR A NONLINEAR CONTAINER CRANE USING A NEURAL NETWORK

T. J. J. van den Boom, J. B. Klaassens and R. Meiland
Delft Center for Systems and Control
Mekelweg 2, 2628 CD Delft, The Netherlands
T.J.J.vandenBoom@dcsc.tudelft.nl

Keywords: Time-optimal crane control, Nonlinear Model Predictive Control, Optimization, binary search algorithm, neural networks, Bayesian regularization.

Abstract: This paper considers time-optimal control for a container crane based on a Model Predictive Control approach. The model we use is nonlinear and it is planar, i.e. we only consider the swing (not the skew) and we take constraints on the input signal into consideration. Since the time required for the optimization makes time-optimal not suitable for fast systems and/or complex systems, such as the crane system we consider, we propose an off-line computation of the control law by using a neural network. After the neural network has been trained off-line, it can then be used in an on-line mode as a feedback control strategy.

1 INTRODUCTION

The need for fast transport of containers from quay to ship and from ship to quay, is growing more and more. Since ships and their container capacity grow larger, a more time efficient manner of (un)loading containers is required. Shipping companies focus on maximizing the sailing hours and reducing the hours spent in port. A longer stay in port will eliminate the profit gained at sea for the large vessels and can hardly be considered as an option.

Much research has been done on the crane modelling and control (Marttinen et al., 1990; Fliess et al., 1991; Hämäläinen et al., 1995; Bartolini et al., 2002; Giua et al., 1999) however most models are linearized. In this paper we study time-optimal control for a container crane using a nonlinear model. The drawback of time-optimal control, in the presence of constraints, is its demand with respect to computational complexity. This doesn't make time-optimal control suitable for fast systems, such as the crane system. To overcome this problem a neural network can be used. It can be trained off-line to 'learn' the control law obtained by the time-optimal controller. After the neural network has been trained off-line, it can then be used in an on-line mode as a feedback control strategy. In Nonlinear Model Predictive Control (MPC) an off-line computation of the control law using a feed-forward neural network was already proposed by (Parisini and Zoppoli, 1995). The off-line approach was also followed in (Pottman and Seborg, 1997), where a radial basis function network was used to 'memorize' the control actions. In this paper we extend these ideas to time-optimal control.

Section 2 describes the continuous-time model of the crane and the conversion from the continuous-time model to a discrete-time model. Section 3 discusses time-optimal control. Section 4 gives an outline of a feedforward network and discuss the best architecture of the neural network with respect to the provided training data. Section 5 gives conclusions about how well the time-optimal controller performs in real time.

2 CRANE MODEL

A dynamical crane model is presented in this section. A schematic picture of the container crane is shown in Figure 1. The container is picked up by a spreader, which is connected to a trolley by the hoisting cables. One drive in controlling the motion of the trolley and another drive is controlling the hoisting mechanism. The electrical machines produce the force F_T acting on the trolley and the hoisting force F_H and providing the motion of the load. The dynamics of the electrical motors is not included in the model. The combination of the trolley and the container is identified as a

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 79–84.

two-sided pendulum. The elastic deformation in the cables and the crane construction is neglected. The load (spreader and container) is presented as a mass m_c hanging on a massless rope. Friction in the system is neglected. Only the swing of the container is considered while other motions like skew are not taken into account. Further, the sensors are supposed to be ideal without noise. The influence of wind and sensor noise can be included in the model as disturbances.

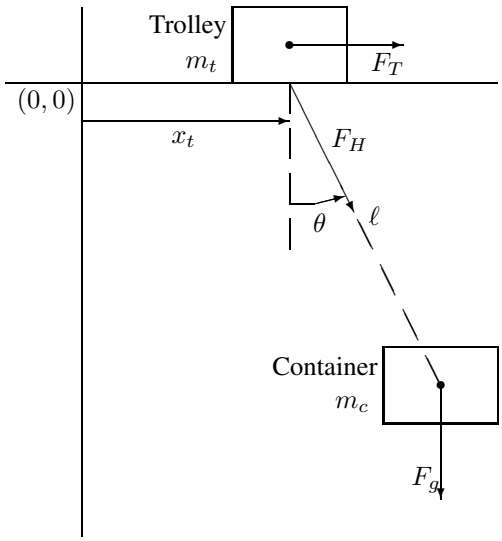


Figure 1: Jumbo Container Crane model.

The continuous time model is presented by the following equations:

$$\ddot{x}_t = \frac{m_c G_h g \sin\theta\cos\theta + m_c G_h l\dot{\theta}^2 \sin\theta}{(m_c+G_h)(m_t+G_t)+G_h m_c(1-\cos^2\theta)} + \frac{F_T(m_c+G_h) - m_c F_H \sin\theta}{(m_c+G_h)(m_t+G_t)+G_h m_c(1-\cos^2\theta)} \quad (1)$$

$$\ddot{\theta} = \frac{-\ddot{x}_t\cos\theta - 2\dot{l}\dot{\theta} - g\sin\theta}{l} \quad (2)$$

$$\ddot{l} = \frac{F_H - m_c\ddot{x}_t\sin\theta + m_c l\dot{\theta}^2 + m_c g\cos\theta}{m_c+G_h} \quad (3)$$

where x_t is position of the trolley, θ is the swing angle, l is the length of the cable, m_c is the container mass, m_t is the trolley mass, G_t is the virtual trolley motor mass, G_h is the virtual hoisting motor mass, F_T transversal force and F_H is the hoisting force.

By defining the following state and control signals

$$z = \begin{bmatrix} x_t \\ \dot{x}_t \\ \theta \\ \dot{\theta} \\ l \\ \dot{l} \end{bmatrix}, \quad u = \begin{bmatrix} F_T \\ (F_H - F_{H0}) \end{bmatrix},$$

where $F_{H0} = -m_c g$ is used to compensate gravity of the container, we obtain the continuous dynamic system in the following form:

$$\dot{z}(t) = f(z(t), u(t)) \quad (4)$$

Discrete-time model

Since the controller we will use is discrete, a discrete-time model is needed. We have chosen Euler's method because it is a fast method. The Euler approximation is given by:

$$z(k+1) = z(k) + T \cdot f(k, z(k), u(k)) \quad (5)$$

where the integration interval $\triangle t$ is the sampling time T.

3 TIME-OPTIMAL CONTROL

In this paper we consider time-optimal control for the crane. Some papers recommend the planning of a time-optimal trajectory and use this as a reference path for the container to follow (Gao and Chen, 1997; Kiss et al., 2000; Klaassens et al., 1999). We have chosen not to determine a pre-calculated time-optimal path and subsequently use this as a reference, in stead we calculate the time-optimal path using a two step iteration. In a first step we propose a specific time interval N and evaluate if there is a feasible solution that brings the container to the desired end-position $(x_{c,\mathrm{des}}, y_{c,\mathrm{des}})$ within the proposed time interval N. In the second step we enlarge the time interval if no feasible solution exists, or we shrink the interval if there is a feasible solution. We iterate over step 1 and step 2 until we find the smallest time interval N_{opt} for which there exists a feasible solution.

First step:
To decide, within the first step, whether there is a feasible solution for a given time interval N, we minimize a Model Predictive Control (MPC) type cost-criterion $J(u,k)$ with respect to the time interval constraint and additional constraint for a smooth operation of the crane. In MPC we consider the future evolution of the system over a given prediction period $[k+1, k+N_p]$, which is characterized by the prediction horizon N_p (which is much larger than the

proposed time interval), and where k is the current sample step. For the system Eq. (5) we can make an estimate $\hat{z}(k+j)$ of the output at sample step $k+j$ based on the state $z(k)$ at step k and the future input sequence $u(k), u(k+1), \ldots, u(k+j-1)$. Using successive substitution, we obtain an expression of the form

$$\hat{z}(k+j) = F_j(z(k), u(k), u(k+1), \ldots, u(k+j-1))$$

for $j = 1, \ldots, N_p$. If we define the vectors

$$\tilde{u}(k) = \left[u^{\mathrm{T}}(k) \quad \ldots \quad u^{\mathrm{T}}(k+N_p-1) \right]^{\mathrm{T}} \tag{6}$$

$$\tilde{z}(k) = \left[\hat{z}(k+1) \quad \ldots \quad \hat{z}(k+N_p) \right]^{\mathrm{T}} , \tag{7}$$

we obtain the following prediction equation:

$$\tilde{z}(k) = \tilde{F}(z(k), \tilde{u}(k)) \ . \tag{8}$$

The cost criterion $J(u,k)$ used in MPC reflects the reference tracking error ($J_{\text{out}}(\tilde{u},k)$) and the control effort ($J_{\text{in}}(\tilde{u},k)$):

$$\begin{aligned} J(\tilde{u},k) &= J_{\text{out}}(\tilde{u},k)(k) + \lambda J_{\text{in}}(\tilde{u},k)(k) \\ &= \sum_{j=1}^{N_p} |\hat{x}_c(k+j) - x_{c,\text{des}}|^2 + |\hat{y}_c(k+j) - y_{c,\text{des}}|^2 \\ &\quad + \lambda |u(k+j-1)|^2 \end{aligned} \tag{9}$$

where $x_c = \hat{z}_1 + \hat{z}_5 \sin \hat{z}_3$ is the x-position of the container, $y_c = \hat{z}_5 \cos \hat{z}_3$ is the y-position of the container, and λ is a nonnegative integer. From the above it is clear that $J(k)$ is a function of $\tilde{z}(k)$ and $\tilde{u}(k)$, and so is a function of $z(k)$ and $\tilde{u}(k)$.

In practical situations, there will be constraints on the input forces applied to the crane:

$$\begin{array}{rcl} -F_{T\,\max} \leq & u_1 & \leq F_{T\,\max}, \\ F_{H\,\max} - F_{H0} \leq & u_2 & \leq -F_{H0}. \end{array} \tag{10}$$

where, because of the sign of F_H, we have $F_{H\,\max} < 0$ and $F_{H0} = -m_c g < 0$. Further we have the time interval constraints that

$$\begin{array}{rcl} |\hat{x}_c(N+i) - x_{c,\text{des}}| & \leq & \epsilon_x, \ \ i \geq 0 \\ |\hat{y}_c(N+i) - y_{c,\text{des}}| & \leq & \epsilon_y, \ \ i \geq 0 \end{array} \tag{11}$$

which means that at the end of the time interval N the container must be at its destination with a desired precision ϵ_x and ϵ_y, respectively.

Consider the constrained optimization problem to find at time step k a $\tilde{u}(k)$ where:

$$\tilde{u}^*(k) = \arg\min_{\tilde{u}} J(\tilde{u},k)$$

subject to Eq. (10) and (11). Note that the above optimization is a nonlinear optimization with constraints. To reduce calculation time for the optimization we can rewrite the constrained optimization problem into an unconstrained optimization problem by introducing auxiliary input variables for the force constraints and penalty functions to account for the time interval constraint. For the force constraints we consider the auxiliary inputs v_1 and v_2:

$$u_1 = \alpha \arctan\left(\frac{v_1}{\alpha}\right)$$

$$u_2 = \begin{cases} \beta \arctan\left(\frac{v_2}{\beta}\right), & v_2 < 0 \\ v_2, & 0 < v_2 < (\beta\pi/2) \\ \gamma + \beta \arctan\left(\frac{v_2-\gamma}{\beta}\right), & v_2 > (\beta\pi/2) \end{cases}$$

where $\alpha = 2F_{T\,\max}/\pi$, $\beta = 2(F_{H\,\max} - F_{H0})/\pi$ and $\gamma = F_{H\,\max} - 2F_{H0}$. Note that for all $v_1, v_2 \in \mathbb{R}$ input force constraints Eq. (10) will be satisfied.

For the time interval constraints we define the penalty function:

$$\begin{aligned} J_{\text{pen}}(\tilde{u},k) &= \sum_{j=N-k}^{N_p} \mu |\hat{x}_c(k+j) - x_{c,\text{des}}|^2 \\ &\quad + \mu |\hat{y}_c(k+j) - y_{c,\text{des}}|^2 \end{aligned} \tag{12}$$

where $\mu \gg 1$. Beyond the time interval (so for $k+j \geq N$) the influence of any deviation from the desired end point is large and the container position and speed must then be very accurate.

Instead of the constrained optimization problem we have now recast the problem as an unconstrained optimization problem at time step k:

$$\tilde{v}^*(k) = \arg\min_{\tilde{v}} J(\tilde{u}(\tilde{v}),k) + J_{\text{pen}}(\tilde{u}(\tilde{v}),k)$$

where

$$\tilde{v}(k) = \left[v^{\mathrm{T}}(k) \quad \ldots \quad v^{\mathrm{T}}(k+N_p-1) \right]^{\mathrm{T}}$$

For the optimization we use an iterative optimization algorithm where in each iteration step we first select a search direction and then we perform a line search, i.e., an optimization along the search direction. The search direction is according the Broyden-Fletcher-Goldfarb-Shanno method and for the line search we have chosen a mixed quadratic and cubic polynomial method.

Second step:
In the first step we have a constant penalty function shifting point N, which has to be chosen differently for every different initial state and steady state. When we have chosen a value for N for a certain initial state and steady state such that the states converge, we can lower the value of N. On the other hand, when we have chosen a value for N for a certain initial state and steady state such that the states do not converge within the allowed region, we have to increase the value of N. When we have found the optimal value $N = N_{\text{opt}}$ if for N there exists a feasible solution, and reduction of N will lead to an infeasible problem. In other words, The determination of N_{opt} has

become a feasibility study. To determine the optimal value N_{opt} for each different initial state z_0 and steady state $(x_{c,\text{des}}, y_{c,\text{des}})$ in an efficient way, we have implemented a simple binary search algorithm.

4 NEURAL NETWORK

Since the time required for the optimization makes time-optimal control not suitable for fast systems, we propose an off-line computation of the control law using a neural network. We assume the existence of a function that maps the state to the optimal control action, and this function is continuous. Continuous functions can be approximated to any degree of accuracy on a given compact set by feedforward neural networks based on sigmoidal functions, provided that the number of neural units is sufficiently large. However, this assumption is only valid if the solution to the optimization problem is unique. After the neural network controller has been constructed off-line, it can then be used in an on-line mode as a feedback control strategy. Because the network will always be an approximation, it cannot be guaranteed that constraints are not violated. However, input constraints, which are the only constraints we consider, can always be satisfied by limiting the output of the network.

Training of the neural network

We have covered the workspace of the crane as can be seen in Table 1. We have considered all initial

Table 1: Values of x_0 and x_{des}.

x_{t_0}	=	0	[m]
$x_{t,\text{des}}$	=	$\{0, 5, 10, \ldots, 60\}$	[m]
l_0	=	$\{5, 10, 15, \ldots, 50\}$	[m]
l_{des}	=	$\{5, 10, 15, \ldots, 50\}$	[m]

speeds zero, i.e. $\dot{x}_{t_0}, \dot{\theta}_0, \dot{l}_0$ as well as the swing angle θ_0 are zero. The initial state for the trolley, x_{t_0}, is always zero, and the steady state is within the range $0 \leq x_{t,\text{des}} \leq 60$ m , with steps of 5 m. The dynamical behavior of the crane depends on the distance of the trolley travelling $x_t - x_{t,\text{des}}$ and not on its position. This explains why we only consider $x_{t_0} = 0$ m.

We don't consider simulations where we start and end in the same states, or in other words, where we stay in equilibrium. Thus the total amount of different combinations of initial states x_0 and steady states x_{des} is $13 \times 10 \times 10 - 10 = 1290$.

It is of utmost importance to keep the number of inputs and outputs of the neural network as low as possible. This to avoid unnecessary complexity with respect to the architecture of the neural network. Notice that most of the steady states we use for the control problem, are always zero and can be disregarded for the input signal of the neural network. The only exceptions are $x_{t,\text{des}}$ and l_{des}. Furthermore, we can reduce the number of inputs by taking the distance of the trolley travelling $x_t - x_{t,\text{des}}$, while still providing the same dynamic behavior. We cannot reduce the number of the outputs, hence for the minimum number of inputs (z) and outputs (y) we have:

$$z = \begin{bmatrix} x_t - x_{t,\text{des}} \\ \dot{x}_t \\ \theta \\ \dot{\theta} \\ l \\ \dot{l} \\ l_{\text{des}} \end{bmatrix}, \; u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

We can reduce the dimension of the input vector even more by using principal component analysis as a preprocessing strategy. We eliminate those principal components which contribute less than 2 percent. The result is that the total number of inputs now is 6 in stead of 7.

We have trained the neural network off-line with the Levenberg-Marquardt algorithm. We have used one hidden layer and we have used Bayesian regularization to determine the optimal setting of hidden neurons. For more detail about Bayesian regularization we refer to (Mackay, 1992) and (Foresee and Hagan, 1997).

For the results we refer to Table 2 where m_1 denotes the number of neurons of the first (and only) hidden layer, $\mathcal{E}_{Tr}$, $\mathcal{E}_{Tst}$ and $\mathcal{E}_{Val}$ denote the sum of squared errors on the training subset, test subset and on the validation subset respectively. The sum of squares on the error weights is denotes by $\mathcal{E}_w$ and W_{eff} is the effective number of parameters.

Table 2: Bayesian regularization results for a 6-m_1-2 feedforward network.

m_1	$\mathcal{E}_{Tr}$	$\mathcal{E}_{Tst}$	$\mathcal{E}_{Val}$	$\mathcal{E}_w$	W_{eff}
5	42318	26759	14182	176	46.9
10	34463	29379	11568	226	90.6
20	24796	32502	8425	2164	180
30	24318	32819	8270	1219	268
40	21636	33573	7411	1099	357
50	18726	34617	6420	2270	445
60	19830	34152	6831	813	535
70	3462	7315	1424	1453	618
80	3599	7350	1473	828	704
90	3337	7459	1409	1232	793
100	3404	7473	1459	923	875
110	3225	7371	1401	1100	964
120	3237	7401	1437	1005	1046
130	3512	7281	1415	982	977

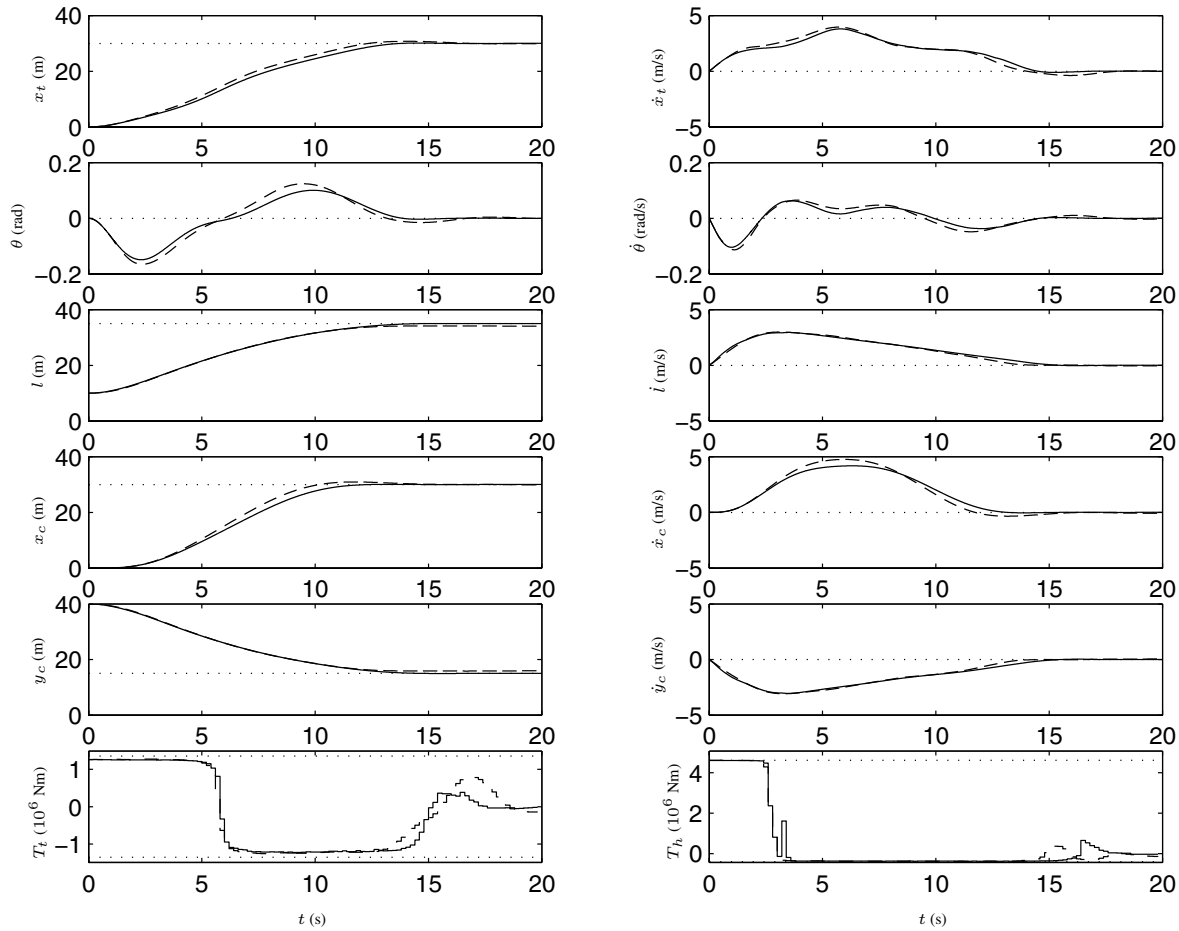


Figure 2: Comparison of simulation results between the time-optimal controller (solid line) and the neural network approximation (dashed line).

We have tested the neural network controller as a feedback control strategy in an online mode. Figure 2 shows a comparison between the neural network controller and the time-optimal controller where the dashed line denotes the simulation for the time-optimal controller and the solid line denotes the neural network simulation. The result seems satisfactory. The total cpu time of the neural network was 2.5 s and the total cpu time of the time-optimal controller was 17 minutes and 58 seconds. The neural network controller can easily be implemented in an online mode as a feedback control strategy.

5 DISCUSSION

In this paper an implementation of a time-optimal controller for a planar crane system is presented, based on an MPC approach. A nonlinear state space system was used for a model and we have implemented on the inputs. We have trained a neural network off-line with the training data obtained from the time-optimal controller. We have used Bayesian regularization to determine the optimal settings of the total number of hidden neurons. The trained neural network can be used in an online feedback control strategy.

In future research we will search methods to obtain the data, necessary for training the neural networks, in an efficient way, and to avoid redundancy. Further we will include the skew motion of the container, and introduce trajectory constraints to prevent collision of the container with other objects.

REFERENCES

Bartolini, G., Pisano, A., and Usai, E. (2002). Second-order sliding-mode control of container cranes. *Automatica*, 38:1783–1790.

Fliess, M., Lévine, J., and Rouchon, P. (1991). A simplified approach of crane control via a generalized state-

space model. Proceedings of the 30th Conference on Decision and Control, Brighton, England.

Foresee, F. and Hagan, M. (1997). Gauss-newton approximation to bayesian learning. *Proceedings of the 1997 International Joint Conference on Neural Networks*, pages 1930–1935.

Gao, J. and Chen, D. (1997). Learning control of an overhead crane for obstacle avoidance. Proceedings of the American Control Conference, Albuquerque, New Mexico.

Giua, A., Seatzu, C., and Usai, G. (1999). Oberserver-controller design for cranes via lyapunov equivalence. *Automatica*, 35:669–678.

Hämäläinen, J., Marttinen, A., Baharova, L., and Virkkunen, J. (1995). Optimal path planning for a trolley crane: fast and smooth transfer of load. *IEEE Proc.-Control Theory Appl.*, 142(1):51–57.

Kiss, B., Lévine, J., and Mullhaupt, P. (2000). Control of a reduced size model of us navy crane using only motor position sensors. In: *Nonlinear Control in the Year 2000*, Editor: Isidori, A., F. Lamnabhi-Lagarrigue and W. Respondek. Springer, New York, 2000, Vol.2. pp. 1–12.

Klaassens, J., Honderd, G., Azzouzi, A. E., Cheok, K. C., and Smid, G. (1999). 3d modeling visualization for studying control of the jumbo container crane. *Proceedings of the American Control Conference, San Diego, California*, pages 1754–1758.

Mackay, D. (1992). Bayesian interpolation. *Neural Computation*, 4(3):415–447.

Marttinen, A., Virkkunen, J., and Salminen, R. (1990). Control study with a pilot crane. *IEEE Transactions on education*, 33(3):298–305.

Parisini, T. and Zoppoli, R. (1995). A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 31(10):1443–1451.

Pottman, M. and Seborg, D. (1997). A nonlinear predictive control strategy based on radial basis function models. *Comp. Chem. Eng.*, 21(9):965–980.

PART 2

Robotics and Automation

IMAGE-BASED AND INTRINSIC-FREE VISUAL NAVIGATION OF A MOBILE ROBOT DEFINED AS A GLOBAL VISUAL SERVOING TASK

C. Pérez, N. García-Aracil, J. M. Azorín, J. M. Sabater[2],
L. Navarro and R. Saltarén[1]
[1]*Departamento de Automática, Electrónica e Informática Industrial Universidad Politécnica de Madrid.*
[2]*Dept. Ingeniería de Sistemas Industriales. Universidad Miguel Hernández.*
Avd. de la Universidad s/n. Edif. Torreblanca. 03202 Elche, Spain
Email: nicolas.garcia@umh.es

Keywords: Visual servoing, mobile robot navigation, continuous path control.

Abstract: The new contribution of this paper is the definition of the visual navigation as a global visual control task which implies continuity problems produced by the changes of visibility of image features during the navigation. A new smooth task function is proposed and a continuous control law is obtained by imposing the exponential decrease of this task function to zero. Finally, the visual servoing techniques used to carry out the navigation are the image-based and the intrinsic-free approaches. Both are independent of calibration errors which is very useful since it is so difficult to get a good calibration in this kind of systems. Also, the second technique allows us to control the camera in spite of the variation of its intrinsic parameters. So, it is possible to modify the zoom of the camera, for instance to get more details, and drive the camera to its reference position at the same time. An exhaustive number of experiments using virtual reality worlds to simulate a typical indoor environment have been carried out.

1 INTRODUCTION

Image-based visual servoing approach is now a well known control framework (Hutchinson et al., 1996). A new visual servoing approach, which allows to control a camera with changes in its intrinsic parameters, has been published in the last years (Malis and Cipolla, 2000; Malis, 2002c). In both approaches, the reference image corresponding to a desired position of the robot is generally acquired first (during an off-line step), and some image features extracted. Features extracted from the initial image or invariant features calculated from them are used with those obtained from the desired one to drive back the robot to its reference position.

The framework for robot navigation proposed is based on pre-recorded image features obtained during a training walk. Then, we want that the mobile robot repeat the same walk by means of image-based and intrinsic-free visual servoing techniques. The main contribution of this paper are the definition of the visual navigation as a global visual control task. It implies continuity problems produced by the changes of visibility of image features during the navigation and the computing of a continuous control law associated to it.

According to our knowledge, the approximation proposed to the navigation is totally different and new in the way of dealing with the features which go in/out of the image plane during the path and similar to some references (Matsumoto et al., 1996) in the way of specifying the path to be followed by the robot.

2 AUTONOMOUS NAVIGATION USING VISUAL SERVOING TECHNIQUES

The strategy of the navigation method used in this paper is shown in Figure 1. The key idea of this method is to divide the autonomous navigation in two stages: the first one is the *training step* and the second one is the *autonomous navigation step*. During the *training step*, the robot is human commanded via radio link or whatever interface and every sample time the robot acquires an image, computes the features and stores them in memory. Then, from near its initial position, the robot repeat the same walk using the reference features acquired during the *training step*.

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 87–93

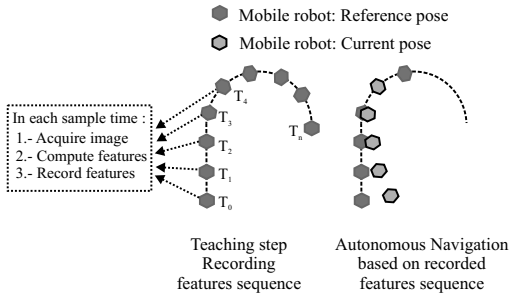


Figure 1: The strategy of the navigation method implemented. First a training step and the autonomous navigation.

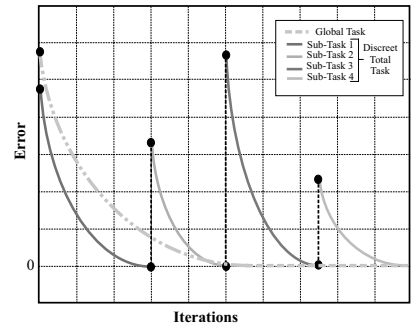


Figure 2: Navigation as a global task vs discretization of the navigation task.

2.1 Control Law

As it was mentioned in Section 1, image-based and intrinsic-free visual servoing approaches (Hutchinson et al., 1996; Malis and Cipolla, 2000) was used to develop the autonomous navigation of the robot. Both approaches are based on the selection of a set $\mathbf{s}$ of visual features or a set $\mathbf{q}$ of invariant features that has to reach a desired value $\mathbf{s}^*$ or $\mathbf{q}^*$. Usually, $\mathbf{s}$ is composed of the image coordinates of several points belonging to the considered target and $\mathbf{q}$ is computed as the projection of $\mathbf{s}$ in the invariant space calculated previously. In the case of our navigation method, $\mathbf{s}^*$ or $\mathbf{q}^*$ is variable with time since in each sample time the reference features is updated with the desired trajectory of $\mathbf{s}$ or $\mathbf{q}$ stored in the robot memory in order to indicate the path to be followed by the robot.

To simplify in this section, the formulation presented is only referred to image-based visual servoing. All the formulation of this section can be applied directly to the invariant visual servoing approach changing $\mathbf{s}$ by $\mathbf{q}$. The visual task function (Samson et al., 1991) is defined as the regulation of an global error function instead of a set of discrete error functions (Figure 2):

$$\mathbf{e} = \mathbf{C}(\mathbf{s} - \mathbf{s}^*(t)) \tag{1}$$

The derivative of the task function, considering $\mathbf{C}$ constant, will be:

$$\dot{\mathbf{e}} = \mathbf{C}(\dot{\mathbf{s}} - \dot{\mathbf{s}}^*) \tag{2}$$

It is well known that the interaction matrix $\mathbf{L}$, also called image jacobian, plays a crucial role in the design of the possible control laws. $\mathbf{L}$ is defined as:

$$\dot{\mathbf{s}} = \mathbf{L}\ \mathbf{v} \tag{3}$$

where $\mathbf{v} = (\mathbf{V^T}, \omega^\mathbf{T})$ is the camera velocity screw ($\mathbf{V}$ and ω represent its translational and rotational component respectively).

Plugging the Eq. (3) in (2) we obtain:

$$\dot{\mathbf{e}} = \mathbf{CLv} - \mathbf{C}\,\dot{\mathbf{s}}^* \tag{4}$$

A simple control law can be obtained by imposing the exponential convergence of the task function to zero:

$$\dot{\mathbf{e}} = -\lambda\,\mathbf{e} \qquad \text{so} \qquad \mathbf{CLv} = -\lambda\,\mathbf{e} + \mathbf{C}\,\dot{\mathbf{s}}^* \tag{5}$$

where λ is a positive scalar factor which tunes the speed of convergence:

$$\mathbf{v} = -\lambda\,(\mathbf{CL})^{-1}\mathbf{e} + (\mathbf{CL})^{-1}\mathbf{C}\,\dot{\mathbf{s}}^* \tag{6}$$

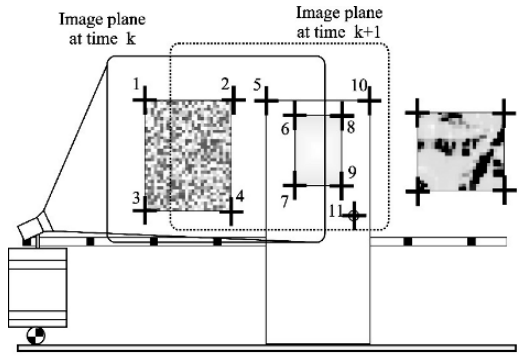
if $\mathbf{C}$ is setting to $\mathbf{L}^+$, then $(\mathbf{CL}) > 0$ and the task function converge to zero and, in the absence of local minima and singularities, so does the error $s - s^*$. Finally substituting $\mathbf{C}$ by $\mathbf{L}^+$ in Eq. (6), we obtain the expression of the camera velocity that is sent to the robot controller:

$$\mathbf{v} = -\lambda\,\mathbf{L}^+\,(\mathbf{s} - \mathbf{s}^*(t)) + \mathbf{L}^+\,\dot{\mathbf{s}}^* \tag{7}$$

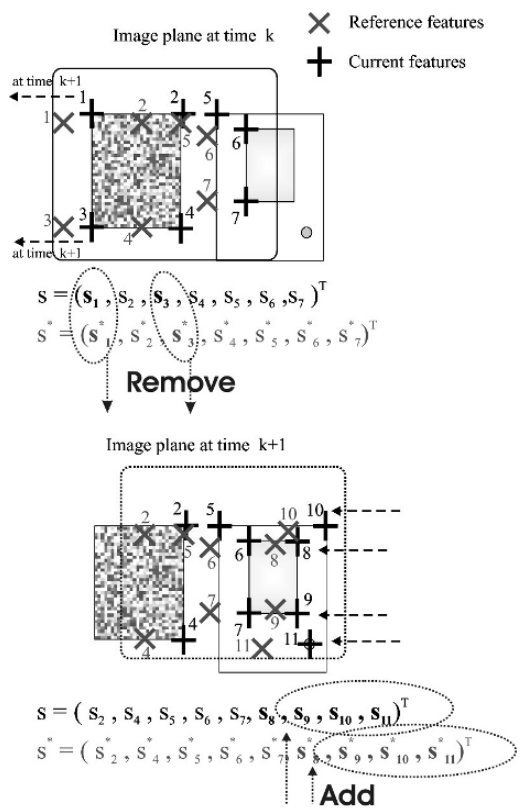
Remember that the whole formulation is directly applicable to the invariant visual servoing approach changing $\mathbf{s}$ by $\mathbf{q}$. As will be shown in the next section, discontinuities in the control law will be produced by the appearance/disappearance of image features during the navigation of the robot. The answer to the question: *why are these discontinuities produced* and their solution are presented in the following section.

3 DISCONTINUITIES IN VISUAL NAVIGATION

In this section, we describe more in details the discontinuity problem that occurs when some features go in/out of the image during the vision-based control. A simple simulation illustrate the effects of the discontinuities on the control law and on the performances of the visual servoing. The navigation of a mobile robot is a typical example where this kind of problems are produced.



(a) Croquis of the autonomous navigation of the robot



(b) Appearance/Disappearance of image features from time k to k+1

Figure 3: Navigation of a mobile robot controlled by visual servoing techniques.

3.1 What Happens When Features Appear or Disappear from the Image Plane?

During autonomous navigation of the robot, some features appear or disappear from the image plane so they will must be added to or removed from the visual error vector (Figure 3). This change in the error vector produces a jump discontinuity in the control law. The magnitude of the discontinuity in control law depends on the number of the features that go in or go out of the image plane at the same time, the distance between the current and reference features, and the pseudoinverse of interaction matrix.

In the case of using the invariant visual servoing approach to control the robot, the effect produced by the appearance/disappearance of features could be more important since the invariant space $\mathcal{Q}$ used to compute the current and the reference invariant points($\mathbf{q}, \mathbf{q}^*$) changes with features(Malis, 2002c).

4 CONTINUOUS CONTROL LAW FOR NAVIGATION

In the previous section, the continuity problem of the control law due to the appearance/disappearance of features has been shown. In this section a solution is presented. The section is organized as follows. First, the concept of weighted features is defined. Then, the definition of a smooth task function is presented. Finally, the reason to reformulate the invariant visual servoing approach and its development is explained.

4.1 Weighted Features

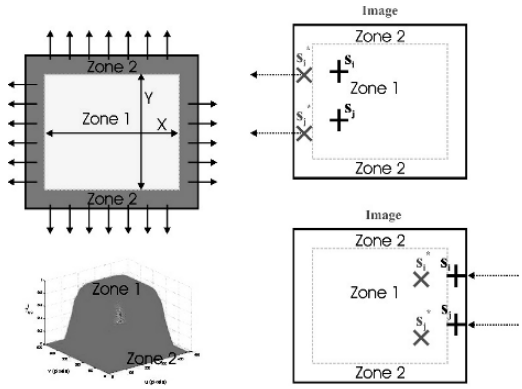
The key idea in this formulation is that every feature (points, lines, moments, etc) has its own weight which may be a function of image coordinates(u,v) and/or a function of the distance between feature points and an object which would be able to occlude them, etc (García et al., 2004). To compute the weights, three possible situations must be taking into account:

4.1.1 Situation 1: Changes of Visibility Through the Border of the Image (Zone 2 in Figure 4a)

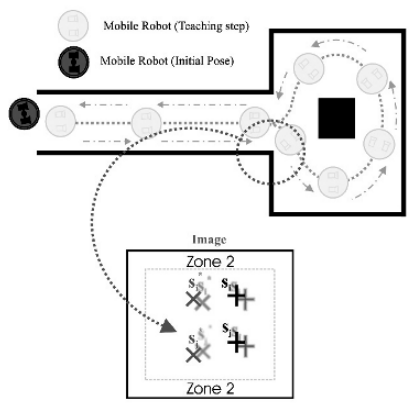
To anticipate the changes of visibility of features through the border, a total weight Φ_{uv} is computed as the product of the weights of the current and reference features which are function of their position in the image (γ^i_{uv}, $\gamma^{i\ *}_{uv}$). The weight $\gamma^i_{uv} = \gamma^i_u \cdot \gamma^i_v$ is computed using the definition of the function $\gamma_y(x)$ ($\gamma^i_u = \gamma_y(u_i)$ and $\gamma^i_v = \gamma_y(v_i)$ respectively) (García et al., 2004).

4.1.2 Situation 2: The Sudden Appearance of Features on the Center of the Image (Zone 1 in Figure 4b)

To take into account this possible situation, every new features (current and reference) must be checked



(a) Border of the image



(b) Center of the image

Figure 4: Appearance/Disappearance of image features through the border and the center of the image during the navigation of the mobile robot.

previously to known if they are in Zone 2 or Zone 1. If the new features are in Zone 2, the appearance of the features are considered in Situation 1. If they are in Zone 1, a new weight function must be defined to add these new features in a continuos way.

The weight function proposed Φ_a^i is a exponential function that tends to 1, reaching its maximum after a certain number of steps which can be modified with the α and β parameters (Figure 5):

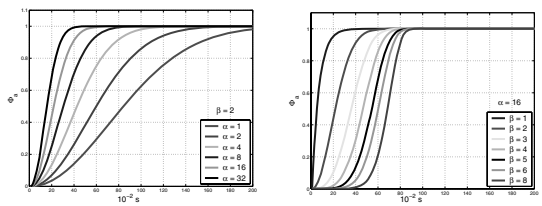
$$\Phi_a^i(t) = 1 - e^{-\alpha \cdot t^{\beta}} \quad \alpha,\ \beta\ > 0 \tag{8}$$

4.1.3 Situation 3: The Sudden Disappearance of Features on the Center of the Image because of a Temporal or Definitive Occlusion (Zone 1 in Figure 4b)

In this situation, the occlusions produced in the teaching step on the Zone 1 are only considered since they can be easily anticipated by the observation of reference features vector prerecorded previously.

To take into account this possible situation, a new weight function must be defined to remove these features from the current and reference vector in a continuos way. The weight function proposed Φ_o^i is a exponential function that tends to 0, reaching its minimum after a certain number of steps which can be modified with the ν and σ parameters:

$$\Phi_o^i(t) = e^{-\nu \cdot t^{\sigma}} \quad \nu,\ \sigma\ > 0 \tag{9}$$



(a) $\Phi_a(\alpha)$ (b) $\Phi_a(\beta)$

Figure 5: Plotting Φ_a in function of α and β.

4.1.4 Global Weight Function Φ^i

In this section, a global weight function Φ^i which includes the three possible situations commented before is presented. This function is defined as the product of the three weight functions (Φ_{uv}^i, Φ_a^i y Φ_o^i) which takes into account the three possible situations:

$$\Phi^i = \Phi_{uv}^i \cdot \Phi_a^i \cdot \Phi_o^i \ \text{ where } \ \Phi^i \in [0,1] \tag{10}$$

4.2 Smooth Task Function

Suppose that n matched points are available in the current image and in the reference features stored. Everyone of these points (current and reference) will have a weight Φ^i which can be computed as it's shown in the previous subsection 4.1. With them and their weights, a task function can be built (Samson et al., 1991):

$$\mathbf{e} = \mathbf{C}\mathbf{W}\ (\mathbf{s} - \mathbf{s}^*(t)) \tag{11}$$

where $\mathbf{W}$ is a $(2n \times 2n)$ diagonal matrix where its elements are the weights Φ^i of the current and reference features multiplied by the weights of the reference features.

The derivate of the task function will be:

$$\dot{\mathbf{e}} = \mathbf{C}\mathbf{W}\ (\dot{\mathbf{s}} - \dot{\mathbf{s}}^*) + (\mathbf{C}\dot{\mathbf{W}} + \dot{\mathbf{C}}\mathbf{W})(\mathbf{s} - \mathbf{s}^*(t)) \tag{12}$$

Plugging the equation ($\dot{\mathbf{s}} = \mathbf{L}\,\mathbf{v}$) in (12) we obtain:

$$\dot{\mathbf{e}} = \mathbf{CW}\,(\mathbf{Lv} - \dot{\mathbf{s}}^*) + (\mathbf{C}\dot{\mathbf{W}} + \dot{\mathbf{C}}\mathbf{W})(\mathbf{s} - \mathbf{s}^*(t)) \quad (13)$$

A simple control law can be obtained by imposing the exponential convergence of the task function to zero ($\dot{\mathbf{e}} = -\lambda\mathbf{e}$), where λ is a positive scalar factor which tunes the speed of convergence:

$$\begin{aligned} \mathbf{v} &= -\lambda\,(\mathbf{CWL})^{-1}\mathbf{e} + (\mathbf{CWL})^{-1}\mathbf{CW}\dot{\mathbf{s}}^* + \\ &- (\mathbf{CWL})^{-1}\,(\mathbf{C}\dot{\mathbf{W}} + \dot{\mathbf{C}}\mathbf{W})\,(\mathbf{s} - \mathbf{s}^*(t)) \end{aligned} \quad (14)$$

if $\mathbf{C}$ is setting to $(\mathbf{W}^*\mathbf{L}^*)^+$,then $(\mathbf{CWL}) > 0$ and the task function converge to zero and, in the absence of local minima and singularities, so does the error $\mathbf{s} - \mathbf{s}^*$. In this case, $\mathbf{C}$ is constant and therefore $\dot{\mathbf{C}} = 0$. Finally substituting $\mathbf{C}$ by $(\mathbf{W}^*\mathbf{L}^*)^+$ in Eq. (14), we obtain the expression of the camera velocity that is sent to the robot controller:

$$\begin{aligned} \mathbf{v} &= -(\mathbf{W}^*\mathbf{L}^*)^+\,(\lambda\,\mathbf{W} + \dot{\mathbf{W}})\,(\mathbf{s} - \mathbf{s}^*(t)) + \\ &+ (\mathbf{W}^*\mathbf{L}^*)^+\mathbf{W}\dot{\mathbf{s}}^* \end{aligned} \quad (15)$$

A block diagram of the controller proposed is shown in Figure 6.

4.3 Visual Servoing Techniques

The visual servoing techniques used to carry out the navigation are the image-based and the intrinsic-free approaches. In the case of image-based visual servoing approach, the control law (15) is directly applicable to assure a continuous navigation of a mobile robot. On the other hand, when the intrinsic-free approach is used, this technique must be reformulated to take into account the weighted features.

4.3.1 Intrinsic-free Approach

The theoretical background about invariant visual servoing can be extensively found in (Malis, 2002b; Malis, 2002c). In this section, we modify the approach in order to take into account weighted features (García et al., 2004).

Basically, the weights Φ^i defined in the previous subsection must be *redistributed*(γ_i) in order to be able to build the invariant projective space $\mathcal{Q}^{\gamma_i}$ where the control will be defined.

Similarly to the standard intrinsic-free visual servoing, the control of the camera is achieved by stacking all the reference points of space $\mathcal{Q}^{\gamma_i}$ in a $(3n\times 1)$ vector $\mathbf{s}^*(\boldsymbol{\xi}^*) = (\mathbf{q}_1^*(t), \mathbf{q}_2^*(t), \cdots, \mathbf{q}_n^*(t))$. Similarly, the points measured in the current camera frame are stacked in the $(3n\times 1)$ vector $\boldsymbol{s}(\boldsymbol{\xi}) = (\boldsymbol{q}_1(t), \boldsymbol{q}_2(t), \cdots, \boldsymbol{q}_n(t))$. If $\boldsymbol{s}(\boldsymbol{\xi}) = \boldsymbol{s}^*(\boldsymbol{\xi}^*)$ then $\boldsymbol{\xi} = \boldsymbol{\xi}^*$ and the camera is back to the reference position whatever the camera intrinsic parameters.

In order to control the movement of the camera, we use the control law Eq. (15) where $\mathbf{W}$ depends on the weights previously defined and $\mathbf{L}$ is the interaction matrix. The interaction matrix depends on current normalized points $\boldsymbol{m}_i(\boldsymbol{\xi}) \in \mathcal{M}$ ($\boldsymbol{m}_i$ can be computed from image points $\boldsymbol{m}_i = \boldsymbol{K}^{-1}\,\boldsymbol{p}_i$), on the invariant points $\boldsymbol{q}_i(\boldsymbol{\xi}) \in \mathcal{Q}^{\gamma}$, on the current depth distribution $\boldsymbol{z}(\boldsymbol{\xi}) = (Z_1, Z_2, ..., Z_n)$ and on the current redistributed weights γ_i. The interaction matrix in the weighted invariant space ($\boldsymbol{L}_{qi}^{\gamma_i} = \boldsymbol{T}_{mi}^{\gamma_i}\,(\boldsymbol{L}_{mi} - \boldsymbol{C}_i^{\gamma_i})$)is obtained like in (Malis, 2002a) but the term $\boldsymbol{C}_{\boldsymbol{i}}^{\boldsymbol{\gamma_i}}$ must be recomputed in order to take into account the *r*edistributed weights γ_i.

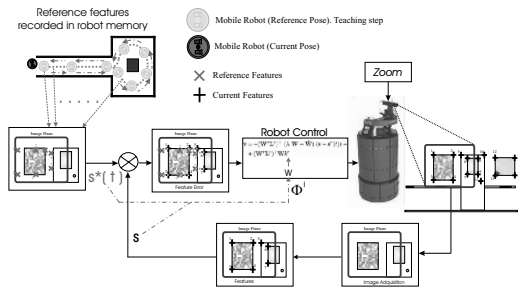
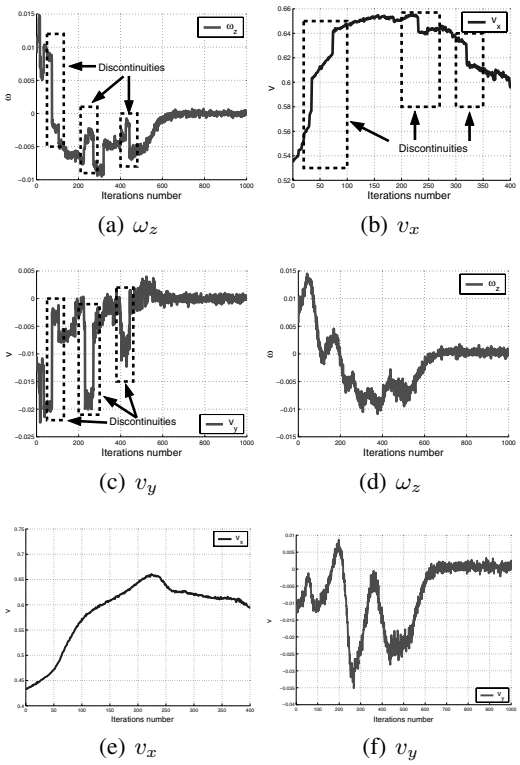


Figure 6: Block diagram of the controller proposed.

5 EXPERIMENTS IN A VIRTUAL INDOOR ENVIRONMENT

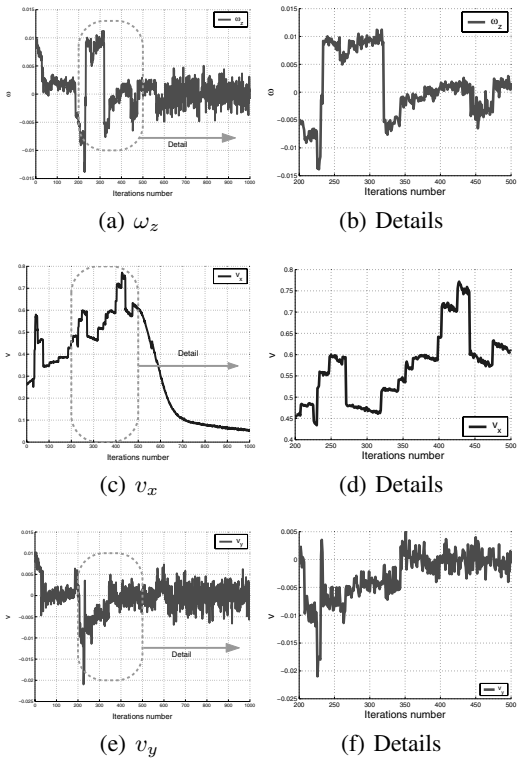
Exhaustive experiments have been carried out using a virtual reality tool for modeling an indoor environment. To make more realistic simulation, errors in intrinsic and extrinsic parameters of the camera mounted in the robot and noise in the extraction of image features have been considered. An estimation $\widehat{\mathbf{K}}$ of the real matrix $\mathbf{K}$ has been used with an error of 25% in focal length and a deviation of 50 pixels in the position of the optical center. Also an estimation $\widehat{\mathbf{T}}_{RC}$ of the camera pose respect to the robot frame has been computed with a rotation error of $u\theta = [3.75\ 3.75\ 3.75]^T\ degrees$ and translation error of $t = [2\ 2\ 0]^T\ cm$. An error in the extraction of current image features has been considered by adding a normal distribution noise to the accurate image features extracted.

In Figure 7, the control signals sent to the robot controller using the classical image-based approach and the image-based approach with weighted features are shown. In Figure 7 (a,b,c), details of the control law using the classical image-based approach, where



(a) ω_z (b) v_x (c) v_y (d) ω_z (e) v_x (f) v_y

Figure 7: Control law: Classical image-based approach (a-b-c) and image-based approach with weighted features (d-e-f).The translation and rotation speeds are measured respectively in $\frac{m}{s}$ and $\frac{deg}{s}$.



(a) ω_z (b) Details (c) v_x (d) Details (e) v_y (f) Details

Figure 8: Discontinuities in the control law: Intrinsic-free approach.

the discontinuities can be clearly appreciated, are presented. To show the improvements of the new formulation presented in this paper, the control law using the image-based with weighted features can be seen in Figure 7 (d,e,f).

The same experiment, but in this case using the intrinsic-free visual servoing approach, is performed. In Figure 8, the control signals sent to the robot controller using the intrinsic-free approach and some details, where the discontinuities can be clearly appreciated, are shown. The improvements of the new formulation of the intrinsic-free approach with weighted features are presented in Figure 9. The same details of the control law shown in Figure 8 are presented in Figure 9. Comparing both figures and their details, the continuity of the control law is self-evident despite of the noise in the extraction of image features.

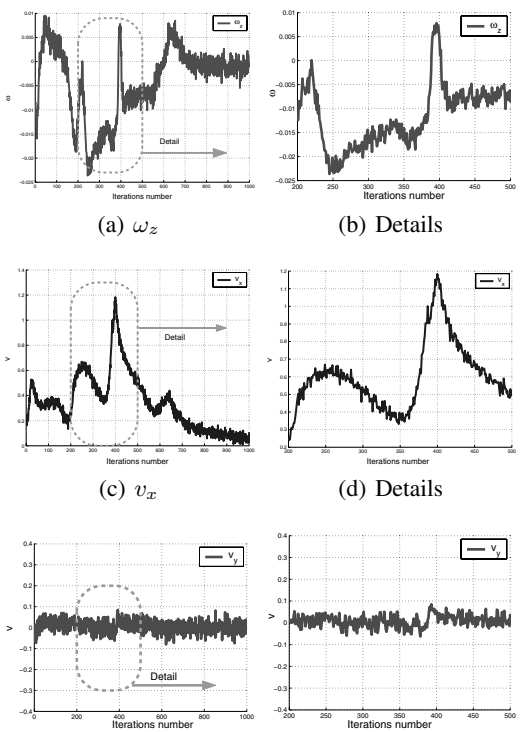
Also in (García et al., 2004), a comparison between this method and a simple filtration of the control law was presented. The results presented in that paper corroborate that the new approach with weighted features to the problem works better than a simple filter of the control signals.

6 CONCLUSIONS

In this paper the originally definition of the visual navigation as a global visual control task is presented. It implies continuity problems produced by the changes of visibility of image features during the navigation which have been solved by the definition of a smooth task function and a continuous control law obtained from it. The results presented corroborate that the new approach is continuous, stable and works better than a simple filter of the control signals. The validation of this results with a real robot is on the way by using a B21r mobile robot from iRobot company.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish Government through the 'Comision Interministerial de Ciencia y Tecnologa' (CICyT) through project "Técnicas avanzadas de teleoperación y realimentación sensorial aplicadas a la cirugía asistida por robots" DPI2005-08203-C02-02.



(a) ω_z (b) Details

(c) v_x (d) Details

(e) v_y (f) Details

Figure 9: Continuous control law: Intrinsic-free approach with weighted features.

REFERENCES

García, N., Malis, E., Reinoso, O., and Aracil, R. (2004). Visual servoing techniques for continuous navigation of a mobile robot. In *1st International Conference on Informatics in Control, Automation and Robotics*, volume 1, pages 343–348, Setubal, Portugal.

Hutchinson, S. A., Hager, G. D., and Corke, P. I. (1996). A tutorial on visual servo control. *IEEE Trans. Robotics and Automation*, 12(5):651–670.

Malis, E. (2002a). Stability analysis of invariant visual servoing and robustness to parametric uncertainties. In *Second Joint CSS/RAS International Workshop on Control Problems in Robotics and Automation*, Las Vegas, Nevada.

Malis, E. (2002b). A unified approach to model-based and model-free visual servoing. In *European Conference on Computer Vision*, volume 2, pages 433–447, Copenhagen, Denmark.

Malis, E. (2002c). Vision-based control invariant to camera intrinsic parameters: stability analysis and path tracking. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 217–222, Washington, USA.

Malis, E. and Cipolla, R. (2000). Self-calibration of zooming cameras observing an unknown planar structure. In *Int. Conf. on Pattern Recognition*, volume 1, pages 85–88, Barcelona, Spain.

Matsumoto, Y., Inaba, M., and Inoue, H. (1996). Visual navigation using view-sequenced route representation. In *IEEE International Conference on Robotics and Automation (ICRA'96)*, volume 1, pages 83–88, Minneapolis, USA.

Samson, C., Le Borgne, M., and Espiau, B. (1991). *Robot Control: the Task Function Approach.* volume 22 of Oxford Engineering Science Series.Clarendon Press., Oxford, UK, 1st edition.

SYNTHESIZING DETERMINISTIC CONTROLLERS IN SUPERVISORY CONTROL

Andreas Morgenstern and Klaus Schneider
University of Kaiserslautern, Department of Computer Science
P.O. Box 3049, 67653 Kaiserslautern, Germany
http://rsg.informatik.uni-kl.de
{*morgenstern,Klaus.Schneider*}*@informatik.uni-kl.de*

Keywords: Controller Synthesis, Supervisory Control, Discrete Event Systems.

Abstract: Supervisory control theory for discrete event systems is based on finite state automata whose inputs are partitioned into controllable and uncontrollable events. Well-known algorithms used in the Ramadge-Wonham framework disable or enable controllable events such that it is finally possible to reach designated final states from every reachable state. However, as these algorithms compute the least restriction on controllable events, their result is usually a nondeterministic automaton that can not be directly implemented. For this reason, one distinguishes between supervisors (directly generated by supervisory control) and controllers that are further restrictions of supervisors to achieve determinism. Unfortunately, controllers that are generated from a supervisor may be blocking, even if the underlying discrete event system is nonblocking. In this paper, we give a modification of a supervisor synthesis algorithm that enables us to derive deterministic controllers. Moreover, we show that the algorithm is both correct and complete, i.e., that it generates a deterministic controller whenever one exists.

1 INTRODUCTION

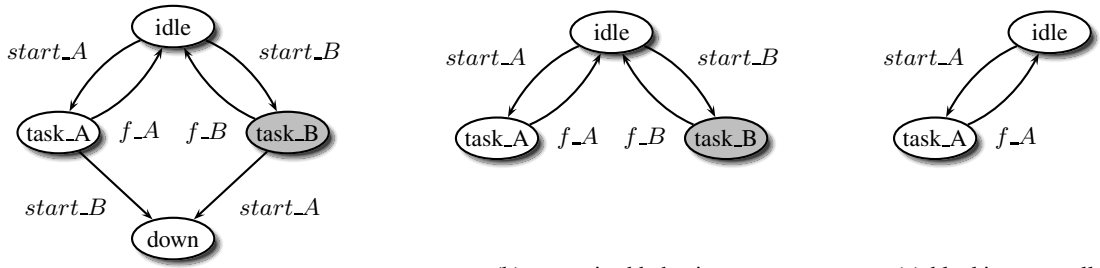
New applications in safety critical areas require the verification of the developed systems. In the past two decades, a lot of verification methods for checking the temporal behavior of a system have been developed (Schneider, 2003), and the research lead to tools that are already used in industrial design flows. These tools are able to check whether a system $\mathcal{K}$ satisfies a given temporal specification φ. There are a lot of formalisms, in particular, the μ-calculus (Kozen, 1983), ω-automata (Thomas, 1990), as well as temporal (Pnueli, 1977; Emerson and Clarke, 1982; Emerson, 1990) and predicate logics (Büchi, 1960b; Büchi, 1960a) to formulate the specification φ (Schneider, 2003). Moreover, industrial interest lead already to standardization efforts on specification logics (Accellera, 2004).

Besides the verification problem, where the entire system $\mathcal{K}$ and its specification must be already available, one can also consider the controller synthesis problem. The task is here to check whether there is a system $\mathcal{C}$ such that the coupled system $\mathcal{K} \parallel \mathcal{C}$ satisfies φ. Obviously, this problem is more general than the verification problem. Efficient solutions for this problem could be naturally used to guide the development of finite state controllers.

The controller synthesis problem is not new; several approaches exist for the so-called supervisory control problem. In particular, the supervisory control theory initiated by Ramadge and Wonham (Ramadge and Wonham, 1987) provides a framework for the control of discrete event systems. The system (also called a plant) is thereby modeled as a generator of a formal language. The control feature is represented by the fact that certain events can be disabled by a so-called supervisor. One result of supervisory control theory is that in case of formal languages, i.e., finite state machines, such a supervisor can be effectively computed.

However, if an implementation has to be derived from a supervisor, several problems have to be solved (Dietrich et al., 2002; Malik, 2003). A particular problem that we consider in this paper is the *derivation of a deterministic controller* from a supervisor that guarantees the *nonblocking property*. A system is thereby called nonblocking, if it is always possible to complete some task, i.e. to reach some designated (marked) state from every reachable state. If we consider the events as signals that can be sent to the plant, a valid controller should decide in every state what signal should be sent to the plant to ensure that the marked state is actually reached. However, even if the generated supervisor is nonblocking, a controller

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 95–102



(a) uncontrolled behavior (b) supervised behavior (c) blocking controller

Figure 1: Generation of a Blocking Controller.

that is derived by simply selecting in each state one of the allowed events/signals could be blocking.

As an example, consider the automaton that is given in Figure 1(a). This automaton represents a system with two tasks *task_A* and *task_B* that can be started with events *start_A* and *start_B*, respectively. These events are controllable, i.e. they can be disabled by a supervisor. If one of the machines completes its task, the (uncontrollable) events *f_A* and *f_B* occur, respectively, leading again to the initial state *idle*. Whenever both machines work at the same time, the system breaks down, since the state *down* is reached from where on no further progress is possible. Supervisory control theory can fix the problem that state *down* is reached by disabling events *start_B* in state *task_A* and *start_A* in state *task_B* (Figure 1(b)). However, when we have to implement a *deterministic* controller that has to select one of the signals *start_A* and *start_B*, we get a serious problem: if the controller always selects *start_A*, the marked state *task_B* is never reached, and therefore the nonblocking property is violated (Figure 1(c)).

In (Malik, 2003; Dietrich et al., 2002), the generation of deterministic controllers is restricted to cases where certain conditions hold. It is proved that these conditions guarantee that *every* deterministic controller derived from the supervisor is nonblocking. However, no controller can be constructed in case the discrete event system does not satisfy these conditions. In particular, a valid controller may exist, even if the conditions of (Malik, 2003; Dietrich et al., 2002) do not hold. For example, this is the case for the automaton given in Figure 1. A valid controller is obtained by selecting *start_B* in state *idle*.

In this paper, *we present a new approach to generate deterministic controllers from supervisors* that does not suffer from the above problem. To this end, we introduce a more general property than nonblocking which we call *forceable nonblocking*. A discrete event system satisfies this property if and only if there exists a deterministic controller that ensures that every run (either finite or infinite) of the controlled system visits a marked state. Obviously, this requirement is stronger than the nonblocking property. Our algorithm guarantees that a marked state will be reached, no matter how the plant behaves. In contrast, the nonblocking property only requires that the plant *has the chance* to reach a marked state. Although our property is more general than nonblocking, our algorithm is just a slight adjustment of the original supervisor synthesis algorithm which is known to have moderate complexity bounds.

The paper is organized as follows: In the next Section, we present the basics of supervisory control theory. In Section 3, we present our new algorithm to compute deterministic nonblocking controllers from supervisors whenever this is possible. Finally, the paper ends with some conclusions and directions for future work.

2 SUPERVISORY CONTROL THEORY

In this section, we will give a brief introduction to the supervisory control theory as initiated by Ramadge and Wonham (Ramadge and Wonham, 1987). For a more detailed treatment of the topic we refer to (Wonham, 2001).

Traditionally, control theory has focused on control of systems modeled by differential equations, so-called continuous variable dynamic systems. There, the feedback signal from the controller influences the behavior of the system, enforcing a given specification that would not be met by the open-loop behavior. Another important class of system models are those where states have symbolic values instead of numerical ones. These systems change their state whenever an external or internal event occurs. This class of systems, called *discrete event systems (DES)*, is the focus of supervisory control theory (Ramadge and Wonham, 1987).

The theoretical roots of supervisory control theory explain some of the terminology used. In the Ramadge Wonham (RW) framework, one speaks of a

plant, a system which generates events and encompasses the whole physically possible behavior of the system to be controlled (including unwanted situations). The *specification* is a subset of this behavior that should be matched by adding a controller. A *supervisor* is an entity that is coupled with the plant through a communication channel that allows the supervisor to influence the behavior of the plant by enabling those events that may be generated in the next state of the system (see Figure 2). Usually, in a physical system, not all of the events can be influenced by an external supervisor. This is captured by distinguishing between events that can be prevented from occurring, called *controllable events*, and those that cannot be prevented, called *uncontrollable events*. We denote the sets of uncontrollable and controllable events as Σ_u and Σ_c, respectively, and define $\Sigma = \Sigma_c \cup \Sigma_u$.

The Ramadge Wonham formulation of the supervisory control problem makes use of formal language theory and automata: A finite automaton is a 5-tuple $\mathcal{A} = \langle Q, \Sigma, \delta, q^0, M\rangle$ where Σ is a set of events, Q is a set of states, $\delta : Q \times \Sigma \to Q$ is a transition function, and $q^0 \in Q$ is the initial state. The states in the set $M \subseteq Q$ are chosen to mark the completion of tasks by the system and are therefore called *marker states*. We write $\delta(q, \sigma) \downarrow$ to signify that there exists a transition labeled with σ, leaving q. It is often necessary to refer to the set of events for which there is a transition leaving state q. We refer to these events as active events:

Definition 1 (Active Events) *Given an automaton $\mathcal{A} = \langle Q, \Sigma, \delta, q^0, M\rangle$ and a particular state $q \in Q$, the set of active events of q is:*

$$\mathsf{act}_{\mathcal{A}}(q) := \{\sigma \in \Sigma \mid \delta(q, \sigma) \downarrow\}$$

If the plant and the supervisor are represented using finite automata, the control action of the supervisor is captured by the synchronous product:

Definition 2 (Automata Product) *Given automata $\mathcal{A}_\mathcal{P} = \langle \Sigma, Q_\mathcal{P}, \delta_\mathcal{P}, q^0_\mathcal{P}\ M_\mathcal{P}\rangle$ and $\mathcal{A}_\mathcal{S} = \langle \Sigma, Q_\mathcal{S}, \delta_\mathcal{S}, q^0_\mathcal{S}, M_\mathcal{S}\rangle$, the product $\mathcal{A}_\mathcal{P} \times \mathcal{A}_\mathcal{S}$ is the automaton $\langle \Sigma, Q_\mathcal{P} \times Q_\mathcal{S}, \delta_{\mathcal{P}\times\mathcal{S}}, (q^0_\mathcal{P}, q^0_\mathcal{S}), M_\mathcal{P} \times M_\mathcal{S}\rangle$, where*

$$\delta_{\mathcal{P}\times\mathcal{S}}((p,q),\sigma) = (p',q') \text{ iff } \delta_\mathcal{P}(p,\sigma) = p' \wedge \delta_\mathcal{S}(q,\sigma) = q'$$

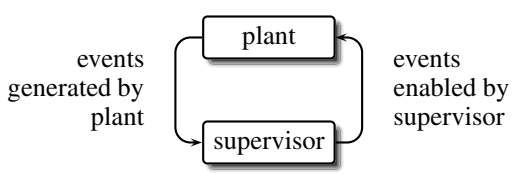


Figure 2: The Ramadge-Wonham Framework.

Note that in a state (p, q) of the synchronous product, the active events are exactly those events that are active both in the plant and the supervisor,i.e.

$$\mathsf{act}_{\mathcal{A}_\mathcal{P}\times\mathcal{A}_\mathcal{S}}((p,q)) = \mathsf{act}_{\mathcal{A}_\mathcal{P}}(p) \cap \mathsf{act}_{\mathcal{A}_\mathcal{S}}(q).$$

Disabling controllable events in the states of the supervisor will therefore also disable them in the product. This is how the supervisor enforces his control function.

The behavior of the plant represented by a finite automaton is closely related to two formal languages over the alphabet of events Σ, the generated language $\mathsf{L}(\mathcal{A})$ and the marked language $\mathsf{L_m}(\mathcal{A})$. The generated language $\mathsf{L}(\mathcal{A})$ represents sequences of events that the plant generates during execution while the marked language $\mathsf{L_m}(\mathcal{A})$ represents those event sequences that lead to a marker state. Formally, the two languages are defined as follows[1]:

Definition 3 (Generated and Marked Language)

$$\mathsf{L}(\mathcal{A}) = \{w \in \Sigma^* : \delta(q^0, w) \downarrow\}$$
$$\mathsf{L_m}(\mathcal{A}) = \{w \in \Sigma^* : \delta(q^0, w) \in M\}$$

Given both the plant $\mathcal{A}_\mathcal{P}$ and the supervisor $\mathcal{A}_\mathcal{S}$, the generated and marked language of the controlled system are denoted by $\mathsf{L}(\mathcal{A}_\mathcal{P}/\mathcal{A}_\mathcal{S})$ and $\mathsf{L_m}(\mathcal{A}_\mathcal{P}/\mathcal{A}_\mathcal{S})$ and defined by the generated and marked language of the product automaton:

$$\mathsf{L}(\mathcal{A}_\mathcal{P}/\mathcal{A}_\mathcal{S}) := \mathsf{L}(\mathcal{A}_\mathcal{P} \times \mathcal{A}_\mathcal{S}) = \mathsf{L}(\mathcal{A}_\mathcal{P}) \cap \mathsf{L}(\mathcal{A}_\mathcal{S})$$

$$\begin{aligned}\mathsf{L_m}(\mathcal{A}_\mathcal{P}/\mathcal{A}_\mathcal{S}) :&= \mathsf{L_m}(\mathcal{A}_\mathcal{P} \times \mathcal{A}_\mathcal{S}) \\ &= \mathsf{L_m}(\mathcal{A}_\mathcal{P}) \cap \mathsf{L_m}(\mathcal{A}_\mathcal{S})\end{aligned}$$

When we consider algorithms, it is also necessary that the specification is given as a finite automaton. The assumption that the uncontrollable events can not be prevented from occurring, places restrictions on the possible supervisors. Therefore, a specification automaton $\mathcal{A}_\mathcal{E}$ is called *controllable with respect to a plant $\mathcal{A}_\mathcal{P}$*, if and only if for every state (p, q) of $\mathcal{A}_\mathcal{P} \times \mathcal{A}_\mathcal{E}$ that is reachable by a string in $\mathsf{L}(\mathcal{A}_\mathcal{P} \times \mathcal{A}_\mathcal{E})$ and every uncontrollable event $\sigma \in \Sigma_u$, the following holds:

$$\sigma \in \mathsf{act}_{\mathcal{A}_\mathcal{P}}(p) \Rightarrow \sigma \in \mathsf{act}_{\mathcal{A}_\mathcal{E}}(q).$$

In other words, $\mathcal{A}_\mathcal{E}$ is controllable if and only if no word of $\mathsf{L}(\mathcal{A}_\mathcal{P})$ that is generated according to the specification, exits from the behavior permitted by the specification if it is followed by an uncontrollable event. Specifications that do not fulfill this requirement are called *uncontrollable*. If a specification is uncontrollable, the product automaton contains one or

[1] As usual, we allow δ to process also words instead of only single events.

more reachable *bad states*, which are states (p, q) that fail to satisfy the following condition:

$$\mathsf{act}_{\mathcal{A}_\mathcal{P} \times \mathcal{A}_\mathcal{E}}((p,q)) \supseteq \mathsf{act}_{\mathcal{A}_\mathcal{P}}(p) \cap \Sigma_u$$

Given a specification automaton $\mathcal{A}_\mathcal{E}$, the language $K = \mathsf{L_m}(\mathcal{A}_\mathcal{E})$ is *controllable* if and only if $\mathcal{A}_\mathcal{P} \times \mathcal{A}_\mathcal{E}$ has no bad states. Besides controllability, another important property of discrete event systems is the *nonblocking* property which states that it is always possible to complete some task, i.e. that from every reachable state $q \in Q$, it is possible to reach a marked state. Formally, an automaton is nonblocking, if and only if for each reachable state $q \in Q$, we have

$$\mathsf{L_m}(q) = \{w \in \Sigma^* \mid \delta(q, w) \in M\} \neq \emptyset.$$

States that have a path to a marked state are called *coreachable*. Ramadge and Wonham have shown that given a specification K which is not controllable, it is possible to construct for every plant $\mathcal{A}_\mathcal{P}$ and every specification $\mathcal{A}_\mathcal{E}$ the *supremal controllable sublanguage* of K, denoted $\mathsf{supC}(K)$. This result is of practical interest: Given that the specification language K is uncontrollable, it is possible to compute $\mathsf{supC}(K)$ and to construct a supervisor $\mathcal{A}_\mathcal{S}$ such that $\mathsf{L_m}(\mathcal{A}_\mathcal{S}/\mathcal{A}_\mathcal{P}) = \mathsf{supC}(K)$. This implies that the controlled system is nonblocking, meaning that the constructed supervisor does not prevent the plant from completing a task. This supervisor is a solution to the following problem:

Definition 4 (Supervisory Control Problem) *Given a plant $\mathcal{A}_\mathcal{P}$, a specification language $K \subseteq \mathsf{L_m}(\mathcal{A}_\mathcal{P})$ representing the desired behavior of $\mathcal{A}_\mathcal{P}$ under supervision, find a nonblocking supervisor $\mathcal{A}_\mathcal{S}$ such that $\mathsf{L_m}(\mathcal{A}_\mathcal{S}/\mathcal{A}_\mathcal{P}) \subseteq K$.*

Given a specification automaton $\mathcal{A}_\mathcal{E}$, we can construct the least restrictive solution from the product automaton $\mathcal{A}_\mathcal{P} \times \mathcal{A}_\mathcal{E}$. The marked language of this least restrictive solution $\mathcal{A}_\mathcal{S}$ is equal to $\mathsf{supC}(K)$. If an automaton $\mathcal{A} = \langle Q, \Sigma, \delta, q^0_\mathcal{A}, M_\mathcal{A}\rangle$ is given that represents the product of the plant and the specification, algorithm 1 can be used to compute this supervisor (Ziller and Schneider, 2003).

Essentially, this algorithm consists of two loops. The inner loop calculates the coreachable states x_C, and the outer loop computes the good states x_G, i.e. states that are not bad states. Since removing bad states could destroy the coreachability property and removing non-coreachable states could result in new bad states, the two loops have to be nested. Based on this algorithm, we will provide an algorithm that calculates a supervisor with the property that every deterministic controller generated from this supervisor is a valid controller, i.e. guarantees that a marked state is reached, irrespectively of the behavior of the plant.

Algorithm 1: Supervisor Synthesis Algorithm.

$x^0_G = Q_\mathcal{A} \setminus \{q \in Q \mid q \text{ is initial bad}\}$;
$j = 0$;
repeat
  $x^{(0,j)}_C = M \cap x^j_G$;
  $i = 0$;
  repeat

$$x^{(i+1,j)}_C = x^j_G \cap \left(x^{i,j}_C \cup \left\{ q \in Q \;\middle|\; \begin{array}{l} \exists \sigma \in \mathsf{act}_\mathcal{A}(q)\,. \\ \\ \delta_\mathcal{A}(q,\sigma) \in x^{(i,j)}_C \end{array} \right\} \right)$$

    $i = i + 1$;
  until $x^{i,j}_C = x^{i-1,j}_C$;
  $x^{j+1}_G = x^j_G \cap \left\{ q \in Q \;\middle|\; \begin{array}{l} \forall \sigma \in \mathsf{act}_\mathcal{A}(q) \cap \Sigma_u. \\ \delta_\mathcal{A}(q,\sigma) \in x^{(i,j)}_C \cap x^j_G \end{array} \right\}$
  $j = j + 1$;
until $x^j_g = x^{j-1}_g$;

3 CONTROLLER SYNTHESIS

We have seen by the example given in Figure 1 that the nonblocking property is too weak to guarantee that a marked state is reached under control by a deterministic controller. This is due to the fact that a state is coreachable even if there exists an infinitely long sequence of events that never visits a marked state. We therefore sharpen the coreachability property as follows:

Definition 5 (Forceably Coreachable States) *A state is forceable coreachable, if it is coreachable and*

$$\exists n \in \mathbb{N}. \forall t \in \Sigma^*. \left\{ \begin{array}{l} \delta(q,t)\downarrow \wedge |t| \geq n \Rightarrow \exists t' \sqsubseteq t.\delta(q,t') \in Q_m \wedge \\ \delta(q,t)\downarrow \wedge |t| < n \Rightarrow \left(\begin{array}{l} \exists t' \sqsubseteq t.\delta(q,t') \in Q_m \vee \\ \mathsf{act}_\mathcal{A}(\delta(q,t)) \neq \emptyset \end{array} \right) \end{array} \right.$$

Intuitively, a state is forceable coreachable, if there exists a threshold after which a marked state is unavoidable. In contrast to the definition of coreachability that imposes a condition on the future, we demand something about the past: we demand that after a certain amount of steps (referenced by n), a marked state must have been visited. As long as this bound n is not reached, we demand that either the system does not stop or that a marked state has already been reached.

In terms of temporal logics, we demand that *on all paths a marked state must be reached*. In contrast, the nonblocking property only states that *for all states there exists a path where M is reached.* We call an automaton *forceable nonblocking*, if each reachable

state is forceable coreachable. The Controller Synthesis Problem is now given as follows:

Definition 6 (Controller Synthesis Problem) *Given a plant $\mathcal{A}_\mathcal{P}$, a specification language $K \subseteq \mathsf{L_m}(\mathcal{A}_\mathcal{P})$ representing the desired behavior of $\mathcal{A}_\mathcal{P}$ under control, find a nonblocking supervisor $\mathcal{A}_\mathcal{C}$ such that*

- $\mathsf{L_m}(\mathcal{A}_\mathcal{C}/\mathcal{A}_\mathcal{P}) \subseteq K$.
- $\mathcal{A}_\mathcal{C} \times \mathcal{A}_\mathcal{P}$ *is forceable nonblocking.*

Hence, a controller ensures that a marked state is actually reached. It is very easy to derive a deterministic controller from such a solution: in every step, we can simply select a controllable event to ensure that a marked state is actually reached. This is due to the fact, that we demand that all paths leaving a forceable coreachable state sooner or later reach a marked state. Therefore, it is irrelevant which of the active controllable events we select.

Theorem 1 *Given $\mathcal{A}_\mathcal{P} = \langle Q, \Sigma, \delta, q^0_{\mathcal{A}_\mathcal{P}}, M_{\mathcal{A}_\mathcal{P}}\rangle$ and $\mathcal{A}_\mathcal{C} = \langle Q, \Sigma, \delta, q^0_{\mathcal{A}_\mathcal{C}}, M_{\mathcal{A}_\mathcal{C}}\rangle$ such that*

$$\mathsf{L}(\mathcal{A}_\mathcal{C}) \subseteq \mathsf{L}(\mathcal{A}_\mathcal{P}) \wedge \mathsf{L_m}(\mathcal{A}_\mathcal{C}) \subseteq \mathsf{L_m}(\mathcal{A}_\mathcal{P}),$$

then, the following holds: If $\mathcal{A}_\mathcal{C}$ is forceable coreachable then $\mathcal{A}_\mathcal{C} \times \mathcal{A}_\mathcal{P}$ is forceable coreachable.

Proof: Let $(q,p) \in Q_{\mathcal{A}_\mathcal{C}} \times Q_{\mathcal{A}_\mathcal{P}}$ be reachable, such that $\delta_{\mathcal{A}_\mathcal{C}\times\mathcal{A}_\mathcal{P}}((q_0^{\mathcal{A}_\mathcal{C}}, q_0^{\mathcal{A}_\mathcal{P}}), s) = (q,p)$. Then, also $q \in Q_{\mathcal{A}_\mathcal{C}}$ must be reachable in $\mathcal{A}_\mathcal{C}$. Therefore, q is forceable coreachable with a constant n. Now, choose a $t \in \Sigma^*$ such that $\delta_{\mathcal{A}_\mathcal{C}\times\mathcal{A}_\mathcal{P}}((q,p),t)\downarrow$. We distinguish between two cases: First, we assume $|t| \geq n$. Then, there exists a $t' \sqsubseteq t$ such that $\delta(q,t') \in M_{\mathcal{A}_\mathcal{C}}$. Therefore, $st' \in \mathsf{L_m}(\mathcal{A}_\mathcal{C}) \subseteq \mathsf{L_m}((\mathcal{A}_\mathcal{P}))$ holds. Since all automata are deterministic, it follows that $\delta((q,p),t') \in (M_{\mathcal{A}_\mathcal{C}} \times M_{\mathcal{A}_\mathcal{P}})$ holds. In the remaining case, we have $|t| < n$. Then, either there exists a $t' \sqsubseteq t$ that visits a marked state as in the first case or $\mathsf{act}_{\mathcal{A}_\mathcal{C}}(\delta(q,t)) \neq \emptyset$. Again, since the language inclusion holds, we have $\mathsf{act}_{\mathcal{A}_\mathcal{C}\times\mathcal{A}_\mathcal{P}}(\delta((q,p),t)) \neq \emptyset$. ∎

4 CONTROLLER SYNTHESIS ALGORITHM

In this section, we develop a controller synthesis algorithm based on the supervisor synthesis algorithm of Section 1. In order to guarantee the forceable nonblocking property, we have to adopt the calculation of the coreachable states. In contrast to the coreachability property, which only demands that a marked state is reachable, i.e. that it is possible to directly reach a marked state or to reach a state which is known to be coreachable, a state is forceable coreachable if it is coreachable and all events lead to forceable coreachable states. State and event pairs that guarantee this property are collected in the set $moves$. This implies that all destination states of uncontrollable transitions leaving a state q must be identified as forceable coreachable before we can add any transition from q to moves. Otherwise, q is bad, which is identified in the x_G-loop. This ensures that the controllability property is not violated. To prevent the plant from looping, we forbid adding new moves, if we had already found a move that lead to a marked state. This is done due to the fact that those newly found moves will need a longer path to reach a marked state than the already introduced moves and may therefore introduce loops. We collect the forceable coreachable states in the set x_C by adding those states that have a path to a marked state where this can be guaranteed. Altogether, we thus have developed algorithm 2.

Algorithm 2: Controller Synthesis Algorithm.

$j = 0$;
$x_G^0 = Q_\mathcal{A} \setminus \{q \in Q_\mathcal{A} \mid q \text{ is initial bad}\}$;
repeat
  $x_C^{(0,j)} = M \cap x_G^j$;
  $i = 0$;
  $move^{(0,j)} = \{\}$;
  repeat
    $$move^{(i+1,j)} = move^{(i,j)} \cup \left\{ (q,\sigma) \;\middle|\; \begin{array}{c} \delta_\mathcal{A}(q,\sigma) \in x_C^{(i,j)} \\ \wedge \\ \left(\begin{array}{l} \forall \sigma \in \mathsf{act}_\mathcal{A}(q) \cap \Sigma_u. \\ \delta_\mathcal{A}(q,\sigma) \in x_C^{(i,j)} \end{array} \right) \\ \wedge \\ \forall \sigma \in \Sigma.\ (q,\sigma) \notin move^{(i,j)} \end{array} \right\}$$
    $$x_C^{(i+1,j)} = x_G^j \cap \left(x_C^{(i,j)} \cup \left\{ q \;\middle|\; \begin{array}{l} \exists (q,\sigma) \in move^{(i+1,j)} \\ \delta_\mathcal{A}(q,\sigma) \in x_C^{(i,j)} \end{array} \right\} \right)$$
    $i = i + 1$;
  until $x_C^i = x_C^{i-1}$;
  $$x_G^{j+1} = x_G^j \cap \left\{ q \in Q \;\middle|\; \begin{array}{l} \forall \sigma \in \mathsf{act}_\mathcal{A}(q) \cap \Sigma_u. \\ \delta_\mathcal{A}(q,\sigma) \in x_C^{(i,j)} \cap x_G^j \end{array} \right\}$$
  $j = j + 1$
until $x_g^j = x_g^{j-1}$;

The above algorithm may only loop for a finite number of iterations, since there are only finitely many states: In x_C, only finitely many states may be added and from x_G only finitely many states may be removed. Therefore, there exists a k such that $x_G^k = x_G^{k+1}$ finally holds. Additionally, for every i there exists a l such that $x_C^{(i,l)} = x_C^{(i,l+1)}$ and $moves^{(i,l)} = moves^{(i,l+1)}$. For this reason, we use the following notation: $x_G^k = x_G^\infty$ and also $x_C^{(i,\infty)} := x_C^{(i,k)}$ as well

as $move^{(i,\infty)} := move^{(i,k)}$ for every i, and finally $x_C^{(\infty,\infty)} := x_C^{(l,k)}$ and $moves^{(\infty,\infty)} := moves^{(l,k)}$ for the last iteration step.
Note that according to the definition of x_C, it holds that $x_C^{(i,j)} \subseteq x_G^j$ for every i, j and thus also $x_C^{(\infty,\infty)} \subseteq x_G^\infty$ holds. Since $move$ does only contain transitions leading to forceable coreachable states, it thus contains only transitions to good states.

If $q_\mathcal{A}^0 \in x_C^{(\infty,\infty)}$ holds, we define a controller as follows: $\mathcal{A}_\mathcal{C} = \langle Q_\mathcal{A}, \Sigma, \delta_{\mathcal{A}_\mathcal{C}}, q_\mathcal{A}^0, M_\mathcal{A} \rangle$ with

$$\delta_{\mathcal{A}_\mathcal{C}}(q,\sigma) = \begin{cases} \delta_\mathcal{A}(q,\sigma) & \text{, if } (q,\sigma) \in move^{(\infty,\infty)} \\ \uparrow & \text{, else} \end{cases}$$

The following lemma shows that we decrease the distance to a marked state whenever we use an event enabled by the controller:

Lemma 1

$$\forall i > 0 \forall q \in \left(x_C^{(i,\infty)} \setminus x_C^{(i-1,\infty)} \right) \forall \sigma \in \mathsf{act}_{\mathcal{A}_\mathcal{C}}(q) . \\ \delta_{\mathcal{A}_\mathcal{C}}(q,\sigma) \in x_C^{(i-1,\infty)}$$

Proof: Let $q \in Q_\mathcal{A}$ such that $q \in x_C^{(i+1,\infty)} \setminus x_C^{(i,\infty)}$. This implies that there must exist a move $(q,\sigma) \in \left(move^{(i+1,\infty)} \setminus move^{(i,\infty)} \right)$ such that $\delta_\mathcal{A}(q,\sigma) \in x_C^{(i,\infty)}$. But this directly implies that $\delta_\mathcal{A}(q,\sigma) \in x_C^{(i,\infty)}$ for every $(q,\sigma) \in move^{(i+1,\infty)}$. We thus have the statement for those moves added in the $i+1$-th iteration step. Additionally, if follows from the definition of $move$ that there can be no move $(q,\sigma') \in move^{(\infty,\infty)} \setminus move^{(i+1,\infty)}$. Therefore $\delta_{\mathcal{A}_\mathcal{C}}(q,\sigma) \in x_C^{(i,\infty)}$ for every $\sigma \in \mathsf{act}_{\mathcal{A}_\mathcal{C}}(q)$. ■

The above lemma does not apply to marked states (those are contained in $x_C^{(0,\infty)}$). And indeed, without the additional set x_G, this would not be true. The next lemma fixes this deficiency.

Lemma 2

$$\forall q \in \left(M_\mathcal{A} \cap x_C^{(\infty,\infty)} \right) \forall \sigma \in \mathsf{act}_{\mathcal{A}_\mathcal{C}}(q) . \\ \delta_{\mathcal{A}_\mathcal{C}}(q,\sigma) \in x_C^{(\infty,\infty)}$$

Proof: Choose an arbitrary state $q \in M_{\mathcal{A}_\mathcal{C}} \cap x_C^{(\infty,\infty)} \subseteq x_G^\infty$. The proof follows directly for uncontrollable events because of the definition of x_G^∞. Thus, consider a controllable event. According to the definition of $\delta_{\mathcal{A}_\mathcal{C}}$, $\sigma \in \mathsf{act}_{\mathcal{A}_\mathcal{C}}(q) \cap \Sigma_c$ implies that $(q,\sigma) \in move^{(\infty,\infty)}$. According to the definition of $move$, we must have $(q,\sigma) \in move^{(i,\infty)}$ for a suitable i. Therefore, we have $\delta_\mathcal{A}(q,\sigma) \in x_C^{(i-1,\infty)}$, and thus $\delta_{\mathcal{A}_\mathcal{C}}(q,\sigma) \in x_C^{(i-1,\infty)}$. ■

Since the $x_C^{(i,\infty)}, i \in \mathbb{N}$ are monotone in i, the following Lemma follows inductively:

Lemma 3

$$\forall q \in x_C^{(\infty,\infty)} \forall t \in \Sigma^* . \\ \delta_{\mathcal{A}_\mathcal{C}}(q,t) \downarrow \Rightarrow \delta_{\mathcal{A}_\mathcal{C}}(q,t) \in x_C^{(\infty,\infty)}$$

While the above lemma only guarantees that the forceable coreachable states are never left, the next lemma shows that the plant may not stop until a marked state is reached:

Lemma 4

$$\forall i > 0 \forall q \in x_C^{(i,\infty)} \exists \sigma \in \mathsf{act}_{\mathcal{A}_\mathcal{C}}(q) . \\ \delta_{\mathcal{A}_\mathcal{C}}(q,\sigma) \in x_C^{(i-1,\infty)}$$

Proof: According to the definition and the monotony of x_C,

$$q \in x_C^{(i,\infty)} \Leftrightarrow q \in x_G^\infty \wedge \\ \left(\begin{array}{l} q \in M_\mathcal{A} \vee \exists (q,\sigma) \in move^{(i,\infty)}(q). \\ \delta_{\mathcal{A}_\mathcal{C}}(q,\sigma) \in x_C^{(i-1,\infty)} \end{array} \right)$$

If $q \in M_\mathcal{A}$ we are done, otherwise the lemma follows from the definition of $\delta_{\mathcal{A}_\mathcal{C}}$ and the monotony of $move^{(i,\infty)}$ with respect to i. ■

Finally, we now have the following theorem:

Theorem 2 *All $q \in x_C^{(\infty,\infty)}$ are coreachable in $\mathcal{A}_\mathcal{C}$.*

Proof: Take some $q \in x_C^{(\infty,\infty)}$. Then, $q \in x_C^{(i,\infty)}$ for some k. If q is marked, we are done. Otherwise, we can iteratively apply Lemma 4 to generate a string $t \in \Sigma^*$ that reaches a marked state. This is due to the fact that if we apply Lemma 4, then the i-index of the destination state decreases in each step. Therefore, after at most k iteration steps, we have constructed a word t such that $\delta(q,t) \in x_C^{(0,\infty)} = M_\mathcal{A} = M_{\mathcal{A}_\mathcal{C}}$. ■

We will show in the next lemma that also the stronger property of forceable coreachability holds:

Theorem 3 *All $q \in x_C^{(\infty,\infty)}$, are forceable coreachable*

Proof: The coreachability property follows from the last theorem. We will now show the rest of the forceable coreachability property for any $q \in x_C^{(\infty,\infty)}$: Since $q \in x_C^{(\infty,\infty)}$, there exists an $i \in \mathbb{N}$ such that $q \in x_C^{(i,\infty)} \setminus x_C^{(i-1,\infty)}$. If $i = 0$, we are done, because then x_c is marked. Otherwise we show that this i is the threshold that is required for forceable coreachability. Let $t \in \Sigma^*$ be such that $\delta_{\mathcal{A}_\mathcal{C}}(q,t) \downarrow$ holds. Applying Lemma 3 shows that $\delta_{\mathcal{A}_\mathcal{C}}(q,t) \in x_C^{(\infty,\infty)}$ holds. We distinguish two cases: If $|t| < i$ holds, then either $\delta_{\mathcal{A}_\mathcal{C}}(q,t) \in M_\mathcal{A}$ or $\exists \sigma \in \mathsf{act}_{\mathcal{A}_\mathcal{C}}(\delta(q,t)) . \delta_{\mathcal{A}_\mathcal{C}}(\delta(q,t),\sigma) \in x_C^{(i-(t+1),\infty)}$

according to Lemma 4. Both cases satisfy the condition of forceable coreachability for the case $|t| < i$.

Now consider any string t with length i and assume that $\delta_{\mathcal{A}_\mathcal{C}}(q,t') \notin M_\mathcal{A}$ for every $t' \sqsubseteq t$. Then, we can iteratively apply Lemma 1 i-times to conclude that $\delta_{\mathcal{A}_\mathcal{C}}(q,t) \in x_C^{(0,\infty)} \subseteq M_\mathcal{A}$ holds. The forceable coreachability property therefore holds for every string with length i and thus also for every string of length greater i. ■

The following theorem gives us the correctness of our algorithm:

Theorem 4 (Correctness of the Algorithm) *If $q_\mathcal{A}^0 \in x_C^{(\infty,\infty)}$, then $\mathcal{A}_\mathcal{C}$ is forceable nonblocking and the generated supervisor $\mathcal{A}_\mathcal{C}$ is a valid solution to the controller synthesis Problem.*

Proof: Since $q_\mathcal{A}^0 \in x_C^{(\infty,\infty)}$ holds, we can conclude from Lemma 3 that every reachable state is contained in $x_C^{(\infty,\infty)}$. The first part of the statement now follows from Theorem 3. For the second part, we note that the generated language is necessarily contained in the specification, because of the construction of $\mathcal{A} = \mathcal{A}_\mathcal{P} \times \mathcal{A}_\mathcal{E}$. The forceable nonblocking property follows now from Theorem 1. The controllability property can be seen as follows: Similar to the original supervisor synthesis algorithm, we can be sure that no initial bad state is reached, because we removed those states from the good states and only good states may be visited. On the other hand, we never remove single uncontrollable transitions due to the definition of $move$. Rather, we remove all states that have an uncontrollable transition to a non-good or non-forceable coreachable state in the condition for the good states. Since $x_C^\infty \subseteq x_G^\infty$ and $q_\mathcal{A}^0 \in x_C^\infty$, we can be sure that only good states are visited. ■

We have shown that the above algorithm is correct. To show also its completeness, i.e. that the algorithm generates a controller, whenever a controller exists, we need the next definition and some additional lemmata. According to the definition of forceable coreachability, for every forceable coreachable state, there exists a constant n after which a marked state is unavoidable. Thus, we can define an ordering on the states by taking the minimal constant n for which the forceable coreachable property holds. Thus, we define for every automaton $\mathcal{A}$:

$$F_\mathcal{A}^n = \left\{ q \in Q_\mathcal{A} \;\middle|\; \begin{array}{l} q \text{ is forceable coreachable} \\ \text{with a minimal constant } n \end{array} \right\}$$

Lemma 5 *For every forceable coreachable automaton $\mathcal{A}$ the following holds:*

$$\forall i > 0 \forall q \in F_\mathcal{A}^i. \left(\begin{array}{l} \forall \sigma \in \mathsf{act}_\mathcal{A}(q).\delta_\mathcal{A}(q,\sigma) \in \bigcup_{j<i} F_\mathcal{A}^j \wedge \\ \\ \exists \sigma \in \mathsf{act}_\mathcal{A}(q).\delta_\mathcal{A}(q,\sigma) \in \bigcup_{j<i} F_\mathcal{A}^j \end{array} \right)$$

Proof: Let q in $F_\mathcal{A}^i$. For the first part, assume that there exists $\sigma \in \mathsf{act}_\mathcal{A}(q)$ such that $\delta_\mathcal{A}(q,\sigma) \notin \bigcup_{j<i} F_\mathcal{A}^j$ holds. Then, we can distinguish two cases: if $q' = \delta_\mathcal{A}(q,\sigma)$ is not forceable coreachable, then there exists an infinite string t with $\delta(q',t)\downarrow$ that avoids all marked states. Accordingly, q can not be forceable coreachable, because otherwise all marked states are avoidable by the infinite string σt. On the other hand, if $\delta_\mathcal{A}(q,\sigma)$ is forceable coreachable, but with a constant greater or equal i, then q can not have a minimal constant i. To prove the second part of the lemma, we first note that q can not be marked, because otherwise $q \in F_\mathcal{A}^0$. Therefore, there exists a successor state, because otherwise q can not be coreachable. However, this successor state must be contained in $\bigcup_{j<i} F_\mathcal{A}^j$ according to the proof of the first part. ■

Lemma 6 *If there exists an automaton $\mathcal{A}_\mathcal{C}$ such that $\mathcal{A} \times \mathcal{A}_\mathcal{C}$ is forceable nonblocking and $\mathcal{A}_\mathcal{C}$ respects the controllability property with respect to $\mathcal{A}$, then $q_\mathcal{A}^0 \in x_C^{(\infty,\infty)}$.*

Proof:

Since $\mathcal{A} \times \mathcal{A}_\mathcal{C}$ is forceable nonblocking, every reachable state is forceable coreachable, therefore contained in some $F_{\mathcal{A}\times\mathcal{A}_\mathcal{C}}^i$. We will show by induction on i:

$$\text{if } (p,q) \in F_{\mathcal{A}\times\mathcal{A}_\mathcal{C}}^i \text{ for some } i, \text{ then } q \in x_C^{(i,\infty)}$$

The above lemma follows then from the fact that the initial state $(q_\mathcal{A}^0, q_{\mathcal{A}_\mathcal{C}}^0)$ must be forceable nonblocking and therefore contained in some $F_{\mathcal{A}\times\mathcal{A}_\mathcal{C}}^i$.

Inductive Base: $i = 0$. Then (p,q) is marked and we are done.

Inductive Step: Let $(p,q) \in F_{\mathcal{A}\times\mathcal{A}_\mathcal{C}}^{i+1}$. Then according to lemma 5 the following holds:

$$\forall \sigma \in \mathsf{act}_{\mathcal{A}\times\mathcal{A}_\mathcal{C}}((p,q)).\delta_{\mathcal{A}\times\mathcal{A}_\mathcal{C}}((p,q),\sigma) \in \bigcup_{j<i} F_{\mathcal{A}\times\mathcal{A}_\mathcal{C}}^j$$

Since the controllability property holds, we have that every uncontrollable event in q is also active in (p,q). Therefore

$$\forall \sigma \in \mathsf{act}_\mathcal{A}(q) \cap \Sigma_u.\delta_{\mathcal{A}\times\mathcal{A}_\mathcal{C}}((p,q),\sigma) \in \bigcup_{j<i} F_{\mathcal{A}\times\mathcal{A}_\mathcal{C}}^j$$

It now follows from the inductive hypothesis and the determinacy of $\mathcal{A}$, that

$$\forall \sigma \in \mathsf{act}_{\mathcal{A}}(q) \cap \Sigma_u . \delta_{\mathcal{A}}(q,\sigma) \in \bigcup_{j<i} x_C^{(i,\infty)} = x_C^{(i,\infty)}$$

Again considering Lemma 5, we obtain:

$$\exists \sigma \in \mathsf{act}_{\mathcal{A}\times\mathcal{A}_\mathcal{C}}((p,q)) . \delta_{\mathcal{A}\times\mathcal{A}_\mathcal{C}}((p,q),\sigma) \in F^i_{\mathcal{A}\times\mathcal{A}_\mathcal{C}}$$

Therefore, using the inductive hypothesis, we obtain

$$\exists \sigma \in \mathsf{act}_{\mathcal{A}\times\mathcal{A}_\mathcal{C}}((p,q)) \subseteq \mathsf{act}_{\mathcal{A}}(q) . \quad \delta_{\mathcal{A}}(q,\sigma) \in \bigcup_{j<i} x_C^{(j,\infty)} = x_C^{(i,\infty)}$$

Now either (q,σ) is added to $move^{i+1}$, or there exists another move (q,σ') that has been already added to $move$. In both cases, we have $q \in x_C^{(i+1,\infty)}$. ∎

We are now ready to show completeness of the algorithm:

Theorem 5 (Completeness of the Algorithm) *Given a plant $\mathcal{A}_\mathcal{P}$ and a specification $\mathcal{A}_\mathcal{E}$ where the controller synthesis problem is solvable. Then, $x^0_\mathcal{A} \in x_C^{(\infty,\infty)}$, i.e. the presented algorithm generates a valid controller.*

Proof: Let $\mathcal{A}_\mathcal{C}$ be an automaton that solves the controller synthesis problem. Then, necessarily $\mathsf{L}(\mathcal{A}_\mathcal{C} \times \mathcal{A}_\mathcal{P}) \subseteq \mathsf{L}(\mathcal{A}_\mathcal{E})$ holds as well as $\mathsf{L_m}(\mathcal{A}_\mathcal{C} \times \mathcal{A}_\mathcal{P}) \subseteq \mathsf{L_m}(\mathcal{A}_\mathcal{E})$. $\mathcal{A}_\mathcal{C} \times \mathcal{A}_\mathcal{P}$ is forceable nonblocking. Therefore, $\mathcal{A}_\mathcal{C} \times \mathcal{A}_\mathcal{P} \times \mathcal{A}_\mathcal{E} = \mathcal{A}_\mathcal{C} \times \mathcal{A}$ is forceable nonblocking. According to the definition of controller synthesis problem, $\mathcal{A}_\mathcal{C}$ needs to be controllable with respect to $\mathcal{A}_\mathcal{E}$. Therefore, $\mathcal{A}_\mathcal{C}$ must be also controllable with respect to $\mathcal{A}_\mathcal{P} \times \mathcal{A}_\mathcal{E} = \mathcal{A}$. The statement follows now from Lemma 6. ∎

5 CONCLUSION

In this paper, we have developed an algorithm for the generation of valid controllers from a supervisory control model as used in the Ramadge-Wonham framework. To this end, we have strengthened the coreachability property in order to guarantee that a marked state is eventually reached, irrespective of the plant's behavior. We have proved the correctness and the completeness of our algorithm. In the future, we plan to implement our Algorithm on top of our toolset Averest (Averest, 2005) to evaluate the runtime behaviour of the algorithm.

REFERENCES

Accellera (2004). PSL/Sugar. http://www.haifa.il.ibm.com/projects/verification/sugar.

Averest (2005). www.averest.org.

Büchi, J. (1960a). On a decision method in restricted second order arithmetic. In Nagel, E., editor, *International Congress on Logic, Methodology and Philosophy of Science*, pages 1–12, Stanford, CA. Stanford University Press.

Büchi, J. (1960b). Weak second order arithmetic and finite automata. *Z. Math. Logik Grundlagen Math.*, 6:66–92.

Dietrich, P., Malik, R., Wonham, W., and Brandin, B. (2002). Implementation considerations in supervisory control. In B. Caillaud, P. Darondeau, L. Lavagno, and X. Xie, editors, Synthesis and control of discrete event systems, pages 185–201. Kluwer Academic Publishers.

Emerson, E. (1990). Temporal and modal logic. In *Handbook of Theoretical Computer Science*, volume B, chapter Temporal and Modal Logics, pages 996–1072. Elsevier.

Emerson, E. and Clarke, E. (1982). Using branching-time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming*, 2(3):241–266.

Kozen, D. (1983). Results on the propositional μ-calculus. *Theoretical Computer Science*, 27:333–354.

Malik, P. (2003). *From Supervisory Control to Nonblocking Controllers for Discrete Event Systems*. PhD thesis, University of Kaiserslautern, Kaiserslautern, Germany.

Pnueli, A. (1977). The temporal logic of programs. In *Symposium on Foundations of Computer Science (FOCS)*, volume 18, pages 46–57, New York. IEEE Computer Society.

Ramadge, P. and Wonham, W. (1987). Supervisory control of a class of discrete event processes. *SIAM Journal of Control and Optimization*, 25(1):206–230.

Schneider, K. (2003). *Verification of Reactive Systems – Formal Methods and Algorithms*. Texts in Theoretical Computer Science (EATCS Series). Springer.

Thomas, W. (1990). Automata on infinite objects. In *Handbook of Theoretical Computer Science*, volume B, chapter Automata on Infinite Objects, pages 133–191. Elsevier.

Wonham, W. (2001). Notes on control of discrete-event systems. Technical Report ECE 1636F/1637S 2001-02, Department of Electrical and Computer Engineering, University of Toronto.

Ziller, R. and Schneider, K. (2003). A generalized approach to supervisor synthesis. In *Formal Methods and Models for Codesign (MEMOCODE)*, pages 217–226. Mont Saint-Michel, France. IEEE Computer Society.

AN UNCALIBRATED APPROACH TO TRACK TRAJECTORIES USING VISUAL–FORCE CONTROL

Jorge Pomares, Gabriel J. García, Laura Payá, Fernando Torres
Physics, Systems Engineering and Signal Theory Department
University of Alicante, Alicante, Spain
{jpomares, Fernando.Torres, laura.paya}@ua.es

Keywords: Force control, image-based control, autocalibration.

Abstract: This paper proposes the definition of a new adaptive system that combines visual and force information. At each moment, the proportion of information used from each sensor is variable depending on the adequacy of each sensor to control the task. The sensorial information obtained is processed to allow the use of both sensors for controlling the robot and avoiding situations in which the control actions are contradictory. Although the visual servoing systems have certain robustness with respect to calibration errors, when the image-based control systems are combined with force control we must accurately know the intrinsic parameters. For this purpose an adaptive approach is proposed which updates the intrinsic parameters during the task.

1 INTRODUCTION

Image-based visual servoing is now a well-known approach for positioning the robot with respect to an object observed by a camera mounted at the robot end-effector (Hutchinson et al., 1996). However, in applications in which the robot must interact with the workspace, the visual information must be combined with the sensorial information obtained from the force sensor. A great number of approaches employed for fusing the information obtained from both sensors have been based, up to now, on hybrid control. Concerning hybrid visual-force systems, we should mention studies like (Baeten and De Schutter, 2002) which extend the "task frame" formalism (Bruyninckx and De Schutter, 1996). In (Namiki et al., 1999) a system for grasping objects in real time, which employs information from an external camera and that obtained from the force sensors of a robotic hand, is described. Another strategy used for the combination of both sensorial systems is the use of impedance control. Based on the basic scheme of impedance control, we should mention several modifications like the one described in (Morel et al., 1998), which adds an external control loop that consists of a visual controller which generates the references for an impedance control system. In works such as (Tsuji et al., 1997), the use of virtual forces applied to approaching tasks without contact, is proposed.

In this paper we are not interested in image processing issues, so that the tracked target is composed of four grey marks which will be the extracted features during the tracking. This paper proposes the definition of a new adaptive system which combines visual and force information. Similar approaches has been developed in works such as (Baeten et al., 2002; Olson et al., 2002) however these approaches do not consider the possibility of both sensors providing contradictory information at a given moment of the task. Thus, in unstructured environments it can happen that the visual servoing system establishes a movement direction that is impossible according to the interaction information obtained from the force sensor. In this paper, we consider this possibility and the sensory information obtained is processed to allow the use of both sensors for controlling the robot.

An original aspect of the proposed system is that the proportion of information used from each sensor is variable and depends on the criterion described in Section 4. At each moment, this criterion provides information about the sensor more adequate to develop the task.

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 103–108.

This paper is organized as follows: The main characteristics of the trajectory to be tracked and the notation used is described in Section 2. Section 3 shows the way in which the tracking of the trajectory in the image is carried out. In Section 4, the strategy used for fusing force information with that from the visual servoing system is described. Section 5 describes how the fusion system manages situations in which contradictory control actions are obtained from both sensorial systems. The autocalibration system employed to update the intrinsic parameters is described in Section 6. In Section 7, experimental results, using an eye-in-hand camera, confirm the validity of the proposed algorithms. The final section presents the main conclusions arrived at.

2 NOTATION

In this paper, the presence of a planner, which provides the robot with the 3-D trajectory, γ(t), to be tracked (i.e., the desired 3-D trajectory of the camera at the end-effector), is assumed. These trajectories are generated from a 3-D geometric model of the workspace, so that it is necessary to employ a visual servoing system that performs the tracking of the 3-D trajectory using visual information and, at the same time, tests whether it is possible to carry out such tracking, depending on the interaction forces obtained.

By sampling γ(t) (with period T), a sequence of N discrete values is obtained, each of which represents N intermediate positions of the camera ${}^{k}\boldsymbol{\gamma}/k \in 1...N$. From this sequence, the discrete trajectory of the object in the image $S = \left\{ {}^{k}\mathbf{s}/k \in 1...N \right\}$ can be obtained, where ${}^{k}\mathbf{s}$ is the set of M point or features observed by the camera at instant k, ${}^{k}\mathbf{s} = \left\{ {}^{k}\boldsymbol{f}_i/i \in 1...M \right\}$. As we have previously indicated, in this paper we are not interested in image processing issues, therefore, the tracked target is composed of four grey marks whose centres of gravity will be the extracted features (see Section 7).

The following notations are used. The commanded velocity for the visual servoing and for the force control systems are $\boldsymbol{v}_V^C$ and $\boldsymbol{v}_F^C$ respectively. $\boldsymbol{F}$ (f_x, f_y, f_z, n_x, n_y, n_z) are force (N) and torque (N m) exerted by the environment onto the robot and k is the tool stiffness (N m or N m rad^{-1}). λ_V and λ_F are the proportional control gains for the visual and force controllers respectively.

3 VISUAL TRACKING OF TRAJECTORIES

Each sample, ${}^{k}\mathbf{s}$, is generated from each position ${}^{k}\boldsymbol{\gamma}$. These positions are obtained considering that the time between two consecutive samples is constant, so that $\Delta^{k+1}t = {}^{k+1}t - {}^{k}t = T$ where T is the video rate. The desired trajectory to be tracked in the image is obtained using a natural cubic B-spline (the spline interpolation problem is states as: given image points $S = \left\{ {}^{k}\mathbf{s}/k \in 1...N \right\}$ and a set of parameter values $\Gamma = \left\{ {}^{k}t/k \in 1...N \right\}$ we want a cubic B-spline curve $\mathbf{s}(t)$ such that $\mathbf{s}(t_k)={}^{k}\mathbf{s}$):

$$\mathbf{s}_d(t) = {}^{k}\mathbf{A}t^3 + {}^{k}\mathbf{B}t^2 + {}^{k}\mathbf{C}t + {}^{k}\mathbf{D} \tag{1}$$

where ${}^{k}\mathbf{A}, {}^{k}\mathbf{B}, {}^{k}\mathbf{C}, {}^{k}\mathbf{D}$ are obtained from the samples in the image space at the given instants.

To perform the tracking of the desired trajectory in the image space, an image-based control scheme to regulate to 0 the following vision-based task function is used (Mezouar and Chaumette, 2002):

$$\boldsymbol{e} = \hat{\mathbf{J}}_f^+ \cdot \left(\mathbf{s} - \mathbf{s}_d(t) \right) \tag{2}$$

where $\mathbf{s}$ are the extracted features from the image and $\hat{\mathbf{J}}_f^+$ is an estimation of the pseudoinverse of the interaction matrix. To carry out the tracking of the trajectory, the following velocity must be applied to the robot (with respect to the coordinate frame located at the eye-in-hand camera):

$$\boldsymbol{v}_V^C = -\lambda_V \cdot \boldsymbol{e} + \hat{\mathbf{J}}_f^+ \cdot \frac{\partial \mathbf{s}_d(t)}{\partial t} \tag{3}$$

where $\lambda_V > 0$ is the gain of the proportional controller.

4 FUSION VISUAL-FORCE CONTROL

Up to now, the majority of approaches for fusing visual and force information are based on hybrid control. Only recently (Baeten et al., 2002) has it been possible to find studies on the control of a given direction using force and vision simultaneously (shared control). These approaches are based on the "task frame" formalism

(Bruyninckx and De Schutter, 1996). These works suppose the presence of a high level descriptor of the actions to be carried out in each direction of the work-space at each moment of the task. Thus, the geometric properties of the environment must be known previously. The approach described in this section does not require specifying the sensorial systems to be used for each direction. Furthermore, the proportion of information used from each sensor depends on the criterion described in this section.

The GLR algorithm (Generalized Likelihood Ratio) (Willsky and Jones, 1976) applied to the obtained forces is employed for fusing visual and force information (the setup of the different parameters of the GLR can be seen in our previous works (Pomares and Torres, 2005)). If a given task consists of using visual and force information for maintaining a constant contact with a surface, when the value of GLR increases, this can obtained when, for several possible reasons (irregularities in the surface, errors in the trajectory generated by the visual servoing system, high velocity, etc.) the tracking is not correctly done and, therefore, the system cannot maintain a constant force on the surface. The behaviour is then more oscillatory, and changes are generated in the interaction forces, increasing the value of GLR. To correct this behaviour, the proportion of information used from the force sensor can be augmented when the value of GLR increases, as described below.

The final control action, $\boldsymbol{v}^{C}$, will be a weighted sum obtained from the visual servoing system, $\boldsymbol{v}_{V}^{C}$, and from the force sensor, $\boldsymbol{v}_{F}^{C} = \lambda_{F} \cdot (\boldsymbol{F} - \boldsymbol{F}_{d})/k$, so that $\boldsymbol{v}^{C} = p_{V} \cdot \boldsymbol{v}_{V}^{C} + p_{F} \cdot \boldsymbol{v}_{F}^{C}$. Depending on the value of GLR, we obtain the following control actions:

GLR≤U_1. Normal functioning of the system. In this case, both control actions are weighted with the same proportion (empirically U =500 is obtained):

$$\boldsymbol{v}^{C} = 0,5 \cdot \boldsymbol{v}_{V}^{C} + 0,5 \cdot \boldsymbol{v}_{F}^{C} \qquad (4)$$

$U_1 \leq GLR \leq U_2$. Range of values of GLR that can be obtained when a change in the surface begins or when the system works incorrectly (empirically U_2=1000). In this case, the weight applied to the control action corresponding to the visual servoing system is reduced with the aim of correcting defects in the tracking. Before describing the weight function for this range of GLR, two parameters that characterize this function, are defined. These parameters (p_1, p_2) identify the velocity range that the visual servoing system can establish for different values of GLR. Thus, when GLR is equal to U_1, or lower, the velocity established by the computer vision system will be $\boldsymbol{v}_{Vmax}^{C} = -\frac{\lambda_V}{2} \cdot \boldsymbol{e} + \widehat{\mathbf{J}}_{f}^{+} \cdot \frac{\partial \mathbf{s}_{d}(t)}{\partial t}$, that is to say, the normal velocity defined to carry out the tracking of the trajectory in the image space. In the previous expression, we can see the term $\lambda_V/2$ due to the weight in the control action obtained from the computer vision system, $\boldsymbol{v}_{V}^{C}$, in the global control action, $\boldsymbol{v}^{C}$, that is to say, p_1=0,5 (see Equation (4)). However, when GLR is equal to U_2, we define $\boldsymbol{v}_{Vmin}^{C} = -\lambda_V \cdot p_2 \cdot \boldsymbol{e} + \widehat{\mathbf{J}}_{f}^{+} \cdot \frac{\partial \mathbf{s}_{d}(t)}{\partial t} < \boldsymbol{v}_{Vmax}^{C}$ as the minimum velocity, empirically obtained, to carry out the tracking of the trajectory and which allows the system to correct the possible defects in this trajectory (the effect of the force control in the trajectory is increased in the global control action). Thus, the value of the weight associated with the velocity provided by the visual servoing system, will be $p_2 = 0,5 \cdot \frac{\boldsymbol{v}_{Vmin}^{C}}{\boldsymbol{v}_{Vmax}^{C}}$. Therefore, considering a decreasing evolution of the weight function applied to the velocity obtained from the visual servoing system, this function will have the following value in the range $U_1 \leq GLR < U_2$:

$$p_v = \frac{p_2 - p_1}{U_2 - U_1} \cdot GLR + p_1 - \frac{p_2 - p_1}{U_2 - U_1} \cdot U_1 \qquad (5)$$

Obviously, the weight associated with the force control system will be $p_F = 1 - p_v$.

GLR ≥ U_2. When GLR is in this range, the behaviour established is to continue with the minimum velocity, $\boldsymbol{v}_{Vmin}^{C}$.

5 MANAGING CONTRADICTORY CONTROL ACTIONS

Up to now, the approaches for fusing visual and force information do not consider the possibility of both sensors providing contradictory information at a given moment of the task (the visual servoing system establishes a movement direction that is impossible according to the interaction information obtained from the force sensor).

To assure that a given task in which it is required an interaction with the setting is correctly developed,

the system must carry out a variation of the trajectory in the image, depending on the spatial restrictions imposed by the interaction forces. Therefore, given a collision with the setting and having recognized the normal vector of the contact surface (Pomares and Torres, 2005), the transformation $\mathbf{T}_r$ that the camera must undergo to fulfil the spatial restrictions, is determined. This transformation is calculated so that it represents the nearest direction to the one obtained from the image-based control system, and which is contained in the plane of the surface. Thus, we guarantee that the visual information will be coherent with the information obtained from the force sensor. To do so, considering $\boldsymbol{f}$ to be the position of a given feature extracted by the camera at a given instant, and $[\mathbf{R}_i\ \mathbf{t}_i]$ (rotation and translation) a sampling of the transformation $\mathbf{T}_r$ that the camera undergoes during the tracking of the recognized surface, the feature $\boldsymbol{f}_i^{'}$ extracted in each one of these positions will be:

$$\boldsymbol{f}_i^{'} = \mathbf{A} \cdot \mathbf{R}_i \cdot \mathbf{A}^{-1} \cdot \boldsymbol{f} + \mathbf{A} \cdot \mathbf{t}_i / z \tag{6}$$

where z is the distance between the camera and the object from which the features are extracted and **A** is the following intrinsic parameter matrix:

$$\mathbf{A} = \begin{bmatrix} f \cdot p_u & -f \cdot p_u \cdot \cot(\theta) & u_0 \\ 0 & f \cdot p_v / \sin(\theta) & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Considering the homogeneous image coordinates of a feature $\boldsymbol{f}_i=[u_i,\ v_i,\ 1]$, u_0 and v_0 are the pixel coordinates of the principal point, f is the focal length, p_u and p_v are the magnifications in the *u* and *v* directions respectively, and θ is the angle between these axes.

From the sampling of the desired trajectory in the image, $\boldsymbol{f}_i^{'}$, a spline interpolator is applied to obtain the desired trajectory in the image (see Section 3).

6 AUTOCALIBRATION

It is well known that the visual servoing systems have certain robustness with respect to calibration errors. However, the knowledge of the intrinsic parameters is important when visual and force information is combined, in order to deal with contradictory control actions obtained from both sensorial systems. As can be seen in (6) it is necessary to know **A** for determining the new trajectory in the image once the collision is detected. The matrix **A** is obtained by a previous calibration of the camera using the Zhang's method (Zhang, 2000). However, during the task the intrinsic and extrinsic parameters can be modified. In order to update the camera intrinsic and extrinsic parameters the following method is employed.

We assume that the focal length in *u* and *v* directions differ, denoting f_u, f_v respectively. The estimated camera intrinsic parameters are $P_I = [f_u, f_v, u_0, v_0]$. At a given instant k, using these parameters we obtain a set of features ${}^k\mathbf{s}_I = \left\{ {}^k\boldsymbol{f}_{Ii} / i \in 1...M \right\}$. When the set P_I varies, the derivative of $\mathbf{s}_I$ with respect to the change of the intrinsic parameters is:

$$\dot{\mathbf{s}}_I = \frac{\partial \mathbf{s}_I}{\partial P_I} \cdot \frac{\partial P_I}{\partial t} \tag{8}$$

Considering $\mathbf{s}$ the true features extracted from the image, the error function $\xi = \mathbf{s} - \mathbf{s}_I$ is defined. Therefore:

$$\dot{\xi} = \mathbf{J}_f \mathbf{T} + \frac{\partial \mathbf{s}_I}{\partial P_I} \cdot \frac{\partial P_I}{\partial t} \tag{9}$$

where **T** is the variation with respect the time of the extrinsic parameters, and $\mathbf{J}_f$ the interaction matrix for four points (Marchand and Chaumette, 2002) corresponding to the four features.

As we have previously described, the intrinsic parameters must be known when a collision is detected. When ξ is equal to 0 the intrinsic parameters, P_I, corresponds with the true ones. To make ξ decrease exponentially to 0 we form the feedback loop to this system where the feedback value should be:

$$\begin{bmatrix} \mathbf{T} \\ \dot{P}_I \end{bmatrix} = -k_C \cdot \begin{bmatrix} \mathbf{J}_f & \frac{\partial \mathbf{s}_I}{\partial P_I} \end{bmatrix}^{+} \cdot \xi \tag{10}$$

Therefore, the extrinsic and intrinsic parameters must be determined when a collision occurs. To do so, we move the camera according to the **T** component and the intrinsics with $P_I = P_I + \dot{P}_I$ until ξ is 0. At this moment the true camera parameters will be know and the Equation (6) can be applied to obtain the new image trajectory which must be tracked.

7 RESULTS

In this section, we describe the different tests carried out that show the correct behaviour of the system in disassembly tasks. For the tests we have used an eye-in-hand camera system composed of a JAI-M536 mini-camera in the end-effector of a 7 d.o.f. Mitsubishi PA-10 robot also equipped with a force sensor. The system is able to acquire up to 30 frames/second and is previously submitted to a calibration process (focal length is 7,5 mm). In the experiments described in this paper, the tracked target is composed of four grey marks. During the disassembly the head of the screw is in contact with the guide so that we apply the sensorial fusion to disassemble the screw (see Figure 1).

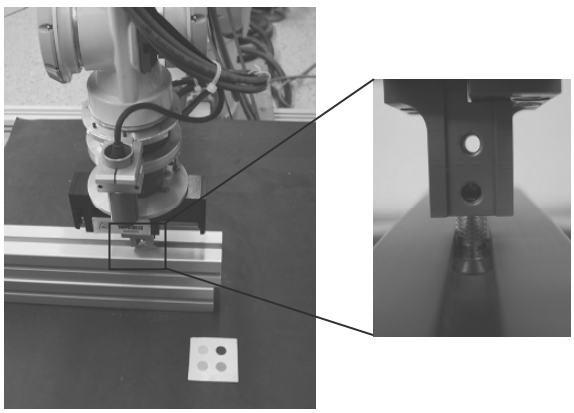


Figure 1: Experimental setup.

Figures 2 and 3 show two experiments for the disassembly of the screw. The first graph of each figure represents the applied force in z direction fusing visual and force information with constant weights. In the second graph the proposed strategy of variable weights is used (see Section 4).

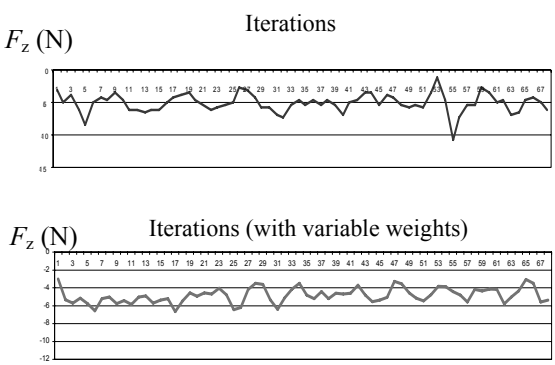


Figure 2: Comparison between the obtained forces without using and using the strategy of variable weights. Experiment 1.

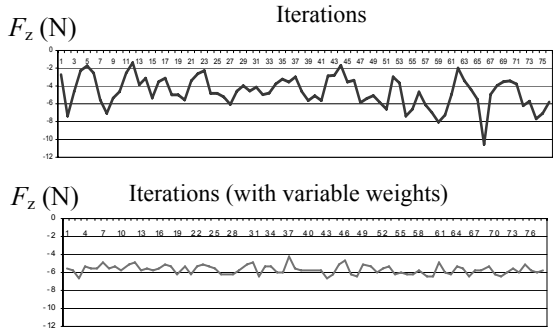


Figure 3: Comparison between the obtained forces without using and using the strategy of variable weights. Experiment 2.

In Figures 2 and 3 we can observe that using the strategy of variable weights the system response is less oscillating. Using this strategy the system allows maintaining the constant contact force between the guide and the head of the screw.

When a collision is detected the system updates the intrinsic parameters to guarantee that the new trajectory is generated correctly. To illustrate the behaviour of the algorithm we show an autocalibration experiment.

Figure 4 shows the evolution of the image error $\xi = \mathbf{s} - \mathbf{s}_I$ for each feature, during the calibration. Once, the error is zero, the correct intrinsic and extrinsic parameters has been obtained.

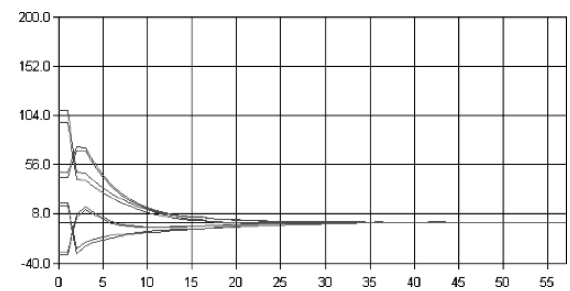


Figure 4: Image error for each feature during the autocalibration.

The convergence of the focal length estimations is shown in Figure 5 (the pixel is almost the same in u and v directions on the image sensor). However, using the autocalibration approach described in Section 6 it is also possible to determine the camera extrinsic parameters, and therefore, to determine the position of the robot during the task. In Figure 6, the virtual camera trajectory during the calibration is shown.

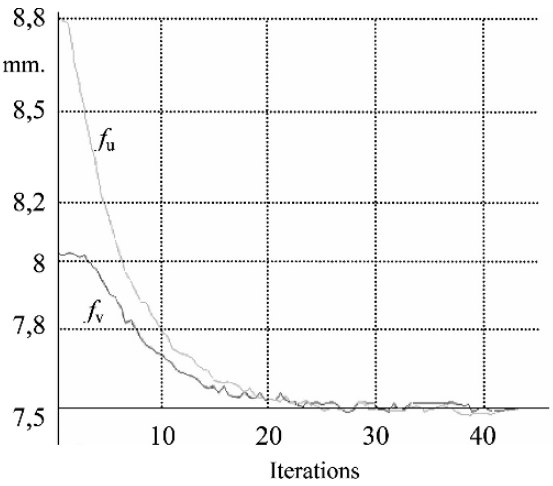


Figure 5: Convergence of the estimated focal lengths.

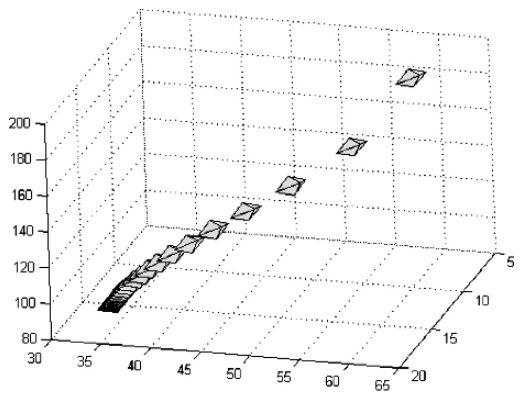


Figure 6: Virtual camera trajectory during the calibration.

8 CONCLUSIONS

We proposed a new method for combining visual and force information which allow us to update the intrinsic parameters during the task by using an autocalibration approach. The visual-force control system has others original aspects which improve the behaviour of the system. Within these aspects we should mention the variable weights applied to each sensor (depending on the GLR parameter) and the possibility of managing contradictory control actions. As the results show, the robot is able to track the image trajectory maintaining a constant force with the workspace using visual and force information simultaneously.

REFERENCES

Baeten, J., De Schutter, J., 2002, Hybrid vision/force control at corners in planar robotic-contour following. IEEE/ASME Transactions on Mechatronics, vol. 7, no. 2, pp. 143–151.

Baeten, J., Bruyninckx, H., De Schutter, J, 2002. Shared control in hybrid vision/force robotic servoing using the task frame. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems.* Lausanne, Suiza, pp. 2128–2133.

Bruyninckx, H., De Schutter, J., 1996. Specification of force-controlled actions in the task frame formalism-A synthesis, IEEE Transactions on Robotics and Automation, vol. 12, no. 4. pp. 581–589.

Hutchinson, S., Hager, G., Corke, P., 1996. A tutorial on visual servo control. IEEE Trans. on Robotics and Automation, vol. 12, no. 5, pp. 651–670.

Marchand, E., Chaumette, F., 2002. Virtual visual servoing: a framework for real-time augmented reality. In *EUROGRAPHICS 2002 Conference Proceeding*, Computer Graphics Forum, Sarrebruck, Germany. vol. 21, no. 3, pp. 289–298.

Mezouar, Y., Chaumette, F., 2002. Path planning for robust image-based control. IEEE Transactions on Robotics and Automation, vol. 18, no. 4, pp. 534–549.

Morel, G., Malis, E., Boudet, S., 1998. Impedance based combination of visual and force control. In *IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, pp. 1743–1748.

Namiki, A., Nakabo, I., Ishikawa, M., 1999. High speed grasping using visual and force feedback. In *IEEE Int. Conf. on Robotics and Automation*, Detroit, MI, pp. 3195–3200.

Olsson, T., Bengtsson, J., Johansson, R., Malm, H., 2002. Force control and visual servoing using planar surface identification, *In IEEE Int. Conf. on Robotics and Automation*. Washington, USA, pp. 4211–4216.

Pomares, J., Torres, F., 2005. Movement-flow based visual servoing and force control fusion for manipulation tasks in unstructured environments. IEEE Transactions on Systems, Man, and Cybernetics—Part C. vol. 35, no. 1. pp. 4–15.

Tsuji, T., Hiromasa, A., Kaneko, M., 1997. Non-contact impedance control for redundant manipulators using visual information, In *IEEE Int. Conf. on Robotics and Automation*, Albuquerque, USA, vol. 3, pp. 2571–2576.

Willsky, A.S., Jones, H.L., 1976. A generalized likelihood ration approach tothe detection and estimation of jumps in linear systems. IEEE Trans. Automat. Contr., vol. 21, no. 1, pp. 108–112.

Zhang, Z., 2000. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330–1334.

A STRATEGY FOR BUILDING TOPOLOGICAL MAPS THROUGH SCENE OBSERVATION

Roger Freitas, Mário Sarcinelli-Filho and Teodiano Bastos-Filho
Departamento de Engenharia Elétrica, Universidade Federal do Espírito Santo
Vitória - E.S., Brazil
{*roger,mario.sarcinelli,tfbastos*}*@ele.ufes.br*

José Santos-Victor
Instituto de Sistemas e Robótica, Instituto Superior Técnico
Lisbon, Portugal
jasv@isr.ist.utl.pt

Keywords: Learning, Topological Navigation, Incremental PCA, Affordances.

Abstract: Mobile robots remain idle during significant amounts of time in many applications, while a new task is not assigned to it. In this paper, we propose a framework to use such periods of inactivity to observe the surrounding environment and *learn* information that can be used later on during navigation. Events like someone entering or leaving a room, someone approaching a printer to pick a document up, etc., convey important information about the observed space and the role played by the objects therein. Information implicitly present in the motion patterns people describe in a certain workspace is then explored, to allow the robot to infer a "meaningful" spatial description. Such spatial representation is not driven by abstract geometrical considerations but, rather, by the role or function associated to locations or objects (affordances) and learnt by observing people's behaviour. Map building is thus bottom-up driven by the observation of human activity, and not simply a top-down oriented geometric construction.

1 INTRODUCTION

In many applications, mobile robots remain idle for significant amounts of time, while a new task is not assigned to it. Similarly, in many research labs mobile robots remain inactive during extended periods of time, while new sensorial information processing or navigation algorithms are being tested.

The motivation of this work is to use those periods of inactivity to observe the surrounding environment and *learn* information that can be used later on during navigation. For example, events like someone entering or leaving a room or approaching a printer to pick a document up, convey important information about the observed space and the role played by the objects therein.

The development of algorithms to extract useful information from the observation of such events could bring significant savings in programming, while affording the robot with an extended degree of flexibility and adaptability. In this work, we explore the information implicitly present in the motion patterns people describe in a certain workspace, to allow the robot to infer a "meaningful" spatial description. Interestingly, such spatial representation is not driven by abstract geometrical considerations but, rather, by the role or function associated to locations or objects and learnt by observing people's behaviour.

The mobile robot we use in this work combines peripheral and foveal vision. The peripheral vision is implemented by an omnidirectional camera that captures the attention stimuli to drive a standard, narrow field of view pan-tilt (perspective) camera (foveal vision).

Other research groups have used information associated to people's trajectories to help robot navigation. In (Bennewitz et al., 2002) mobile robots equipped with laser sensors are used to extract trajectories of people moving in houses and offices. The trajectories are estimated using the Expectation-Maximization (EM) algorithm and the models are used to predict human trajectories in order to improve people following. In (Bennewitz et al., 2003) the same authors propose a method for adapting the behavior of a mobile robot according to the activities of the people in its surrounding. In (Kruse and Wahl, 1998) an off-board camera-based monitoring system is proposed to help mobile robot guidance. In (Appenzeller et al., 1997) it is developed a system that builds topological maps by looking at peo-

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 109–115

ple. Their approach is based on cooperation between *Intelligent Spaces* (Fukui et al., 2003) and robots. *Intelligent Spaces* are environments endowed with sensors like video cameras, acoustic sensors, pressure sensors, monitors and speakers that send information about the environment to a central processing system. Usually, the beings present in the environment are human beings and, in some cases, robots. From the analysis of the sensorial data, the *Intelligent Space* can supply the "users" with necessary information to accomplish some task. For example, this kind of environment is able to build maps and send them to the robots, allowing them to navigate safely. However, this approach is characterized by low scalability, i.e., if the robot is supposed to navigate in a different environment, such environment should be structured *a priori*.

Our approach to this problem is to extract the motion patterns of people from the robot's viewpoint directly, using an on-board vision system. The advantage of such approach is that the robot can *learn* from environments that are not structured for this purpose, thus giving to the learning process more flexibility and scalability. However, the robot cannot observe the entire environment at once, which is a limitation that can be overcome by using an incremental learning strategy. Such a strategy allows the robot to observe the environment from an initial position and to create a partial model representing the observed region. Then, starting from this initial model, the robot may change its position in the environment, and to keep observing it from the new position. From the new observations, the initial model could be validated, changed or enlarged.

The implementation of the incremental learning process is based on an incremental algorithm of Principal Component Analysis (PCA). An incremental algorithm that is based on (Murakami and Kumar, 1982; Hall et al., 1998; Artač et al., 2002) is here adopted. The omnidirectional images that are captured by the robot during the learning process will represent the nodes of a topological map of the environment. The incremental PCA (IPCA) algorithm allows the integration of new images (new nodes) in an online way. This incremental approach, in conjunction with the strategy of observing people's movements, will give the robot a high level of autonomy on building maps, while extracting information that allows the perception of some functionalities associated to specific regions of the environment.

Such topics are hereinafter addressed in the following way: Section 2 describes the overall learning system, and preliminary results are shown in Section 3. Section 4 describes the approach to enlarge the partial map created through observation, and in Section 5 some conclusions and discussions about possible developments are presented.

2 OVERALL LEARNING SYSTEM

Figure 1 shows a scheme of our overall approach. The most important subsystems, which embed increasing level of cognition, are the vision, measurement and modeling subsystems.

The *Vision System* comprises peripheral and foveal visual capabilities. Peripheral vision is accomplished by an omnidirectional camera and is responsible for detecting movement. Foveal vision is accomplished by a perspective camera that is able to execute pan and tilt rotations, and is responsible for tracking moving objects.

The *Measurement System* is responsible for transforming visual information into features the robot is trying to learn, e.g., transforming 2D image information into trajectory points on the floor, referred to a common coordinate frame.

The *Modeling System* is responsible for building models that explain data from the measurement system. This system operates in two different levels of cognition, labelled geometric level and temporal level. The geometric level modeling system outputs strictly geometric models. The temporal level modeling system outputs models that incorporate concepts like temporal analysis and *appearance*. Depending on the kind of model the robot is trying to build, this system could also drive the way the vision system operates (e.g. controlling the gaze direction).

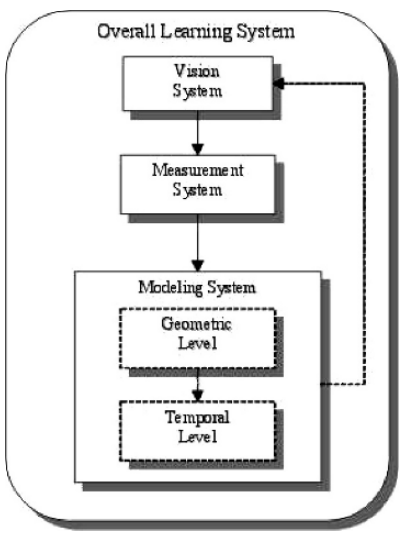


Figure 1: The Overall Learning System.

In this paper, we use the scheme shown in Figure 1 to learn possible trajectories and interesting places in the environment surrounding the robot. In this case, the Measurement System is responsible for transforming 2D image information into trajectory points on the floor. The Modeling System is responsible for

building models of possible trajectories and/or finding interesting places in the environment that should be investigated in more detail (low level modeling).

We assume that the robot has no *prior* knowledge about the structure of the working environment. From any position inside it, the robot should extract useful information to navigate. In order to do that, it should be able to detect moving objects, track these objects and transform this information into possible trajectories (a set of positions in an external coordinate system) to be followed. In the following subsections, we describe in detail each one of these subsystems.

2.1 Vision System

The vision system deals with two types of visual information: peripheral and foveal (see Figure 2). The peripheral vision uses an omnidirectional camera to detect interesting image events and to drive the attention of the foveal camera. The foveal vision system is then used to track the objects of interest, using a perspective camera with a pan-tilt platform.

2.1.1 Attention System

The attention system operates on the omnidirectional images and detects motion of objects or people in the robot vicinity. Other visual cues could be considered, but in the current implementation we deal exclusively with motion information. Motion detection can be easily performed by using background subtraction. Moving objects are detected by subtracting the current image from the background image (previously obtained). In this work, the background is modeled using the method proposed in (Gutchess et al., 2001), which uses a sequence of images taken from the same place and outputs a statistical background model describing the static parts of the scene.

Figure 3 shows an omnidirectional image taken in the laboratory and the result of movement detection. Once the movement is detected, a command is sent

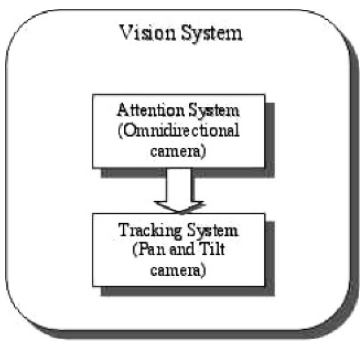


Figure 2: The Vision System.

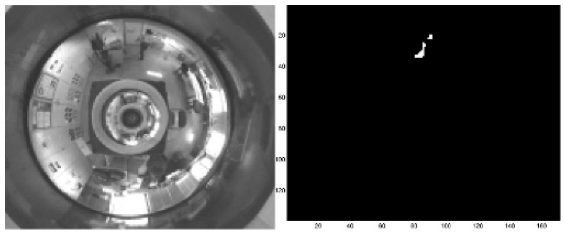


Figure 3: Omnidirectional image captured (a), movement detection (b).

to the pan and tilt camera to drive its gaze direction towards the region of interest and to start tracking the moving object. To direct the camera gaze towards the detected target, we would need to determine the required camera pan and tilt angles. The camera pan angle must be set to the angular position of the target in the omnidirectional image. To determine the tilt angle, we would need to determine the distance to the detected target. Instead, for simplification we always use a reference tilt position that roughly points the camera towards the observed region.

2.1.2 Tracking System

Whenever the Tracking System is activated, the Attention System is deactivated. We are currently using a simple tracking algorithm to illustrate the idea of learning about the environment from observing human actions. The next step is to improve its performance and robustness.

The current tracking routine takes two consecutive images as the input and extracts the pixels displaying some change. The result is that different regions (moving objects) in the two images are highlighted. Then, we calculate a bounding box around the detected area. The point to be tracked is the middle point of the bottom edge of the bounding box (theoretically a point on the floor).

While operating, the system is continuously detecting regions of interest in the peripheral field of view. The foveal vision system then tracks these objects, while they remain visible. If the target is not visible anymore, the Attention System is made active again. The measurement system described in the sequence will integrate the information of different tracked objects into a common coordinate system, from where more global information can be interpreted.

2.2 Measurement System

In order to estimate trajectories relative to the robot, it is necessary to estimate the distance from the robot to the moving object in each image acquired. Usually, this problem is solved using two or more cam-

eras set in different places and applying stereo vision techniques.

As the robot is stationary while observing the environment, consecutive images of a given moving object differ only by camera rotations (pan and tilt). Thus, stereo can not be used to reconstruct the 3D trajectory of the target. The alternative used to solve this problem is to estimate the homography **H** between the floor and the image plane, i.e., to find an *a priori* plane projective transformation that transforms an image point *(u,v)* into a point on the floor *(X,Y,1)*, or

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{H} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}, \quad (1)$$

where **H** is the 3×3 homography matrix. Initially, the homography is estimated using a set of ground plane points, whose 3D positions are known with respect to some reference frame. Then, when the foveal camera moves, the homography is updated as a function of the performed motion. So, as the camera is tracking the object, its pose is changing, and the same happens to the homography between the image plane and the floor. For this reason, we use the pan and tilt angles to update the homography (see Figure 4).

We assume that the intrinsic parameters of the pan-tilt camera are known *a priori*, after an initial calibration step. The intrinsic parameters are used to decompose the homography matrix into a rotation matrix and a displacement vector (camera pose) relating the camera frame to a world frame. Pan and tilt angles generate canonical rotation matrices that multiply the original rotation matrix, thus updating the homography.

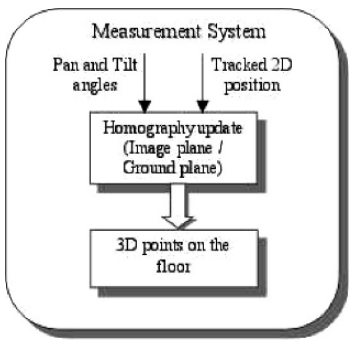


Figure 4: The Measurement System.

In order to recover camera pose, we apply the methodology presented in (Gracias and Santos-Victor, 2000), which we briefly describe next. The homography, **H**, can be written as

$$\mathbf{H} = \lambda \mathbf{K} \mathbf{L} \quad (2)$$

where λ is an unknown scale factor, **K** is the camera intrinsic parameter matrix and **L** is a matrix composed from the full (3 x 3) rotation matrix **R** and the translation vector **t**. Hence,

$$\mathbf{L} = [\ \overline{\mathbf{R}} \quad \mathbf{t}\ ], \quad (3)$$

where $\overline{\mathbf{R}}$ is a 3x2 submatrix comprising the first two columns of matrix **R**. Due to noise in the estimation process, homography **H** will not follow exactly the structure of (2). Alternatively, using the Frobenius norm to measure the distance between matrices, the problem can be formulated as

$$\lambda, \mathbf{L} = \arg \min_{\lambda, \mathbf{L}} \|\lambda \mathbf{L} - \mathbf{K}^{-1}\mathbf{H}\|^2_{frob} \quad (4)$$

subject to $\overline{\mathbf{L}}^T \overline{\mathbf{L}} = \mathbf{I}_2$, where $\overline{\mathbf{L}}$ is a 3x2 submatrix comprising the first two columns of **L**. The solution of (4) can be found through Singular Value Decomposition (SVD). Let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be the SVD of $\mathbf{K}^{-1}\mathbf{H}$. Then, $\overline{\mathbf{L}}$ is given by

$$\overline{\mathbf{L}} = \mathbf{U}\mathbf{V}^T, \quad (5)$$

and

$$\lambda = \frac{tr(\mathbf{\Sigma})}{2}. \quad (6)$$

The last column of **L**, namely **t**, can be found as

$$\mathbf{t} = \mathbf{K}^{-1}\mathbf{H} \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\lambda} \end{bmatrix}, \quad (7)$$

thus resulting

$$\mathbf{L} = [\ \overline{\mathbf{L}} \quad \mathbf{t}\ ]. \quad (8)$$

The last column of rotation matrix **R** can be found by computing the cross product of the the first two columns. The updated rotation matrix is given by

$$\mathbf{NewR} = \mathbf{R} \cdot \mathbf{R}_{PAN} \cdot \mathbf{R}_{TILT}. \quad (9)$$

Finally, the updated homography is then

$$\mathbf{NewH} = \lambda \cdot \mathbf{K} \cdot \mathbf{NewL}, \quad (10)$$

where

$$\begin{aligned} \mathbf{NewL} &= [\ \overline{\mathbf{NewR}} \quad \mathbf{Newt}\ ] \\ \mathbf{Newt} &= \mathbf{NewR} \cdot \mathbf{t}. \end{aligned}$$

We have now a way to project all tracked trajectories onto a common coordinate system associated to the ground plane. In this global coordinate system, the different trajectories described by moving objects can be further analyzed and modeled, as described in the next subsection.

2.3 Modeling System

The modeling system is responsible for building models explaining data emerging from the measurement system. Depending on the nature of the models the robot is building, this system can drive the way vision system operates. This system can operate in two different levels of cognition:

- *geometric level* - the geometric level modeling system outputs strictly geometric models, e.g., metric trajectories that could be followed by the robot;
- *temporal level* - the temporal level modeling system outputs models that incorporate a temporal analysis as well as concepts like *appearance*, e.g., images representing regions of the environment can be associated to a spacial description the robot can use to navigate (topological maps);

2.3.1 Geometric Level

In this work, the modeling system operates on geometric level, once it aims to interpret the observed (global) trajectories onto representations that can be used for navigation. Currently, we consider three main uses of such data:

- the observed trajectories correspond to free (obstacle free) pathways that the robot may use to move around in the environment;
- regions where trajectories start or end might correspond to some important functionality (e.g. doors, tables, tools, etc) and should be represented in a map;
- if many trajectories meet in a certain area, it means that that region must correspond to some important functionality as well.

Hence, from observation the robot can learn the location of interesting places in the scene and the most frequent ways to go from one point to another. Moving further, the robot also might be able to distinguish uncommon behaviours, what could be used in surveillance and monitoring tasks.

2.3.2 Temporal Level

The next step in the modeling process would be the addition of a temporal analysis of the events that occur while the robot observes the scene. Concepts like appearance are incorporated in the model as well. Appearance is often used to solve the problem of mobile robot localization based on video images (Gaspar et al., 2000). Rather than characterizing from strictly known geometric features, the approach is to rely on appearance-based methods and a temporal analysis to enrich the model of the environment. The temporal analysis will allow the characterization of pathways, as well as regions where people usually stop and stay for periods of time while engaged in some activity.

3 EXPERIMENTAL RESULTS

We performed preliminary experiments in the laboratory to verify the performance of the Vision, Measurement and Modeling systems. The robot stayed observing the laboratory while people walked by, along different trajectories. Each trajectory was performed and recorded separately. The positions on the floor, measured by the system, are shown in Figure 5.

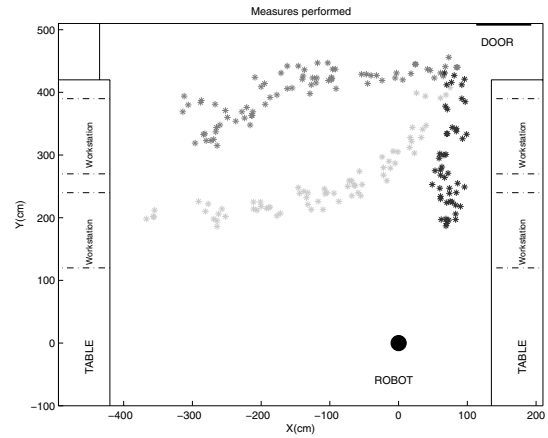


Figure 5: Real data measured from observing people's movements.

The data generated by the Measurement System is then interpreted by the Modeling System. When analyzing the data shown in Figure 5, the most interesting point is the kind of information that can be extracted from such data. One could try, for example, to extract models of observed trajectories. In this case, the model could be obtained statistically (Bennewitz et al., 2002) or deterministically. In the deterministic case, local (e.g. splines) or global (e.g. polynomial) models could be used.

To illustrate the idea, the trajectories shown in Figure 6 were modeled using a linear polynomial model. Places of interest can be detected as well (see Figure 6). In this case, we applied a threshold on the data shown in Figure 5 based on the number of times a po-

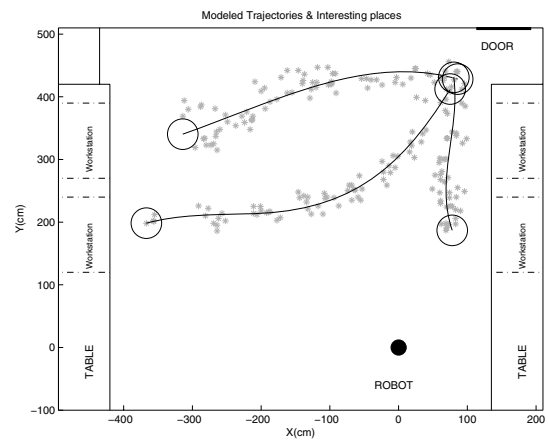


Figure 6: Examples of modeled trajectories and places of interest.

sition was visited. This is done in order to filter the data, thus discarding positions that are not frequently visited. Then, we use a k-means algorithm to cluster the remaining data.

By identifying these places, a strategy for modeling and identification can be derived, thus providing an autonomous way of learning models for such places. For example, as we can see in Figure 6, three of such places of interest appear in front of workstations in the laboratory.

4 ENLARGING THE MAP

The experimental results obtained suggest that, from its initial position, it is unlikely that the robot can model correctly all the trajectories and interesting places in the environment. This is expected to happen due to occlusions and the high uncertainty assigned to distant regions.

Trajectory models based on observations made from the robot's initial position are highly affected by occlusions. Besides the incorrect models, occlusions can lead to a misclassification of the regions of the environment labelled as "interesting places." For example, from the viewpoint of the robot, one can describe (or model) the region where a door is placed as a region where people usually appear and disappear. In most cases, and if occlusions are not present, such a description would suffice to correctly distinguish the object door from other "interesting places" in the environment. However, if occlusions are present, the trajectory points where they occur would be incorrectly modelled as regions corresponding to doors.

From these considerations, it can be concluded that it is necessary that the robot, based on the initial model built from its initial position, changes its position in the environment and restart the observation process, aiming to validate the current model. A strategy that allows the robot to choose the new viewpoint, given the current (and partial) metric map, should now be developed. New measurements could then be compared to the old ones through odometry readings.

Once a trajectory has been validated, the robot could start the topological mapping. The validated trajectory would be followed by the robot, while capturing images and building the map in a incremental way. Each image would be assigned to a map node, representing a position in the environment. The idea consists in representing the robot environment as a topological map, storing a (usually large) set of landmark images. To speedup the comparison of the robot views with these landmark images, it is advantageous to use low-dimensional approximations of the space spanned by the original image set. One example is to use principal component analysis (PCA) that uses the set of input images to extract an orthonormal basis (or model) of a lower dimensional subspace (eigenspace) that approximates the input images.

In the traditional approach to calculate these eigenspace models, known as *batch method*, the robot must capture all the images needed to build the map and then, using either eigenvalue decomposition of the covariance matrix or singular value decomposition of the data matrix, calculate the model. This approach has some drawbacks, however. Since the entire set of images is necessary to build the model, it is impossible to make the robot to build a map while visiting new positions. Update of the existing model is only possible from scratch, which means that original images must be kept in order to update the model, thus requiring a lot of storage capability.

To overcome these problems, some authors (Murakami and Kumar, 1982; Hall et al., 1998) proposed algorithms that build the eigenspace model incrementally (sometimes referred to as subspace tracking in the communications literature). The basic idea behind these algorithms is to start with an initial subspace (described by a set of eigenvectors and associated eigenvalues) and update the model in order to represent new acquired data. This approach allows the robot to perform simultaneous localization and map building. There is no need to build the model from scratch each time a new image is added to the map, thus making easier to deal with dynamic environments. Recently, Artač et al (Artač et al., 2002) improved Hall's algorithm (Hall et al., 1998) by suggesting a way to update the low dimensional projections of the images, thus allowing to discard the image as soon as the model has been updated. Whenever the robot acquires a new image, the first step consists in determining whether or not this image is well represented by the existing subspace model. The component of the new image that is not well represented by the current model is added to the basis as a new vector. Then, all vectors in the basis are "rotated" in order to reflect the new energy distribution in the system. The rotation is represented by a matrix of eigenvectors obtained by the eigenvalue decomposition of a special matrix (see (Freitas et al., 2003) for details).

Through this IPCA algorithm, it is possible to make the transition from geometric to appearance models. The robot will follow the metric trajectory based on odometry, while acquiring images and building the topological map of that trajectory incrementally.

5 CONCLUSIONS AND FUTURE WORK

Currently, the temporal analysis modeling level is under development, and experimental results will be

available soon . A further development of the modeling system could consist of the addition of a *Functional Level*. This level would be associated with the *affordances* of the environment, perceived by the robot. According to Gibson (Gibson, 1979), *"the affordance of anything is a specific combination of the properties of its substance and its surface taken with reference to an animal."* In other words, the term affordance can be understood as the function or role, perceived by an observer, that an object plays in the environment. Such functionalities are quickly perceived through vision, and full tridimensional object models are not always required so that their functionalities in the environment could be perceived.

Even though a robot had a full tridimensional model of the environment and information about the movement of the objects, it wouldn't have a human-like scene vision. When human beings (and animals) observe a scene, they "see" several *possibilities* and *restrictions* (Sloman, 1989), such as possibilities of acquisition of more information through a change in the viewpoint and possibilities of reaching a goal through interaction with objects present in the environment. Hence, Gibson's affordances are closely related to these *possibilities* and *restrictions*. Once the affordances represent a rich source of information to understand the environment, it is important to develop a strategy to identify and extract them from the images captured by the robot. Then, it is possible that the observation of people while executing common tasks reveal some affordances in the environment. For example, one can assign to the doors of an environment the affordance "passage." If the robot could observe people appearing and disappearing in a specific region, it would perceive that region as an access to such an environment.

While the robot is building the map or navigating based on a map previously built, it is likely that the robot faces an object or a person in its way. In order to avoid the collision, it is necessary to develop an obstacle detection algorithm and an obstacle avoidance strategy based on information that can be extracted from images. Besides, an environment inhabited by people is subject to changes in its configuration. If these changes are not detected by the robot and represented in the environment model, the map would not be a correct representation of the environment anymore. Hence, it is also necessary to develop a methodology to detect changes in the environment configuration.

REFERENCES

Appenzeller, G., Lee, J., and Hashimoto, H. (1997). Building topological maps by looking at people: An example of cooperation between intelligent spaces and robots. *Proceedings of the International Conference on Intelligent Robots and Systems (IROS 1997)*, 3:1326–1333.

Artač, M., Jogan, M., and Leonardis, A. (2002). Mobile robot localization using an incremental eigenspace model. *Proceedings of the International Conference on Robotics and Automation (ICRA 2002)*.

Bennewitz, M., Burgard, W., and Thrun, S. (2002). Using EM to learn motion behaviors of persons with mobile robots. *Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2002)*, 1:502–507.

Bennewitz, M., Burgard, W., and Thrun, S. (2003). Adapting navigation strategies using motions patterns of people. *Proceedings of the International Conference on Robotics and Automation (ICRA 2003)*, 2:2000–2005.

Freitas, R., Santos-Victor, J., Sarcinelli-Filho, M., and Bastos-Filho, T. (2003). Performance evaluation of incremental eigenspace models for mobile robot localization. In *Proc. IEEE 11th International Conference on Advanced Robotics (ICAR 2003)*, pages 417–422.

Fukui, R., Morishita, H., and Sato, T. (2003). Expression method of human locomotion records for path planning and control of human-symbiotic robot system based on spacial existence probability model of humans. *Proceedings of the International Conference on Robotics and Automation (ICRA 2003)*.

Gaspar, J., Winters, N., and Santos-Victor, J. (2000). Vision-based navigation and environmental representations with an omni-directional camera. *IEEE Transactions on Robotics and Automation*, 16(6):890–898.

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston.

Gracias, N. and Santos-Victor, J. (2000). Underwater video mosaics as visual navigation maps. *VisLab-TR 07/2000 - Computer Vision and Image Understanding*, 79(1):66–91.

Gutchess, D., Trajković, M., Cohen-Solal, E., Lyons, D., and Jain, A. K. (2001). A background model initialization algorithm for video surveillance. *International Conference on Computer Vision*, 1:733–740.

Hall, P., Marshall, D., and Martin, R. (1998). Incremental eigenanalysis for classification. *British Machine Vision Conference*, 14:286–295.

Kruse, E. and Wahl, F. (1998). Camera-based monitoring system for mobile robot guidance. *Proceedings of the International Conference on Intelligent Robots and Systems (IROS 1998)*, 2:1248–1253.

Murakami, H. and Kumar, B. (1982). Efficient calculation of primary images from a set of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(5):511–515.

Sloman, A. (1989). On designing a visual system (towards a gibsonian computation model of vision). *Journal of Experimental and Theoretical AI*, 1(4):289–337.

A SWITCHING ALGORITHM FOR TRACKING EXTENDED TARGETS

Andreas Kräußling, Frank E. Schneider, Dennis Wildermuth and Stephan Sehestedt
FGAN – Research Institute for Communications, Information Processing and Ergonomics
Neuenahrer Straße 20, 53343 Wachtberg–Werthhoven, Germany
a.kraeussling,frank.schneider,dennis,sehestedt@fgan.de

Keywords: Tracking, extended targets, Kalman filter, Viterbi algorithm, crossing targets.

Abstract: Tracking extended objects like humans in crowded environments is one of the challenges in mobile robotics. Several characteristics must be taken into consideration when evaluating the performance of such a tracking algorithm — e.g. accuracy, the need for computation time and the ability to deal with complex situations like crossing targets. In this paper two different algorithms for tracking extended targets are examined and compared by means of these criterias. One result is that none of the algorithms alone is a sufficient solution to the criterias. Therefore, a switching approach using both algorithms is introduced and tested on real world data.

1 INTRODUCTION AND RELATED WORK

For many real world applications it is essential that a robot is able to interact with its environment. This is true for multi–robot systems where a group of robots has to solve a given task or where robots are supposed to support people. For such situations, the awareness of the position of people and other robots is a fundamental ability for a mobile unit to be able to interact with its environment in an appropriate way.

This problem can be analysed under the superordinate concept of tracking. Tracking denotes the estimation of the position of an object based on consecutive sensor measurements. It is well studied in the field of aerial surveillance with radar devices (Bar-Shalom and Fortmann, 1988). In the area of mobile robots tracking is also a well established research topic (Prassler et al., 1999; Schulz et al., 2001; Fod et al., 2002; Fuerstenberg et al., 2002). In mobile robotics laser range scanners are one of the preferred sensor devices. A Sick laser range scanner for example can measure the distance to the next reflecting obstacle with a high angular resolution of e.g. 0.25 degree. Lasers have rapidly gained popularity for mobile robotic applications such as collision avoidance, navigation, localization and map building in the recent years (Thrun, 1998).

The problem of tracking people and other objects in densely populated environments with a robot–borne laser scanner can be characterized in the following way: most of the readings are from obstacles like walls or other objects and only a few measurements come from the tracked object itself. This fact is illustrated in Figure 1. It shows the measurements of one scan in a real system in our laboratory. In the scenario the observing robot, at which two Sick lasers with a 180 degree field of view each are mounted back to back, is located in the centre with coordinates $(0, 0)$. There are two humans in the field of view of the robot. Furthermore, there are two wall–like obstacles. Most of the measurements originate from the walls of the laboratory. The problem of allocation of data obtained from the presently accounted target is called the data association problem (Bar-Shalom and Fortmann, 1988). As a solution to this problem, a tracking algorithm might use a validation gate which separates the signals belonging to the current target from other signals. A second characteristic of tracking people with laser range scanners is the occurrence of several measurements from the same object. In contrast to common radar based tracking sensors the Sick laser scanner has a much higher resolution and refresh rate. This leads to the fact that the tracked object generates several measurements. Therefore, we have to deal with what we call extended objects instead of punctiform objects like in the common radar track-

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 117–128

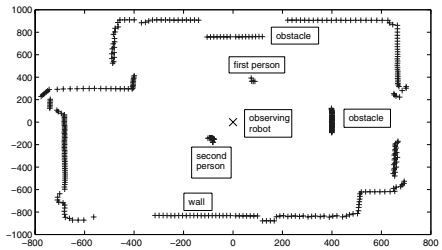


Figure 1: Measurements of one scan.

ing literature. Thereby, punctiform targets are those ones, which are the origin of just one measurement. A third characteristic of tracking in the field of mobile robotics is the occurrence of crossing targets. This means that two targets get very close to each other, so that they cannot be separated by common tracking algorithms (Fortmann et al., 1983), (Kräußling et al., 2004b). This situation can appear e.g. when two humans meet, talk to each other and split again and is a well known problem in mobile robotics (Prassler et al., 1999).

In Section 2 we introduce two algorithms which can deal with tracking extended objects as long as they are not crossing. The first algorithm just makes use of the Kalman filter (Kalman, 1960), whereas the second one also is based on the Viterbi algorithm (Viterbi, 1967). The application of the Kalman filter has a long tradition in mobile robotics (Dissanayake et al., 2001). Additionally, the underlying models for the dynamics and the observation process of the object are proposed and the details of the validation gate are given. In Section 3 the results of the comparison of the accuracy and the computational complexity of the algorithms are presented. The comparison of different algorithms is a well introduced issue in mobile robotics (Gutmann et al., 1998) and (Gutmann and Fox, 2002). In Section 4 the performance of the algorithms under the condition of crossing targets is studied and it is shown, that none of the algorithms can handle this situation sufficiently. Therefore an improved algorithm based on the Viterbi algorithm is introduced. In Section 5 a new hybrid or switching algorithm, which is the main contribution of this work, is proposed as a possible solution of the tracking problem in mobile robotics. It uses the improved algorithm only when a crossing occurs and otherwise it just uses a simple Kalman filter. The performance of the switching algorithm is tested on real data in detail. In Section 6 the switching algorithm is compared to the well established SJPDAF (Schulz et al., 2001). Finally, in Section 7 a summary and an outlook on future work are given.

2 THE MATHEMATICAL BACKGROUD OF THE ALGORITHMS

2.1 The Model

The dynamics of the object to be observed and the observation process itself are modeled by a hidden Gauß–Markov chain with the equations

$$x_k = Ax_{k-1} + w_{k-1} \tag{1}$$

and

$$z_k = Bx_k + v_k. \tag{2}$$

Thereby, x_k is the object state vector at time k, A is the state transition matrix, z_k is the observation vector at time k and B is the observation matrix. Furthermore, w_k and v_k are supposed to be uncorrelated zero mean white Gaussian noises with covariances Q and R.

Since the motion of a target in the plane has to be described a two dimensional kinematic model is used. Therefore, it is

$$x_k = \begin{pmatrix} x_{k1} & x_{k2} & \dot{x}_{k1} & \dot{x}_{k2} \end{pmatrix}^\top \tag{3}$$

with x_{k1} and x_{k2} the Cartesian coordinates of the target and $\dot{x}_{k1}$ and $\dot{x}_{k2}$ the corresponding velocities. z_k gives just the Cartesian coordinates of the target. For the coordinates the equation of a movement with constant velocity is holding, i.e. it is

$$x_{k+1,j} = x_{kj} + \Delta T \dot{x}_{kj}. \tag{4}$$

ΔT is the time interval between two consecutive measurements. For the progression of the velocities we use the equation

$$\dot{x}_{k+1,j} = e^{-\Delta T/\Theta}\dot{x}_{kj} + \Sigma\sqrt{1 - e^{-2\Delta T/\Theta}}u(k) \tag{5}$$

from (van Keuk, 1971) with the zero mean white Gaussian noise $u(k)$ with $E[u(m)u(n)^\top] = \delta_{mn}$. Thus, the velocity is supposed to decline exponentially. The term

$$\Sigma\sqrt{1 - e^{-2\Delta T/\Theta}}u(k) \tag{6}$$

models the process noise and the accelerations.

2.2 The Validation Gate

The validation gate is realised using the Kalman filter. The Kalman filter calculates a prediction $y(k+1|k)$ for the measurements $z_{k+1,l}$ from the actually handled target at time step $k+1$ via the formula

$$y(k+1|k) = B \cdot A \cdot x(k|k). \tag{7}$$

Thereby $x(k|k)$ is the estimate for the position of the target at time step k. For every sensor reading $z_{k+1,l}$

of the time step $k+1$ ($l = 1, \ldots 360$) the Mahalanobis distance λ (Mahalanobis, 1936) with

$$\lambda = (z_{k+1,l} - y(k+1|k))^\top \cdot [S(k+1)]^{-1} \cdot \cdot(z_{k+1,l} - y(k+1|k)) \quad (8)$$

is computed. Then all measurements with $\lambda > \lambda_{max}$ with a given treshold λ_{max} are excluded. See (Bar-Shalom and Fortmann, 1988) for further details. This procedure results in a set $\{\hat{z}_{k+1,i}\}_{i=1}^{m_{k+1}}$ of m_{k+1} selected measurements $\hat{z}_{k+1,i}$ The matrix $S(k+1)$ is the innovations covariance from the Kalman filter. In common filter applications this matrix is calculated from the predictions covariance $P(k+1|k)$ with the equation

$$S(k+1) = BP(k+1|k)B^\top + R \quad (9)$$

with the given covariance matrix R of the measurement noise. The predictions covariance is derived from the equation

$$P(k+1|k) = AP(k|k)A^\top + Q. \quad (10)$$

But for tracking extended objects this approach is not sufficient, since there is an additional influence of the extendedness of the object to the deviation of the measurements from the prediction $y(k+1|k)$. To take care of this feature an accessory positive definite matrix E should be added in Eq. (9). Because the lateral dimension of people usually shows a radius in the range of $30\,cm$, the entries of E should be in the range of 900. Thus, after some optimization process we used

$$E = \begin{pmatrix} 780 & 0 \\ 0 & 780 \end{pmatrix} \quad (11)$$

and

$$S(k+1) = BP(k+1|k)B^\top + R + E. \quad (12)$$

The values of the entries of the matrix E vastly exceed the values of the entries of the matrix R, so that the main contribution in Eq. (12) comes from the matrix E. Of course, a more elaborate model of the target shape like in (Taylor and Kleeman, 2004) or in (Zhao and Shibasaki, 2005) could be used. These authors have developed models for walking, modeling the movement of the two legs of a person explicitly. Thereby they make use of the fact that the laser scanners are usually mounted at the height of the legs. We have rejected such an approach because of the computational burden aligned with these approaches. Moreover looking at the actual data we get from the laser scanners we found that its hard to separate the legs of the persons in most of the scans. Finally, as one of the references has already mentioned, the situation can get very complex when there are crossing targets (Taylor and Kleeman, 2004). This can result in a dramatic increase of the number of hypothesis used for the modeling of the walking persons.

One characteristic of the model proposed in this paper consists of the fact, that the sequence $\{K_k\}_{k=1}^{\infty}$ of the Kalman gains (please note Eq. (15) for a definition) converges very rapidly to a limit. Thus, the calculations of the matrices K_k can be omitted and instead it is sufficient to calculate and use the limit $K = \lim_{k\to\infty} K_k$. This limit can be calculated quite easily, similar to the case of the α–β–filter described in (Ekstrand, 1983). These facts can be exploited for the development of a tracking algorithm for real time applications.

2.3 The Kalman Filter Algorithm with Equal Weights

This algorithm first calculates an unweighted mean z_{k+1} of the m_{k+1} measurements $\{\hat{z}_{k+1,l}\}_{l=1}^{m_{k+1}}$, that have been selected by the gate, i.e. it is

$$z_{k+1} = \frac{1}{m_{k+1}} \sum_{l=1}^{m_{k+1}} \hat{z}_{k+1,l}. \quad (13)$$

This mean is used as the input for the updating equation of the Kalman filter, i.e. it is

$$x(k+1|k+1) = x(k+1|k) + K_{k+1}(z_{k+1} - y(k+1|k)) \quad (14)$$

with the predictions $x(k+1|k) = Ax(k|k)$ and $y(k+1|k)$ and the Kalman gain K_{k+1} derived from the Kalman filter via the formula

$$K_{k+1} = P(k+1|k)B^\top [S(k+1)]^{-1} \quad (15)$$

or as supposed above by using the limit K of the sequence $\{K_k\}_{k=1}^{\infty}$. The covariances are then updated with the equation

$$P(k+1|k+1) = P(k+1|k) - K_{k+1}S(k+1)[K_{k+1}]^\top. \quad (16)$$

Finally, the estimates $x(k+1|k+1)$ are further improved by the use of the Kalman smoother (Shumway and Stoffer, 2000). The corresponding algorithm is called Kalman filter algorithm (KFA).

2.4 The Viterbi Based Algorithm

The Viterbi algorithm has been introduced in (Viterbi, 1967). A good description is also given in (Forney Jr., 1973). It has been recommended for tracking punctiform targets in clutter in (Quach and Farooq, 1994) and for tracking extended targets in (Kräußling et al., 2004a). It calculates for each selected measurement $\hat{z}_{k,i}$ a separate estimate $x(k|k)_i$. For the calculation of the estimates $x(k|k)_i$ in the update equation the measurement $\hat{z}_{k,i}$ and the predictions $x(k|k-1)_j$ and $y(k|k-1)_j$ from the predecessor j are used. When tracking punctiform targets in clutter, the predecessor is determined by minimizing the length of the paths

ending in $\hat{z}_{k,i}$. When regarding extended targets in most cases all measurements in the validation gate are from the target, so that it is not meaningful to consider the lengths of the paths ending in the possible predecessors $\hat{z}_{k-1,j}$ when determining the predecessor. Therefore, a better choice for the predecessor is that one for which the Mahalanobis distance (Mahalanobis, 1936)

$$\nu_{k,j,i}^{\top}[S_k]^{-1}\nu_{k,j,i} \tag{17}$$

is kept to a minimum. Thereby $\nu_{k,j,i}$ is the innovation

$$\nu_{k,j,i} = z_{k,i} - y(k|k-1)_j \tag{18}$$

and S_k is the innovations covariance. This procedure is similar to a nearest neighbour algorithm. As already mentioned above, there is a major difference to the tracking of punctiform targets, when calculating the matrix S_k for tracking extended targets. The extendedness of the targets can increase the distance of the measurements from the prediction $y(k|k-1)_j$. Thus we add an additional positive definite matrix when calculating the innovations covariance. Otherwise a lot of the measurements from the target would be excluded by the gating process.

When applying the Viterbi algorithm the application of the validation gate is performed in the following way. At first for every selected measurement $\hat{z}_{k-1,j}$ the gate is applied to the measurements at time k. That results in the sets $Z_{k,j}$ of measurements which have passed the particular gate for the measurement $\hat{z}_{k-1,j}$ successfully. The set of all measurements $\hat{z}_{k,i}$, that are associated with the target, is then just the union of these sets. The corresponding algorithms can deal with multimodal distributions to some extend, which is a major improvement when dealing with crossing targets.

The estimates delivered by the Viterbi algorithm are used as follows. One of these estimates is chosen as an estimate of the position of a target. This estimate is the one with index one. Again, different from tracking a punctiform target in clutter, it is not meaningful to make use of the lengths of the paths corresponding to the estimates. The corresponding algorithm is called Viterbi based algorithm (VBA) and has been introduced in (Kräußling et al., 2004a).

3 EVALUATION OF THE ALGORITHMS

3.1 Tracking a Circular Object

In mobile robotics there are mainly two classes of objects to track – other robots or people. Since a lot of service robots are of circular shape and these objects can be treated analytically we will concentrate on this class of objects in this section. We start with the following conjecture: the algorithm that uses a mean (i.e. KFA) estimates a point in the interior of the object, that is the mean of the points on the surface of the object, which are in the view of the observer. This mean, that we will call balance point, will be calculated as follows. To simplify the problem, it is assumed, that the centre of the observed circular object is at the origin of the planar coordinate system and the centre of the observer lies on the x–axis. The radius of the observed object is r and the distance from the centre of the observed object to the centre of the observer is denoted by d. The coordinates of the mean S in the interior are denoted by (x, y). Because of the symmetry of the problem it immediately follows that $y = 0$. Moreover,

$$x = \frac{1}{\phi}\int_0^{\phi} x(\theta)d\theta. \tag{19}$$

For the definition of the angle ϕ we refer to Figure 2 and for the definition of the distance $x(\theta)$ we refer to Figure 3. The latter is calculated from the known values d and r and the angle θ as follows. By the proposition of Pythagoras

$$r^2 = h^2 + x^2(\theta) \tag{20}$$

and

$$(d - x(\theta))^2 + h^2 = l^2. \tag{21}$$

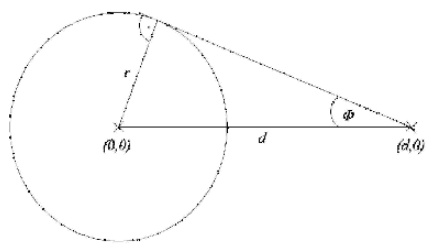


Figure 2: Derivation of the angle ϕ.

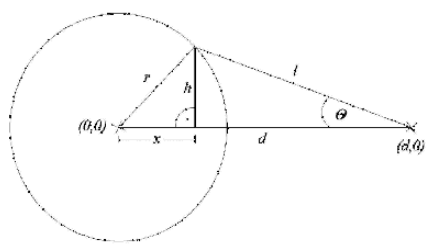


Figure 3: Derivation of the distance $x(\theta)$.

Furthermore

$$\sin\theta = \frac{h}{l}\,. \tag{22}$$

From these three equations the term

$$x(\theta) = d\sin^2\theta + \cos\theta\sqrt{r^2 - d^2\sin^2\theta} \tag{23}$$

for calculating $x(\theta)$ can be derived. Together with Eq. (19) this results in

$$x = \frac{1}{\phi}\int_0^{\phi}\left(d\sin^2\theta + \cos\theta\sqrt{r^2 - d^2\sin^2\theta}\right)\,d\theta. \tag{24}$$

According to Figure 2 there is the expression

$$\sin\phi = \frac{r}{d}\,, \tag{25}$$

which can be used for the derivation of the angle ϕ. For the antiderivative of the first term in the integral it is (Bronstein and Semendjajew, 1987)

$$\int \sin^2\theta d\theta = \frac{1}{2}\theta - \frac{1}{2}\sin\theta\sqrt{1-\sin^2\theta}. \tag{26}$$

The antiderivative of the second term can be found by integration by substitution. We use $u = d\sin\theta$ and therefrom it is

$$\int \cos\theta\sqrt{r^2 - d^2\sin^2\theta} = \frac{1}{d}\int\sqrt{r^2-u^2}du. \tag{27}$$

The antiderivative of $\int\sqrt{r^2-u^2}du$ is (Bronstein and Semendjajew, 1987)

$$\frac{1}{2}\left(u\sqrt{r^2-u^2} + r^2\arcsin\frac{u}{r}\right). \tag{28}$$

With the expression for $\sin\phi$ from Eq. (25) this finally results in

$$x = \frac{d}{2} + \frac{r}{2d\arcsin\frac{r}{d}}\left(\frac{\pi r}{2} - \sqrt{d^2-r^2}\right). \tag{29}$$

3.2 Experimental Results

Table 1 shows the results for different values of d used in our simulations. They are in the range of the typical distances between the laser and the object, which occur in the field of mobile robotics. For the radius r of the object we have set $r = 27cm$, which is in the range of the dimension of a typical mobile robot.

Table 1: Angle ϕ and Distance x.

d/cm	ϕ/rad	ϕ/deg	x/cm
100	0.2734	15.6647	23.3965
200	0.1354	7.7578	22.3607
400	0.0676	3.8732	21.7837
600	0.0450	2.5783	21.6103
800	0.0338	1.9366	21.4803

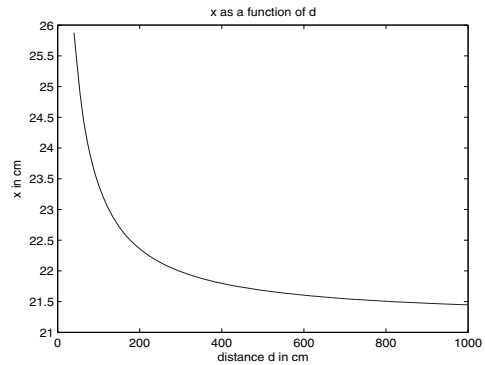


Figure 4: x as a function of the distance d.

Figure 4 shows the effect of the radius d on the balance point. With growing radius d the observable area of the object increases and hence the balance point S moves closer to the centre of the object.

In the following we consider the movement of the circular object around the laser range scanner on a circle with radius R. To evaluate the performance of the algorithms solving this problem simulated data has been used, because we needed to know the true position of the target very accurately. This is hard to achieve using data from a real experiment and has already been mentioned by other authors (Zhao and Shibasaki, 2005). Since we considered a movement on a circle, the process noise only originates from the centripetal force, that keeps the object on the circle. The simulations have been carried out for the values of the radius R introduced in Table 1. The values for the standard deviation σ of the measurement noise have been $0\,cm$, $1\,cm$, $3\,cm$, $5\,cm$, $7.5\,cm$ and $10\,cm$. These values are in the typical range of the errors of the commercial SICK lasers, which are commonly used in mobile robotics. For each pair of R and σ 20 replications have been carried out. For each time step k the Euclidian distance of the output of the tracking algorithm from the balance point has been calculated and therefrom the average of these distances over the whole run has been calculated. Finally, from these averages the average over the 20 cycles had been calculated. The results with unit $1\,cm$ are presented in Table 2.

Table 2: Average Distance from S for the KFA.

R/cm	100	200	400	600	800
$\sigma = 0\,cm$	0.97	0.37	0.22	0.56	0.25
$\sigma = 1\,cm$	0.97	0.37	0.22	0.56	0.26
$\sigma = 3\,cm$	0.97	0.38	0.30	0.57	0.35
$\sigma = 5\,cm$	0.99	0.43	0.41	0.65	0.54
$\sigma = 7.5\,cm$	1.05	0.56	0.57	0.79	0.78
$\sigma = 10\,cm$	1.11	0.63	0.74	0.96	1.01

The corresponding standard deviations calculated from the 20 cycles are small. They reach from about

$0.01\,cm$ for the smaller standard deviations of the measurement noise to about $0.1\,cm$ for the larger ones.

It is apparent, that the outputs of the KFA produce a good estimate for the balance point S. Thus, an estimate for the centre of the circular object can be derived directly. This is due to the fact, that the Euclidian distance x of the balance point to the centre of the object can be calculated depending on R as above. Furthermore, the observer, the balance point S and the centre C of the object are lying on a straight line as indicated in Figure 5. Thus, KFA delivers good information about the position of the object.

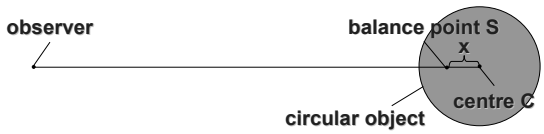


Figure 5: Determination of the centre C.

The VBA algorithm calculates one position estimate for every single range reading that originates from one target. So each estimate corresponds to one point on the surface of this object. The algorithm then chooses one of these estimates without having the knowledge to which point the estimation is corresponding. Therefore, we have a great uncertainty about the estimated position.

This is illustrated in Figures 6–10. In Figures 6 and 7 the points on the surface of the object, which are in the view of the observer, are reproduced. The figures show that for a greater distance of the object to the observer there is a larger amount of points in the view of the observer. Hence, for a greater distance there is a greater uncertainty about which point on the surface is estimated. The angle φ_G indicated in the figures is the same as the angle ϕ introduced in Figure 2.

Figures 8 and 9 show the possible positions of the centre of the object. Again, there is a greater uncertainty for a greater distance.

Finally, Figure 10 presents some examples for possible positions of the object. From Figures 8 and 9 the following statement can be concluded: there is a great uncertainty in the estimate of the centre of the object when applying VBA. For large distances this uncertainty is in the range of the diameter of the object. There are two further problems that complicate the situation:

- first, the points on the surface that are hit by the laser beam, change from time step to time step.
- second, there is an additional error in form of the measurement noise, that corrupts the data.

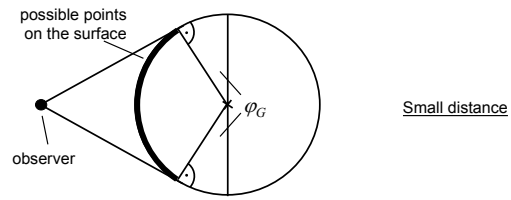


Figure 6: Possible points, small distance.

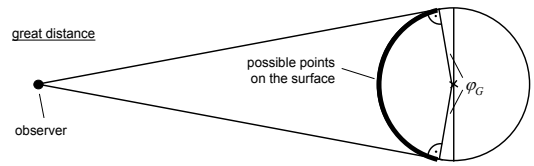


Figure 7: Possible points, large distance.

Recapitulating it can be concluded that the VBA delivers only sparse information about the true position of the target.

Now it is referred to a second criterion for the comparison of the algorithms, the computational complexity. As a measure for this property the need of computing time for the fulfilment of the calculations for one time step is used. The reason for this procedure is, that the VBA is very complex and therefore it would be very difficult to estimate for instance the number of matrix multiplications. The algorithms have been implemented in MATLAB and have been conducted on a Pentium IV with 2.8 GHz. The KFA needed about $20\,ms$ per time step for all combinations of radius R and standard deviation σ of the measurement noise. The results for the VBA are given in Table 3 in seconds. It shows that the computation time varies from about $80\,ms$ to about $1.5\,s$ for the VBA. Thus, the values of the VBA depend on the radius R. This is due to the fact that the number of measurements from the target highly depends on the radius. The complexity of the VBA strongly depends on the number of measurements from the object. For example the predecessor, that has to be determined for every new measurement, has to be chosen among all the measurements from the last time step.

Table 3: Computation time for the VBA.

R/cm	100	200	400	600	800
$\sigma = 0\,cm$	1.45	0.41	0.17	0.10	0.08
$\sigma = 1\,cm$	1.46	0.41	0.17	0.10	0.08
$\sigma = 3\,cm$	1.46	0.42	0.17	0.10	0.08
$\sigma = 5\,cm$	1.48	0.41	0.16	0.10	0.08
$\sigma = 7.5\,cm$	1.47	0.41	0.17	0.10	0.08
$\sigma = 10\,cm$	1.48	0.42	0.17	0.10	0.08

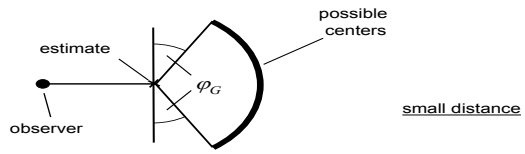


Figure 8: Possible centers, small distance.

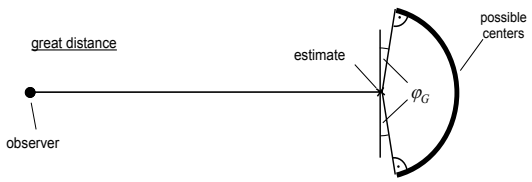


Figure 9: Possible centers, large distance.

4 THE PROBLEM OF CROSSING TARGETS

Two targets are crossing, if their validation gates intersect, i.e. some measurements are lying in the validation gates of both of the two targets. Figure 11 shows a typical situation.

Figures 12 and 13 show the behaviour of the two introduced algorithms when being applied to the problem of crossing targets using real data originating from an experiment with two walking persons in our laboratory. They show the estimates for the position of the objects calculated by the two algorithms by use of ellipses. Thereby the estimated position is the centre of the ellipse, whereas the shape of the ellipse represents the actual geometry of the tracked object. The objects start in the left and move to the right as indicated in Figure 11.

Obviously none of the algorithms can deal with the problem of crossing targets. They all locate both objects at the same position after the crossing. Obviously the targets are not separated after the crossing. Thus, tracking extended crossing objects is a difficult situation for a tracking algorithm. There are several methods for tracking punctiform crossing targets in clutter:

1. the MHT (Multi Hypothesis Tracker) introduced by Reid in 1979 (Reid, 1979).
2. the JPDAF (Joint Probabilistic Data Association Filter) introduced by Fortmann, Bar–Shalom and Scheffe in 1983 (Fortmann et al., 1983).
3. the PMHT (Probabilistic Multi Hypothesis Tracker) introduced by Streit and Luginbuhl in 1994 (Streit and Luginbuhl, 1994).

Of course, an extension of this algorithms to tracking extended crossing objects is straightforward. But

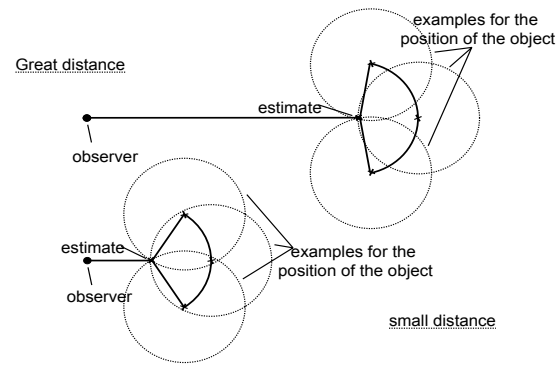


Figure 10: Examples for the position of the object.

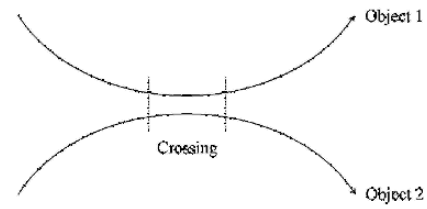


Figure 11: Two crossing objects.

there are several reasons, why such approaches might fail:

- in most cases there are several measurements from the same target
- the crossing can last for a longer time period
- one of the objects might be occluded by the other object for some time
- the objects can accomplish very abrupt manoeuvres during the crossing or especially at the end of the crossing.

As an example we give the results for an EM based method (Stannus et al., 2004), which is an extension of the PMHT (Streit and Luginbuhl, 1994) to extended targets. Figure 14 shows the results when applying this algorithm to real data. The circumstances for an extension of the JPDAF are illustrated in Section 6. Moreover the computational burden for applying these algorithms is very high when applied to extended targets, since these objects can be the origin of up to ten measurements. Thus, we have developed an improved algorithm based on the VBA that can deal with the problem of crossing targets (Kräußling et al., 2004b). It uses the fact, that the VBA is able to cope

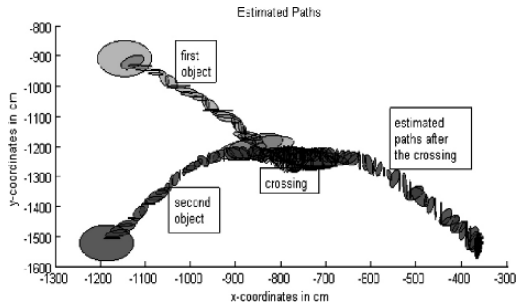


Figure 12: Application of the KFA to crossing targets, real data.

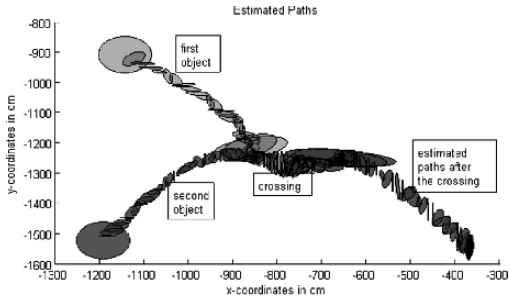


Figure 13: Application of the VBA to crossing targets, real data.

with multimodal densities to some degree. This feature is due to the fact, that the VBA calculates separate validation gates and state respectively position estimates for every selected measurement. The handling of multimodal densities is a characteristic that the VBA algorithm has in common with the SJPDAF algorithm (Schulz et al., 2001). While the SJPDAF algorithm uses particle filtering (Gordon et al., 1993), (Pitt and Shephard, 1997) and thus has to deal with several hundreds of particles, the Viterbi algorithm only handles a few points or state estimates. Additionally, these points contain some information about the geometry of the tracked object as proposed in

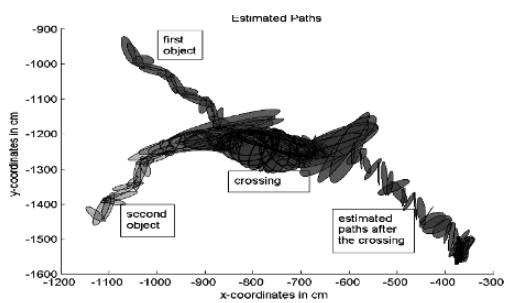


Figure 14: Crossing of two objects, real data, EM based method.

(Kräußling et al., 2004a). When a crossing between two targets occurs the VBA shows the following behaviour: as soon as the crossing takes place the algorithm tracks all points originated from both objects simultanously. When the crossing is over, these points are again separated into two distinct clusters of points and these clusters are still tracked simultaneously for both objects. Only the assignment of the clusters to the objects is wrong, since in most cases one cluster is associated to both objects by the VBA algorithm at the end of the tracking process like in Figure 13. Our new approach is based on these observations. It uses the results of the VBA and furthermore performs the following three distinct steps:

1. At every time step and for each pair of objects it is examined, wether a crossing has occurred. This is supposed to be the case, if at least one measurement is associated with both of the targets by the gating process.
2. For each pair of targets for those a crossing has been detected it is examined whether the crossing has finished. This is done by testing if the estimates delivered by the VBA have dispersed into two distinct clusters with a minimum Euclidian distance. For people tracking this is the case when there are at least two measurements with a minimal Euclidian distance of 300 cm. This is due to the fact, that the maximal distance of the points on the two different legs of a person in human walking can be assumed to not exceed 150 cm. In our experiments the value of 300 cm worked well also with regard to the problem that the gates of the two targets should not intersect again once the end of the crossing has been detected.
3. As soon as the end of the crossing has been observed the two corresponding clusters of estimates have to be separated and assigned to the two objects. In order to do so an arbitrarily chosen point of the combined cluster is assigned to the object with the lower index. Then, for every other point it is determined wether the Euclidian distance to the first point is larger or lower than 150 cm. In the first case it is assigned to the object with the higher index and otherwise to that with the lower one. Of course, since the first point is arbitrarily chosen the two objects might be interchanged after the crossing by this procedure. But in our opinion there is no general solution to this problem which is based only on laser distance information.

Thus the three steps are carried out based on geometrical considerations and can be viewed under the superordinate concept of data mining (Han and Kamber, 2001). Finally, like for the VBA at the end of the tracking process for each object the path with index one is selected and a Kalman smoother is applied. This improved algorithm is called Cluster Sorting Al-

gorithm (CSA). For further details see (Kräußling et al., 2004b). Figure 15 shows the application of the CSA to the previously used data.

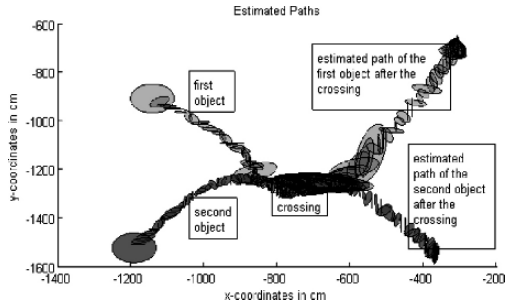


Figure 15: Crossing targets, handled by the CSA, real data.

5 A NEW SWITCHING ALGORITHM

Since the CSA is able to deal with crossing targets it could, of course, be used for the whole tracking process. But since this algorithm is based on the VBA algorithm it is not as accurate as the KFA as long as no crossing takes place and needs much more computation time. Therefore, we developed a new switching or hybrid algorithm (SA), which uses the CSA only when a crossing takes place. For the rest of the time it uses the very accurate and fast KFA. This choice is also motivated by the fact, that the KFA is faster and more accurate than other algorithms developed by our research group for tracking extended objects (Kräußling et al., 2005). Thereby, crossings are detected as in the case of the CSA. Figure 16 shows the flowchart of the SA. Figure 17 shows that the SA can

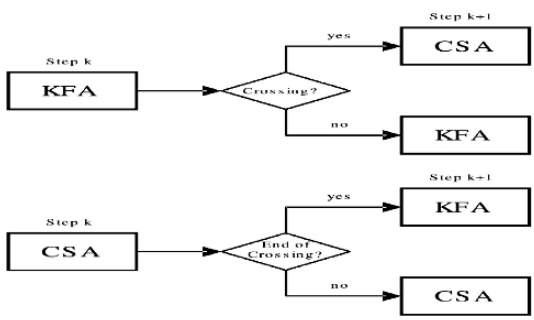


Figure 16: Flowchart of the switching algorithm.

deal with crossing targets as well as the CSA using the data from the last section.

To illustrate the power of the SA further experiments with real data have been carried out. There, two persons were walking around in our laboratory. The measurements were recorded with two SICK lasers,

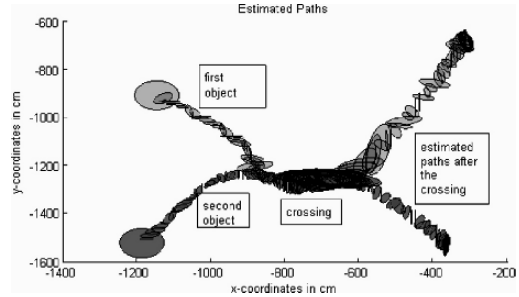


Figure 17: Crossing targets, handled by the new switching algorithm, real data.

each of them with a 180 degree field of view, mounted on a mobile robot. The evaluation of the algorithms was performed by means of five similar scenarios. In each scenario two persons walked separated for some time interval t_1 at the beginning of the experiment. Then the persons met each other and walked together for some time interval t_2, so that a crossing took place. Finally, the persons split again and walked alone for the time interval t_3. Thereby, the time interval t_2 was arranged to be approximately 30 seconds for each scenario. Furthermore, the time intervals t_1 and t_3 were of same length for each run varying from 30 seconds to 150 seconds. Figure 18 shows an example of the results for the estimated paths using the SA. Like in Section 4 KFA and VBA always failed, whereas CSA and SA behaved well for all five scenarios.

Table 4 describes the results for the needed computing time. It contains the average time t_a needed for the calculations of one time step in milliseconds. The table shows the improvement that can be achieved using the SA in comparison to the CSA. Moreover, with growing intervals t_1 and t_3 the gain increases rapidly.

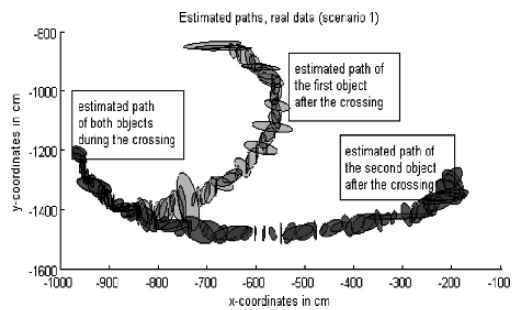


Figure 18: Crossing targets, real data of scenario 1, handled by the new SA.

Table 4: Average Computing Time.

scenario	1	2	3	4	5
KFA	54.1	53.8	54.6	53.9	54.4
VBA	379.0	355.7	478.6	366.3	469.3
CSA	294.3	258.6	329.5	260.7	320.3
SA	171.3	123.7	119.2	110.8	91.5

6 THE SWITCHING ALGORITHM COMPARED TO THE SJPDAF

Tracking multiple moving objects generally requires to keep track of the joint probability distribution of all objects. However, this is intractable in practice already for a small number of objects. Therefore, one commonly used approach is to track the individual objects independently, using factorial representations of the states. Then one has to associate the measurments to the corresponding object. For this, the Joint Probabilistic Association Filter can be used. In the following we briefly describe a sample based approach, known as SJPDAF (Schulz et al., 2001).

Let $X(t) = \{x_1^t, ..., x_K^t\}$ be the state of the K tracked objects at time t. Note that each x_i^t is a random variable over the state space of a single target. Furthermore, $Z(t) = \{z_1^t, ..., z_{m_t}^t\}$ denotes a measurement at time t, with z_i^t being one feature of such a measurement. To assign the observed features to a particular target we assume punctiform objects. Moreover, sometimes targets are not detected by the sensor and false alarms can occur.

In the JPDAF framework, a joint association event θ is a set of pairs $(j, i) \in \{0, ..., m_t\} \times \{1, ..., K\}$.. Each θ determines which feature is assigned to which object. With Θ_{ji} being the set of all valid association events which assign feature j to object i, JPDAF computes the posterior probability that feature j is caused by the target i by

$$\beta_{ji} = \sum_{\theta \in \Theta_{ji}} P(\theta \mid Z(t)) \tag{30}$$

SJPDAF uses a sample-based representation of the belief $p(x_i^t)$. To represent the density $p(x_i^t \mid Z(t))$ a set of N weighted samples is used. Such a sample set is a discrete approximation of a probability distribution. Each sample is a pair $\{x_{i,n}^t, \omega_{i,n}^t\}$ consisting of a state $x_{i,n}^t$ and the corresponding importance factor $\omega_{i,n}^t$. The prediction is done by drawing samples from the prior belief and by updating their state according to the prediction model $p(x_i^t \mid p(x_i^{t-1}, \delta t)$. To correct the prediction a feature set $Z(t)$ is applied. There, we have to take the assignment probabilities β_{ji} into account. The weights of the samples are computed by

$$\omega_{i,n}^t = \alpha \sum_{j=0}^{m_t} \beta_{ji} p(z_j^t \mid x_{i,n}^t) \tag{31}$$

α is a normalizer, ensuring that the sum of all weights is 1. Finally, we obtain N new samples by bootstrap resampling.

We now apply the SJPDAF to the exemplary problem introduced in Section 3. Since the SJPDAF uses an unweighted mean of the measurements like KFA, one might conjecture that the output of this algorithm gives an estimate for the balance point S like KFA and SA. As Table 5 shows there is a good match between the balance point and the results of the SJPDAF. Thereby, SA performs slightly better with regard to Table 2. The computation time needed by the SJPDAF for one time step is about $89\,ms$ for all combinations of standard deviations and distance of the observer to the object. Since the SA needs only about $20\,ms$ per time step it is superior to SJPDAF with respect to this criterion. Of course, the results for the SJPDAF depend on the number of particles that are used. In the experiments we used 400 particles. As our research has shown, a too vast decrease of this number might result in a loss of the target.

Table 5: Average Distance from S for the SJPDAF.

R/cm	100	200	400	600	800
$\sigma = 1\,cm$	0.53	0.73	1.45	2.24	3.10
$\sigma = 3\,cm$	0.83	0.77	1.49	1.73	2.04
$\sigma = 5\,cm$	1.27	1.08	1.62	1.84	2.21
$\sigma = 7.5\,cm$	1.82	1.48	1.89	2.12	2.50
$\sigma = 10\,cm$	2.44	1.92	2.23	2.49	2.89

We now examine the performance of the SJPDAF when applied to crossing targets. Thereby, the circumstances are more complicated than for the SA. Since SJPDAF uses random numbers, the results can differ from run to run. Therefore, for each scenario 20 cycles have been conducted. The results are given in the following (Table 6). Thereby, the percentage of the runs, for those the targets are tracked separately after the crossing is given.

Table 6: Handling of Crossing Targets, SJPDAF.

scenario	1	2	3	4	5
percentage	80	90	15	—	0

In the fourth scenario there was an interference with a static object nearby the wall. For the 5th scenario only ten runs have been conducted. We observe that SJPDAF performs well in scenario 1 and 2, and performs poorly in scenario 3. The fifth setting is not solved by this method. Thus, in all scenarios our switching algorithm outperforms the SJPDAF. Recapitulating it can be said, that the new switching algorithm is a good improvement to the SJPDAF, which is the state of the art in mobile robotics up to now.

7 CONCLUSIONS

In this paper we have addressed the problem of tracking extended targets. Two basic algorithms for the tracking process have been introduced: they are either just using the Kalman filter (KFA) or additionally the Viterbi algorithm (VBA). The comparison of the algorithms has shown that the KFA is much faster and gives much more information about the position of the object than the VBA. Thereafter, the problem of two crossing targets has been introduced. It has been shown that both algorithms produce insufficient results under the constraints of crossing targets. Thus, an enhancement of the VBA in form of the CSA has been proposed, which can deal with crossing targets. Since the CSA is based on the VBA and therefore is imprecise and slow, we finally developed the SA, which makes use of the CSA only when a crossing has been detected and otherwise uses the KFA. The performance of the SA has been demonstrated on real data. Thereby, it has been shown, that the SA can handle crossing targets as well as the CSA but needs much less computing time. Finally, the switching algorithm has been compared to the well established SJPDAF. There, it has turned out that the SA is superior with respect to the three examined criterions – accuracy of the position estimation, computational complexity and handling of crossing targets.

We showed that the problem of target tracking is sufficiently solved by the SA. Further research aims at an extension of the SA from the crossing of two objects to the crossing of an arbitrary number of objects (multi target tracking). Most recent investigations show, that such an extension is feasible. Furthermore, efforts should be undertaken to make the SA usable for real time applications. One approach could be a preselection of the data, since most of the computational burden when applying the CSA arises from the gating process. This preselection can be performed using geometrical features of the problem. One feature that objects in mobile robotics have in common is that they are located in some distance in front of a wall. When investigating the measurements from the laser scan there should be two jumps in the scan line of the observing robot when there is an object in front of a wall. One jump results in an edge where the distance decreases, while the other jump leads to an edge where the distance increases. Figure 19 shows a typical situation. The second feature comes from the fact, that the number of measurements generated by the object is very limited. The limit depends on the extension which is e.g. about $50\,cm$ for robots, and the distance to the object. Using the combination of these two features in most cases nearly all measurements originating from the walls can be eliminated. These measurements are by far the major fraction of the 360 measurements from the scan. Furthermore,

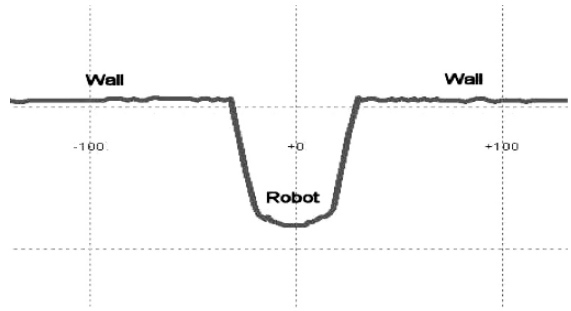


Figure 19: Measurements from an object in front of a wall.

the calculation of the Kalman gains should be substituted by the calculation of the limit as mentioned above. First preliminary results show, that by these modifications the SA needs only about $30\,ms$ for the computations of one time step on a Pentium IV with 2.8 GHz during a crossing, i.e. when applying the slow CSA. Since our SICK lasers have a frequency of $6\,Hz$ this means, that the SA is capable for real time applications after these modifications.

*A*ppendix

The model we use in this paper is inspired by (van Keuk, 1971). Since (van Keuk, 1971) is not available electronically, we give the matrices A, B, Q and R of this model in the following. It is

$$A = \begin{pmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & e^{-\Delta T/\Theta} & 0 \\ 0 & 0 & 0 & e^{-\Delta T/\Theta} \end{pmatrix}, \tag{32}$$

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \tag{33}$$

$$Q = \Sigma^2 \left(1 - e^{-2\Delta T/\Theta}\right) \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{34}$$

and

$$R = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \tag{35}$$

with $r_{11} = \sigma_r^2 \cos^2\phi_k + r_k^2\sigma_\phi^2 \sin^2\phi_k$, $r_{12} = r_{21} = (\sigma_r^2 - r_k^2\sigma_\phi^2)\sin\phi_k\cos\phi_k$ and $r_{22} = \sigma_r^2 \sin^2\phi_k + r_k^2\sigma_\phi^2\cos^2\phi_k$. Thereby r_k and ϕ_k are the polar coordinates of the actual position of the target. σ_r is the standard deviation of the error of the measurement of the distance from the observer to the object and σ_ϕ is the standard deviation of the error of the measurement of the angle corresponding to the position of the object in polar coordinates. Furthermore, for the parameters Θ and Σ the values $\Theta = 20$ and $\Sigma = 60$ are good choices.

REFERENCES

Bar-Shalom, Y. and Fortmann, T. (1988). *Tracking and Data Association*. Academic Press.

Bronstein, I. N. and Semendjajew, K. A. (1987). *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Thun und Frankfurt/Main.

Dissanayake, G. M. W. M., Newman, P., Clark, S., Durrant-Whyte, H., and Csorba, M. (2001). A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17:229–241.

Ekstrand, B. (1983). Analytical steady state solution for a kalman tracking filter. *IEEE Trans. Aerospace and Electronic Systems*, AES–19:815–819.

Fod, A., Howard, A., and Mataric, M. J. (2002). Laser–based people tracking. In *Proceedings of the IEEE Intl. Conf. on Robotics and Automation*, pages 3024–3029.

Forney Jr., G.-D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Fortmann, T. E., Bar-Shalom, Y., and Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, OE–8(3).

Fuerstenberg, K. C., Linzmeier, D. T., and Dietmayer, K. C. J. (2002). Pedestrian recognition and tracking of vehicles using a vehicle based multilayer laserscanner. In *Proceedings of IV 2002, Intelligent Vehicles Symposium*, volume 1, pages 31–35.

Gordon, N., Salmond, D., and Smith, A. (1993). A novel approach to nonlinear/non–gaussian bayesian state estimation. *IEE Proceedings F*, 140:107–113.

Gutmann, J.-S., Burgard, W., Fox, D., and Konolige, K. (1998). An experimental comparison of localization methods. In *International Conference on Intelligent Robots and Systems (IROS 1998), Victoria, Canada*.

Gutmann, J.-S. and Fox, D. (2002). An experimental comparison of localisation methods continued. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002), Lausanne, Switzerland*.

Han, J. and Kamber, M. (2001). *Data Mining*. Academic Press, London.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Trans. ASME, J. Basic Engineering*, 82:34–45.

Kräußling, A., Schneider, F. E., and Wildermuth, D. (2004a). Tracking expanded objects using the viterbi algorithm. In *Proceedings of the IEEE Conference on Intelligent systems, Varna, Bulgaria*.

Kräußling, A., Schneider, F. E., and Wildermuth, D. (2004b). Tracking of extended crossing objects using the viterbi algorithm. In *Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.

Kräußling, A., Schneider, F. E., and Wildermuth, D. (2005). Zur verfolgung ausgedehnter ziele — eine übersicht über ausgewählte algorithmen und ein vergleich deren güte. Technical report, FKIE/FGAN, Wachtberg, Germany.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Science*, 12:49–55.

Pitt, M. and Shephard, N. (1997). Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*.

Prassler, E., Scholz, J., and Elfes, E. (1999). Tracking people in a railway station during rush–hour. In Christensen, H. I., editor, *Computer Vision Systems*, volume 1542, pages 162–179. Springer, lecture notes in computer science edition.

Quach, T. and Farooq, M. (1994). Maximum likelihood track formation with the viterbi algorithm. In *Proceedings of the 33rd Conference on Decision and Control, Lake Buena Vista, Florida*.

Reid, D. B. (1979). An algorithm for tracking multiple targets. *IEEE Trans. Automatic Control*, AC–24:843–854.

Schulz, D., Burgard, W., Fox, D., and Cremers, A. B. (2001). Tracking multiple moving objects with a mobile robot. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*.

Shumway, R. H. and Stoffer, D. S. (2000). *Time Series Analysis and Its Applications*. Springer.

Stannus, W., Koch, W., and Kräußling, A. (2004). On robot–borne extended object tracking using the em algorithm. In *Proceedings of the 5th Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal*.

Streit, R. L. and Luginbuhl, T. E. (1994). Maximum likelihood method for multi–hypothesis tracking. *Signal and Data Processing of Small Targets, SPIE*, 2335.

Taylor, G. and Kleeman, L. (2004). A multiple hypothesis walking person tracker with switched dynamic model. In *Proceedings of the Australasian Conference on Robotics and Automation, Canberra, Australia*.

Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71.

van Keuk, G. (1971). Zielverfolgung nach kalman–anwendung auf elektronisches radar. Technical Report 173, Forschungsinstitut für Funk und Mathematik, Wachtberg–Werthhoven, Germany.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions On Information Theory*, IT–13(2).

Zhao, H. and Shibasaki, R. (2005). A novel system for tracking pedestrians using multiple single–row laser–range scanners. *IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans*, 35(2):283–291.

SFM FOR PLANAR SCENES: A DIRECT AND ROBUST APPROACH*

Fadi Dornaika and Angel D. Sappa
Computer Vision Center
Edifici O Campus UAB
08193 Bellaterra, Barcelona, Spain
{*dornaika,sappa*}*@cvc.uab.es*

Keywords: Structure From Motion, motion field, image derivatives, robust statistics, non-linear optimization.

Abstract: Traditionally, the Structure From Motion (SFM) problem has been solved using feature correspondences. This approach requires reliably detected and tracked features between images taken from widespread locations. In this paper, we present a new paradigm to the SFM problem for planar scenes. The novelty of the paradigm lies in the fact that instead of image feature correspondences, only image derivatives are used. We introduce two approaches. The first approach estimates the SFM parameters in two steps. The second approach directly estimates the parameters in one single step. Moreover, for both strategies we introduce the use of robust statistics in order to get robust solutions in presence of outliers. Experiments on both synthetic and real image sequences demonstrated the effectiveness of the developed methods.

1 INTRODUCTION

Computing object and camera motions from 2D image sequences has been a central problem in computer vision for many years (Adams et al., 2002; Qian and Chellapa, 2004; Xiang and Cheong, 2003; Zelnick-Manor and Irani, 2000). More especially, computing the camera motion and/or its 3D velocity is of particular interest to a wide variety of applications in computer vision and robotics such as calibration, visual servoing, etc. Many algorithms have been proposed for estimating the 3D relative camera motions (discrete case) (Jonathan and Sclaroff, 2002; Weng et al., 1993; Zucchelli et al., 2002) and the 3D velocity (differential case) (Brooks et al., 1997; Srinivasan, 2000). The discrete case requires feature matching/tracking, and the differential case the computation of the optical flow field (2D velocity). These tasks are generally ill-conditioned due to significant local appearance changes and/or large disparities. Most of the SFM algorithms are general in the sense that they assume no prior knowledge about the scene. In many practical cases, planar or quasi-planar structures occur frequently in real images. In this paper, we introduce a novel paradigm to deal with the SFM problem of planar scenes using image derivatives only. This paradigm has the following advantages. First, we need not to extract features nor to track them in several images. Second, robust statistics are invoked in order to get stable estimates. We introduce two approaches. The first approach estimates the SFM parameters in two steps. The second approach directly estimates the parameters in one single step. Using image derivatives has been exploited in (Brodsky and Fermuller, 2002) to make camera intrinsic calibration. In our study, we deal with the 3D motion of the camera as well as with the plane structure. The paper is organized as follows. Section 2 states the problem. Section 3 describes a two-step approach. Section 4 describes a one-step approach. Section 5 shows how image derivatives are computed. Experimental results on both synthetic and real image sequences are given in Section 6.

*This work was supported by the Government of Spain under The Ramón y Cajal Program.

2 BACKGROUND

Throughout this paper we represent the coordinates of a point in the image plane by small letters (x, y) and the object coordinates in the camera coordinate frame by capital letters (X, Y, Z). In our work we use the perspective camera model as our projection model. Thus, the projection is governed by the fol-

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 129–135

lowing equation were the coordinates are expressed in homogeneous form,

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & s & x_c & 0 \\ 0 & r\,f & y_c & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1)$$

Here, f denotes the focal length in pixels, γ and s the aspect ratio and the skew and (x_0, y_0) the principal point. These are called the intrinsic parameters. In this study, we assume that the camera is calibrated, i.e., the intrinsic parameters are known. For the sake of presentation simplicity, we assume that the image coordinates have been corrected for the principal point and the aspect ratio. This means that the camera equation can be written as in (1) with $\gamma = 1$, and $(x_0, y_0) = (0, 0)$. Also, we assume that the skew is zero (s = 0). With these parameters the projection simply becomes

$$x = f\,\frac{X}{Z} \quad \text{and} \quad y = f\,\frac{Y}{Z} \quad (2)$$

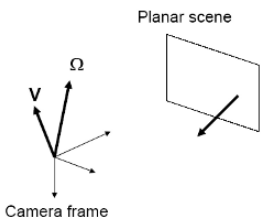


Figure 1: The goal is to compute the camera 3D velocity as well as the plane structure from the image derivatives.

Let $I(x, y, t)$ be the intensity at pixel (x, y) in the image plane at time t. Let $u(x, y)$ and $v(x, y)$ denote components of the motion field in the x and y directions respectively. This motion field is caused by the translational and rotational camera velocities $(\mathbf{V}, \Omega) = (V_x, V_y, V_z, \Omega_x, \Omega_y, \Omega_z)$ Figure 1. Using the constraint that the gray-level intensity is locally invariant to the viewing angle and distance we obtain the well-known optical flow constraint equation:

$$I_x\,u + I_y\,v + I_t = 0 \quad (3)$$

where $u = \frac{\partial x}{\partial t}$ and $v = \frac{\partial y}{\partial t}$ denote the motion field. The spatial derivatives $I_x = \frac{\partial I}{\partial x}$ and $I_y = \frac{\partial I}{\partial y}$ (the image gradient components) can be computed by convolution with derivatives of a 2D Gaussian kernel. They can be computed from one single image - the current image. The temporal derivative $I_t = \frac{\partial I}{\partial t}$ can be computed by convolution between the derivative of a 1D Gaussian and the image sequence (see Section 5).

The perspective camera observes a planar scene[2] described in the camera coordinate system by $Z = A\,X + B\,Y + C$.

One can show that the equations of the motion field as a function of the 3D velocity $(\mathbf{V}, \Omega)$ are given by these two equations:

$$u(x, y) = a_1 + a_2\,x + a_3\,y + a_7\,x^2 + a_8\,xy \quad (4)$$
$$v(x, y) = a_4 + a_5\,x + a_6\,y + a_7\,xy + a_8\,y^2 \quad (5)$$

where the coefficients are depending on the SFM parameters:

$$\begin{aligned} a_1 &= -f\,(\tfrac{V_x}{C} + \Omega_y) \\ a_2 &= (\tfrac{V_x}{C}\,A + \tfrac{V_z}{C}) \\ a_3 &= \tfrac{V_x}{C}\,B + \Omega_z \\ a_4 &= -f\,(\tfrac{V_y}{C} - \Omega_x) \\ a_5 &= (\tfrac{V_y}{C}\,A - \Omega_z) \\ a_6 &= (\tfrac{V_y}{C}\,B + \tfrac{V_z}{C}) \\ a_7 &= \tfrac{-1}{f}(\tfrac{V_z}{C}\,A + \Omega_y) \\ a_8 &= \tfrac{-1}{f}(\tfrac{V_z}{C}\,B - \Omega_x) \end{aligned} \quad (6)$$

One can notice that the two solutions (V_x, V_y, V_z, C) and $\lambda\,(V_x, V_y, V_z, C)$ yield the same motion field. This is consistent with the scale ambiguity that occurs in the general SFM problem.

Our goal is to estimate the instantaneous camera velocity $(\mathbf{V}, \Omega)$ as well as the plane orientation from the image derivatives. The translational velocity can be recovered up to a scale. It should be noticed that for continuous videos the camera motion has to be computed for all time instants during which the camera is moving.

3 A TWO-STEP APPROACH

In this section, we propose a two-step approach. In the first step, the 8 coefficients $(a_1, \ldots, a_8)$ are recovered by solving an over-constrained system derived from (3) using robust statistics. In the second step, the translational and rotational velocities as well as the plane orientation are recovered from Eq.(6) using some non-linear technique.

3.1 Robust Estimation of the 8 Coefficients

We assume that the image contains N pixels for which the spatio-temporal derivatives (I_x, I_y, I_t) have been computed. In practice, N is very large. In order to reduce this number, one can either drop

[2] Our work also addresses the case where the scene contains a dominant planar structure.

pixels having small gradient components or adopt a low-resolution representation of the images. In the sequel, we do not distinguish between the two cases, i.e., N is either the original size or the reduced one.

By inserting Eqs.(4) and (5) into Eq.(3) we get

$$I_x\, a_1 + I_x x\, a_2 + I_x y\, a_3 + I_y\, a_4 + I_y\, x\, a_5 + I_y\, y\, a_6 + (I_x x^2 + I_y\, x\, y)\, a_7 + (I_x\, x\, y + I_y\, y^2)\, a_8 = -I_t \quad (7)$$

By concatenating the above equation for all pixels, we get an over-constrained linear system having the following form:

$$\mathbf{G}\,\mathbf{a} = \mathbf{e} \quad (8)$$

where $\mathbf{a}$ denotes the column vector $(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8)^T$.

It is well known that the Maximum Likelihood solution to the above linear system is given by:

$$\mathbf{a} = \mathbf{G}^{\dagger}\,\mathbf{e} \quad (9)$$

where $\mathbf{G}^{\dagger} = (\mathbf{G}^T\,\mathbf{G})^{-1}\,\mathbf{G}^T$ is the pseudo-inverse of the $N \times 8$ matrix $\mathbf{G}$. This solution is known as the Least Square solution (LS). The above solution is only optimal in the case where the linear system is corrupted by Gaussian noise with a fixed variance. In practice, the system of linear equations may contain outliers. In other words, there are some pixels for which the residual of Eq.(3) is very large and can affect the solution. These outliers can be caused by local planar excursions and derivatives errors. Therefore, our idea is to estimate the 8 coefficients using robust statistics (Huber, 2003). We proceed as follows. First, equations are explored using subsamples of p linear equations (remember that each linear equation in (8) is provided by a pixel). For the problem at hand, p should be at least eight. Second, the solution is chosen according to the consensus measure based on residual errors. A Monte Carlo type technique is used to draw K random subsamples of p different equations/pixels. Figure 2 illustrates the algorithm.

Inlier detection The question now is: Given a subsample k and its associated solution $\mathbf{a}_k$, How do we decide whether or not a pixel is an inlier? In techniques dealing with geometrical features (points and lines) (Fischler and Bolles, 1981), this can be easily achieved using the distance in the image plane between the actual location of the feature and its mapped location. If this distance is below a given threshold then this feature is considered as an inlier; otherwise, it is considered as an outlier.

In our case, however, there are no geometrical features at all since only image derivatives are used. Therefore, our idea is to compute a robust estimation of standard deviation of the residual errors. In the exploration step, for each subsample k, the median of residuals was computed. If we denote by $\overline{M}$ the least median, then a robust estimation of the standard deviation of the residual is given by (Rousseeuw and Leroy, 1987):

$$\hat{\sigma} = 1.4826\,\left[1 + \frac{5}{N-p}\right]\,\sqrt{\overline{M}} \quad (10)$$

Once $\hat{\sigma}$ is known, any pixel j can be considered as an inlier if its residual error satisfies $|r_j| < 3\,\hat{\sigma}$.

Random sampling: Repeat the following three steps K times

1. Draw a random subsample of p different equations/pixels.
2. For this subsample, indexed by k, compute the eight coefficients, i.e. the vector $\mathbf{a}_k$, from the corresponding p equations using a linear system similar to (8).
3. For this solution $\mathbf{a}_k$, determine the median M_k of the squared residuals with respect to the whole set of N equations. Note that we have N residuals corresponding to the linear system (8).

Consensus step:

1. For each solution $\mathbf{a}_k, k = 1, \ldots, K$, compute the number of inliers among the entire set of equations/pixels (see below). Let n_k be this number.
2. Choose the solution that has the highest number of inliers. Let $\mathbf{a}_i$ be this solution where $i = \arg\max_k(n_k), k = 1, \ldots, K$
3. Refine $\mathbf{a}_i$ using the system formed by its inliers, that is, (9) is used without the outliers.

Figure 2: Recovering the eight coefficients using robust statistics.

The number of subsamples K A subsample is "good" if it consists of p good pixels. The number of subsamples is chosen such that the probability P_r that at least one of the K subsamples is good is very close to one (e.g., $P_r = 0.98$). Assuming that the whole set of equations may contain up to a fraction ϵ of outliers, the probability that at least one of the K subsamples is good is given by

$$P_r = 1 - [1 - (1 - \epsilon)^p]^K$$

Given a prior knowledge about the percentage of outliers ϵ the corresponding K can be computed by:

$$K = \frac{\log(1 - P_r)}{\log(1 - (1 - \epsilon)^p)}$$

For example, when $p = 20$, $P_r = 0.98$, and $\epsilon = 20\%$ we get $K = 337$ samples.

3.2 The SFM Parameters

Once the eight coefficients are recovered, it can be shown that the SFM parameters, i.e.

$\frac{V_x}{C}, \frac{V_y}{C}, \frac{V_z}{C}, \Omega_x, \Omega_y, \Omega_z, A$, and B, can be recovered by solving the non-linear equations (6). This is carried out using the Levenberg-Marquardt technique (Press et al., 1992). Non-linear algorithms need an initial solution. In order to get such initial solutions one can adopt assumptions for which Eq.(6) becomes linear. Then, the linear solution is refined using the Levenberg-Marquardt technique. In practice, one can use one of the following two assumptions for which Eq.(6) becomes linear in the parameters:

1. Assume that the translational velocity of the camera along its optical axis is very small compared to its lateral velocity, that is, $\frac{V_z}{V_x} << 1$ and/or $\frac{V_z}{V_y} << 1$. With this assumption, we can set V_z to 0 in Eq.(6) which can be easily solved for the remaining parameters.
2. Assume that the camera motion is a pure translation, then compute the translation velocity and the plane orientation using the resulting linear system.

We point out that the discrepancy between the linear solution and the true one depends on the realism of the made assumption.

It should be noticed that in practice when tracking a video sequence one can use the estimated 3D velocity for the previous frame as an initial solution for the current frame.

4 A ONE-STEP APPROACH

In this section, we propose a second approach that directly estimates the SFM parameters in one single step. To this end, Eqs.(4), (5), and (6) are inserted into Eq.(3). The result is a system with N non-linear equations relating the unknowns to the image derivatives. This can be solved using the Levenberg-Marquardt technique. For each pixel i, Eq. (3) gives a non-linear constraint having the form $f_i = 0$. Thus, the SFM parameters are obtained by minimizing the following cost function:

$$\min_{\mathbf{b}} \sum_{i=1}^{N} f_i^2 \tag{11}$$

where $\mathbf{b} = (\frac{V_x}{C}, \frac{V_y}{C}, \frac{V_z}{C}, \Omega_x, \Omega_y, \Omega_z, A, B)^T$.

The robust version of the one-step approach is obtained from Eq. (11) by retaining only the inlier pixels:

$$\min_{\mathbf{b}} \sum_{i=1}^{N} w_i \, f_i^2, \quad w_i = \begin{cases} 1 & \text{if the pixel } i \text{ is inlier} \\ 0 & \text{otherwise} \end{cases}$$

The detection of inlier pixels is performed using the paradigm described in Section 3.1.

This approach provides a direct estimation of the unknowns from the image derivatives and is expected to be more accurate than the two-step approach (see experiments below). Indeed, in the two-step approach, errors associated with the estimated 8 coefficients **a** will affect the estimation of the SFM parameters in the second step - solving Eq. (6).

5 THE DERIVATIVES

The spatial derivatives associated with the current image are calculated by convolution with derivatives of 2D Gaussian kernels. That is, $I_x = I * G_x$ and $I_y = I * G_y$ where

$$G_x = -\frac{1}{2\pi\sigma_s^4} \, x \, \exp\left(-\frac{x^2+y^2}{2\sigma_s^2}\right) \tag{12}$$

$$G_y = -\frac{1}{2\pi\sigma_s^4} \, y \, \exp\left(-\frac{x^2+y^2}{2\sigma_s^2}\right) \tag{13}$$

The temporal derivatives associated with the current image are calculated using difference approximation involving a temporal window centered on the current image. The weights of the images are taken from the derivatives of a 1D Gaussian kernel. That is, $I_t = I * G_t$ where

$$G_t = -\frac{1}{\sqrt{2\pi}\sigma_t^3} \, t \, \exp\left(-\frac{t^2}{2\sigma_t^2}\right) \tag{14}$$

The images can be smoothed before computing the temporal derivatives using Gaussian kernels having the same spatial scale σ_s. Figure 3 shows 11 weights approximating G_t whose σ_t is set to 2 frames. These weights correspond to 11 subsequent images. The smoothness achieved by the spatial and the temporal Gaussians is controlled by σ_s and σ_t, respectively.

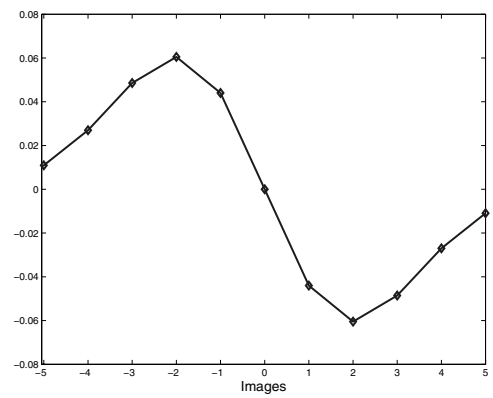


Figure 3: The 11 weights approximating the derivatives of 1D Gaussian whose σ_t is set to 2 frames.

6 EXPERIMENTS

Experiments have been carried out on synthetic and real images.

6.1 Synthetic Images

Experiments have been carried out on synthetic images featuring planar scenes. The texture of the scene is described by:

$$g(X_o, Y_o) = \sin(c_h\, X_o) + \sin(c_v\, Y_o)$$

where X_o and Y_o are the 3D coordinates expressed in the plane coordinates system, see Figure 4. The resolution of the synthesized images is 160×160 pixels. The constants c_h and c_v control the periodicity of the sine waves along each direction (in our example, these constants are set to 1.5). The 3D plane was placed at 100cm from the camera whose focal length is set to 1000 pixels. In order to study the performance of the developed approaches, we have proposed the following framework.

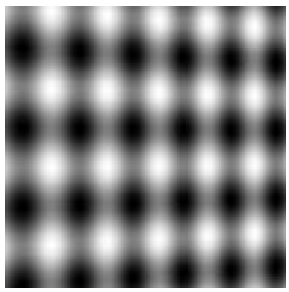


Figure 4: A computer generated image of a 3D plane. The plane is rotated about 40 degrees about the fronto-parallel plane.

A synthesized image sequence of the planar scene is generated according to a nominal camera velocity $(\mathbf{V}_n, \Omega_n)$. A reference image is then fixed for which the image derivatives are computed and for which we like to compute the SFM parameters. Since synthetic data are used ground-truth values for the image derivatives and for the SFM parameters are known. The nominal 3D velocity $(\mathbf{V}_n, \Omega_n)$ is set to $(10cm/s, 10, 1, 0.1rad/s, 0.1, 0.1)^T$. The corresponding linear system (8) is then gradually corrupted by a Gaussian noise having an increasing variance. Our approach is then used to solve the SFM problem using the corrupted linear system.

The discrepancies between the estimated parameters and their ground truth are then evaluated. In our case, the SFM parameters are given by three vectors (see Figure 1): the scaled translational velocity, (ii) the rotational velocity, and (iii) the plane normal in the camera coordinate system. Thus, the accuracy of estimated parameters can be summarized by the angle between the direction of the estimated vector and its ground truth direction.

The goal is to quantify the accuracy of the two-step approach (Section 3). To this end, the simulated linear system was corrupted by a pure Gaussian noise as well as by a 15% of outliers. The standard deviation of the Gaussian noise is gradually increased as a percentage of the mean of the spatio-temporal derivatives (ground truth values). The outliers are uniformly selected in the image.

Figure 5 illustrates the obtained average errors associated with the SFM parameters as a function of the Gaussian noise (using the two-step approach). The solid curve corresponds to the Least Square solution (no robust statistics), and the dotted curve to the robust solution. In this figure, each average error was computed with 50 random realizations. As can be seen, unlike the LS solution the second solution is much more accurate.

Two-step approach versus one-step approach Figure 6(a) shows the average errors associated with the translational and rotational velocities as a function of a pure Gaussian noise. The solid curve corresponds to the two-step approach (Section 3) and the dashed curve corresponds to the one-step approach (Section 4). Figure 6(b) shows the same comparison when both Gaussian noise and outliers are added. As can be seen, the second approach seems to be more accurate than the first one. This behavior holds for the plane orientation.

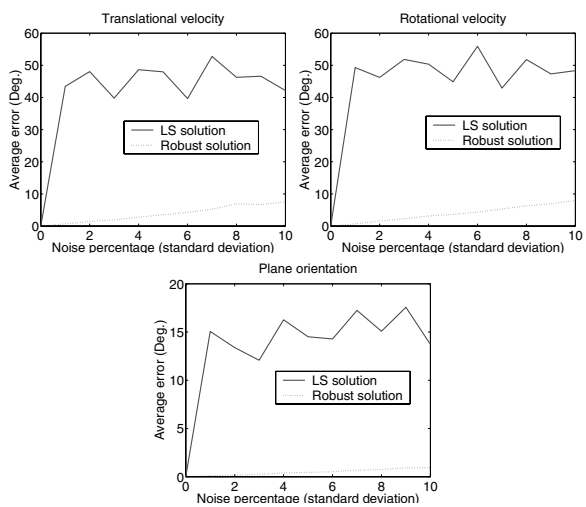


Figure 5: Average errors associated with the SFM parameters when the system is corrupted by both a Gaussian noise and 15 $\%$ of outliers.

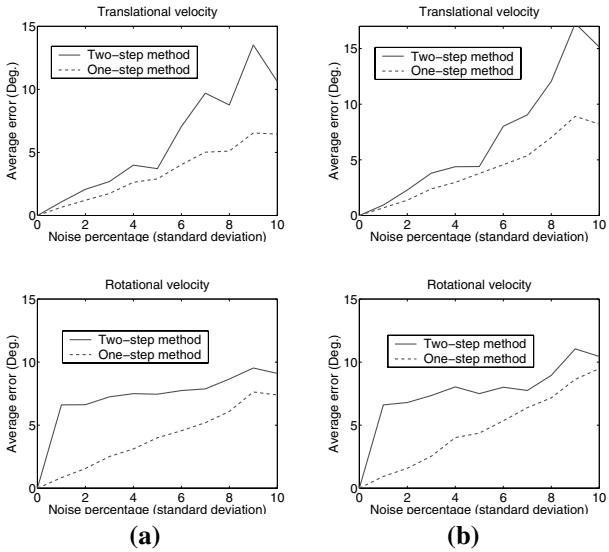


Figure 6: Two-step approach (solid curve) versus the one-step approach (dashed curve). **(a)** Gaussian noise. **(b)** Gaussian noise and outliers.

Table 1: Results of the first experiment.

	Translation	Rotation	A	B
LS sol.	(-.99,-.12,.01)	(-.13,.99,-.01)	.04	-.01
Robust sol.	(-.98,-.17,.01)	(-.18,.98,-.01)	.04	-.01
One step	(-.98,-.18,.01)	(-.17,.98,-.01)	.04	-.00

6.2 Real Images

The first experiment was conducted on a video sequence captured by a moving camera, see Figure 7. This video was retrieved from `ftp://csd.uwo.ca/pub/vision`. We have used 11 subsequent images to compute the SFM parameters associated with the central image (frame 6). The results are summarized in Table 1. The first row corresponds to the LS solution (the two-step approach), the second row to the robust solution (the two-step approach), and the third row to the one-step approach. As can be seen, the motion is essentially a lateral motion. Note the consistency of the results obtained by the three methods. The second experiment was conducted on the sequence depicted in Figure 8(a). The sequence was retrieved from `http://www.cee.hw.ac.uk/~mtc/sofa`. The obtained results are summarized in Table 2. As can be seen, the camera velocity is a rotation about the optical axis combined with a translation about the same axis. Figure 8(b) shows the map of outlier pixels.

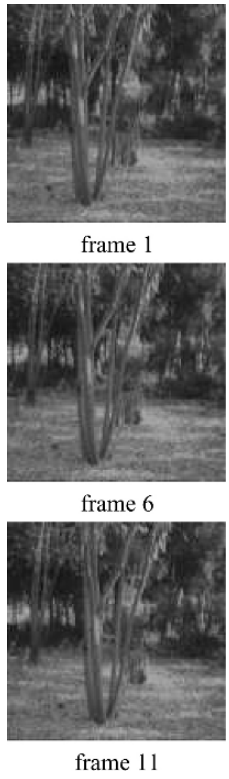


Figure 7: The first experiment. Frame 6 represents the current image for which the SFM parameters are computed. The temporal derivatives are computed using 11 subsequent images.

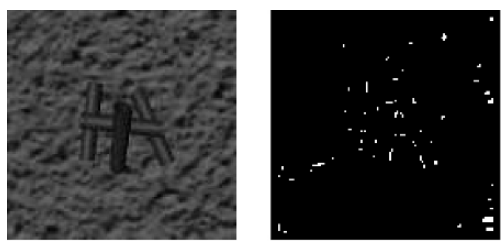


Figure 8: The second experiment. **(a)** The current image for which the SFM parameters are computed. The temporal derivatives are computed using 7 subsequent images. **(b)** The map of outlier pixels.

Table 2: Results of the second experiment.

	Translation	Rotation	A	B
LS sol.	(.00,13,.99)	(.11,.0,-.99)	.34	-1.
Robust sol.	(-.01,.08,.99)	(.07,.0,-.99)	.46	-.89
One step	(.14,.08,.98)	(.07,-.12,-.98)	.55	-.15

7 CONCLUSION

We presented a novel paradigm for the planar SFM problem where only image derivatives have been used. No feature extraction or matching is needed using this paradigm. Two different strategies have been proposed. The first strategy estimates the parameters of the 2D motion field then the SFM parameters. The second strategy directly estimates the SFM parameters.

Methods from robust statistics were included in both strategies in order to get an accurate solution even when data contain outliers. This is very useful for scenes which are not fully described by planar surfaces. The developed strategies do not rely on pixel velocities. However, these velocities are a byproduct of them.

REFERENCES

Adams, H., Singh, S., and Strelow, D. (2002). An empirical comparison of methods for image-based motion estimation. In *IEEE International Conference on Intelligent Robots and Systems*.

Brodsky, T. and Fermuller, C. (2002). Self-calibration from image derivatives. *International Journal of Computer Vision*, 48(2):91–114.

Brooks, M., Chojnacki, W., and Baumela, L. (1997). Determining the egomotion of an uncalibrated camera from instantaneous optical flow. *Journal of the Optical Society of America A*, 14(10):2670–2677.

Fischler, M. and Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communication ACM*, 24(6):381–395.

Huber, P. (2003). *Robust Statistics*. Wiley.

Jonathan, A., M., and Sclaroff, S. (2002). Recursive estimation of motion and planar structure. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C*. Cambridge University Press.

Qian, G. and Chellapa, R. (2004). Structure from motion using sequential Monte Carlo methods. *International Journal of Computer Vision*, 59(1):5–31.

Rousseeuw, P. and Leroy, A. (1987). *Robust Regression and Outlier Detection*. John Wiley & Sons, New York.

Srinivasan, S. (2000). Extracting structure from optical flow using fast error search technique. *International Journal of Computer Vision*, 37(3):203–230.

Weng, J., Huang, T. S., and Ahuja, N. (1993). *Motion and Structure from Image Sequences*. Springer-Verlag, Berlin.

Xiang, T. and Cheong, L. (2003). Understanding the behavior of SFM algorithms: A geometric approach. *International Journal of Computer Vision*, 51(2):111–137.

Zelnick-Manor, L. and Irani, M. (2000). Multi-frame estimation of planar motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1105–1116.

Zucchelli, M., Jose, S., and Christensen, H. (2002). Multiple plane segmentation using optical flow. In *British Machine Vision Conference*.

COMBINING TWO METHODS TO ACCURATELY ESTIMATE DENSE DISPARITY MAPS

Agustín Salgado and Javier Sánchez
Computer Science Department, University of Las Palmas de Gran Canaria
35017 Las Palmas de Gran Canaria, Spain
asalgado@dis.ulpgc.es, jsanchez@dis.ulpgc.es

Keywords: Disparity map, optical flow, stereoscopic vision.

Abstract: The aim of this work is to put together two methods in order to improve the solutions for the problem of 3D geometry reconstruction from a stereoscopic pair of images. We use a method that we have developed in recent works which is based on an energy minimisation technique. This energy yields a partial differential equation (PDE) and is well suited for accurately estimating the disparity maps. One of the problems of this kind of techniques is that it depends strongly on the initial approximation. For this reason we have used a method based on graph–cuts which has demonstrated to obtain good initial guess.

1 INTRODUCTION

In this paper we have put together two methods for computing disparity maps. The first one is based on graph–cuts energy minimisation (Kolmogorov et al., 2001), (Boykov et al., 2004). This method has demonstrated to give good results in integer precision which is enough for a set of applications. If we are looking for better accuracy then it is necessary to use a different technique. In this case we use a method that we have developed recently and which is described in paper (Alvarez et al., 2002). We have also implemented a similar method for optical flow estimation which is explained in (Alvarez et al., 2000). These methods are based on an energy minimisation approach. When we minimize the energy we obtain a system of PDEs which are then embedded into a gradient descent method to obtain the solution. One of the problems of these methods is that they need a good initial approximation in order to obtain a precise solution. In previous works we have always used a correlation based technique to compute this approximation. Comparing graph–cuts and correlation based methods the first one provides more stable solutions.

In this paper we show that the combination of graph–cuts and PDE based methods improves the accuracy of the solution with respect to other initial approximation techniques such as correlation. In the experimental results we compare numerically the different approaches through a synthetic sequence of a corridor and also we show several results for a real stereoscopic pair of images – the Tsukuba sequence.

2 GRAPH-CUTS METHOD

The minimum cut/maximum flow algorithms on graphs emerged as an increasingly useful tool for exact or approximate energy minimisation in low-level vision. Stereo is a classical vision problem where graph-based energy minimisation methods have been successfully applied. The goal of stereo is to compute the correspondence between pixels of two or more images of the same scene obtained by cameras with slightly different view points. Any stereo images of multi-depth objects contain occluded pixels. The presence of occlusions adds significant technical difficulties to the problem of stereo.

The energy function for a configuration f is of the form

$$E(f) = E_{data}(f) + E_{occ}(f) + E_{smooth}(f) \quad (1)$$

The three terms here include

- a data term E_{data}, which results from the differences in intensity between corresponding pixels;
- an occlusion term E_{occ}, which imposes a penalty for making a pixel occluded; and

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 137–143

- a smoothness term E_{smooth}, which makes neighboring pixels in the same image tend to have similar disparities.

3 ENERGY BASED METHOD

The method we use in this paper is energy based. It has the following features:

- We consider a weakly calibrated stereoscopic system. The stereoscopic system is not calibrated and only the knowledge of the so-called fundamental matrix is known.
- This method addresses the problem of accurately determining the dense disparity map while regularizing it along the contours of the gray level image and inhibiting smoothing across the image discontinuities.
- We apply a multi–resolution scheme in order to avoid convergence to irrelevant minima.

The energy function that we propose for 3D geometry reconstruction is as follows:

$$E(\lambda) = \int \left(I_l(\mathbf{x}) - I_r(\mathbf{x} + \mathbf{h}(\lambda(\mathbf{x}))\right)^2 d\mathbf{x} + C \int \Phi\left(\nabla I_l, \nabla\lambda\right) d\mathbf{x}. \quad (2)$$

In this case we have a matching function, $\mathbf{h}$, that depends on a scalar function, λ. This scalar function represents the displacement of pixels on the epipolar lines. In this case $\Phi\left(\nabla I_l, \nabla\lambda\right) = \nabla\lambda^t \cdot D\left(\nabla I_l\right) \cdot \nabla\lambda$, $D\left(I_l \nabla\right)$ is a regularized projection matrix perpendicular to ∇I_l,

$$D\left(\nabla I_l\right) = \frac{1}{|\nabla I_l|^2 + 2\upsilon^2} \cdot \left\{ \begin{bmatrix} \frac{\partial I_l}{\partial y} \\ \frac{-\partial I_l}{\partial x} \end{bmatrix} \begin{bmatrix} \frac{\partial I_l}{\partial y} \\ \frac{-\partial I_l}{\partial x} \end{bmatrix}^t + \upsilon^2 Id \right\} \quad (3)$$

where *Id* denotes the identity matrix. This projection has been introduced by Nagel and Enkelmann in the context of optical flow estimation.

After minimising this energy and applying a gradient descent method we obtain the following diffusion–reaction PDE:

$$\frac{\partial\lambda}{\partial t} = C \operatorname{div}\left(D\left(\nabla I_l\right) \nabla\lambda\right) + \left(I_l(\mathbf{x}) - I_r^{\lambda}(\mathbf{x})\right) \cdot \left(\frac{-b\left(\frac{\partial I_r}{\partial x}\right)^{\lambda}(\mathbf{x})}{\sqrt{a^2+b^2}} + \frac{a\left(\frac{\partial I_r}{\partial y}\right)^{\lambda}(\mathbf{x})}{\sqrt{a^2+b^2}} \right) \quad (4)$$

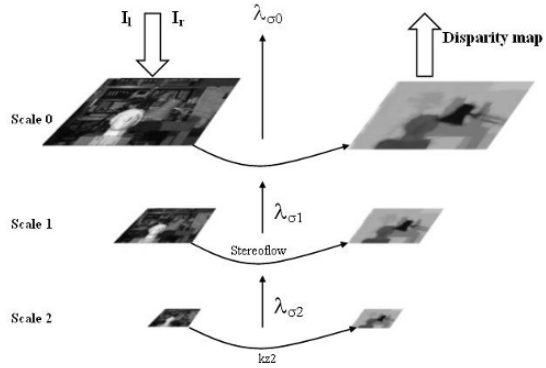


Figure 1: Combination of kz2 and stereoFlow methods.

The details of this method could be found in paper (Alvarez et al., 2002).

4 COMBINING GRAPH-CUTS AND STEREOFLOW METHOD

In this section, we explain how the graph-cuts (kz2) and the previous explained PDE (stereoFlow) methods work together for estimating the dense disparity map. The graph-cuts method labels the image obtaining a disparity map in integer precision. The stereoFlow method obtains a disparity map in float precision. To improve the perfomance of our method, we do not apply the graph-cuts method in the input pair of images. Using a pyramidal approach we scale the image *"n"* times. The number of scales is a parameter defined by user.

The basic idea of embedding our method in a pyramidal approach is as follows: we replace the images I_l and I_r by $I_l^{\sigma} := Z(I_l)$ and $I_r^{\sigma} := Z(I_r)$, where $Z(\ldots)$ is the zoom operator. Thus, we do a $2X$ zoom over each image. We start with a large initial scale σ_0. Next, we choose a number of scales $\sigma_n < \sigma_{n-1} < \cdots < \sigma_0$ and for each scale σ_i we do a zoom. When we reach last scale (σ_n), we compute the disparity λ_{σ_n} with kz2 or with a correlation based technique. Thus, we have an initial approximation. Next, we compute the disparity λ_{σ_i} as the asymptotic state of the above PDE with initial data $\lambda_{\sigma_{i+1}}$. So, the disparity of I_l and I_r is defined by λ_{σ_0}. In Figure 1, we see an example how this algorithm works.

Both correlation–based and graph-cuts methods spend much CPU time to compute the disparity maps. As we can see in Figure 1, the kz2 is applied at the smallest size of the images, so we assure that it is carried out faster than at larger images. Then the stereoFlow technique is applied in the rest of the scales.

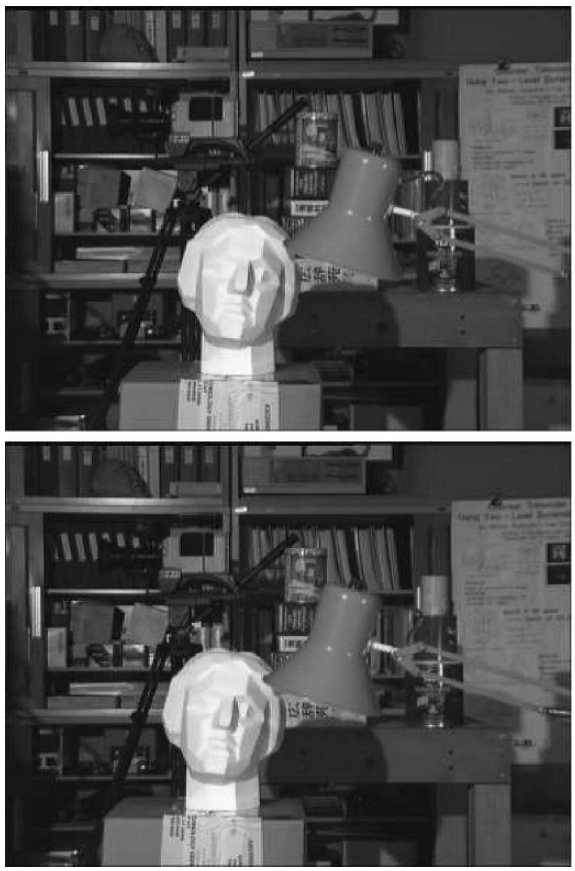


Figure 2: Stereoscopic pair for the Tsubuka sequence.

Figure 3: Disparity map obtained through kz2.

Figure 4: Disparity map obtained through correlation + stereoflow (using a 21x21 correlation window).

5 EXPERIMENTAL RESULTS

In this section we present a comparison between the graph–cuts stereo method (kz2) and the combination of our method (stereoFlow) with different initial approximation (such as kz2 or correlation based technique). We have used three datasets in our tests: a stereoscopic pair from the University of Tsukuba (Figure 2), a stereoscopic pair of a synthetic cylinder (Figure 6) and a stereoscopic pair of a synthetic corridor contaminated by noise of variance 10 (Figure 11).

In paper (Kolmogorov et al., 2001), the head of Tsukuba was used to show the results obtained with graph-cuts stereo method (kz2) in comparison with similar methods. We have used the same dataset to show how our algorithm improves the initial approximation given by kz2 (with/without scales).

The number of zooms defined by user depends on the scene motion. An overzooming (or overscaled) gives us a bad initial approximation, so our method converges to irrelevant local minima. In Figure 3 we can see the disparity map obtained by kz2. Both correlation based techniques and graph-cuts methods spend much CPU time so we must decide between a few or large number of scales.

We have tested our method using the synthetic cylinder dataset to compare its accuracy with different initial approximations. For the initial value we consider two possibilities. The first one is to use the result of a simple classic method for estimating the disparity, for instance a correlation based technique. The second one is to use a graph-cuts method which gives us a disparity map in integer precision.

We have computed the euclidean error between the output of our algorithm and the ideal disparity map, to see the accuracy of our method. In the table 1 and 2 we show the results obtained by the combination of a correlation–based method with stereoFlow, the result for the kz2 and the result for the combination of kz2 and stereoFlow, for the synthetic cylinder and corridor pairs.

Figure 5: Disparity map obtained through kz2 + stereoflow.

From these results we may appreciate that the combination of correlation and stereoFlow method gives better results than the kz2 method and that the combination of kz2 and stereoFlow improves the solution of the correlation–based one.

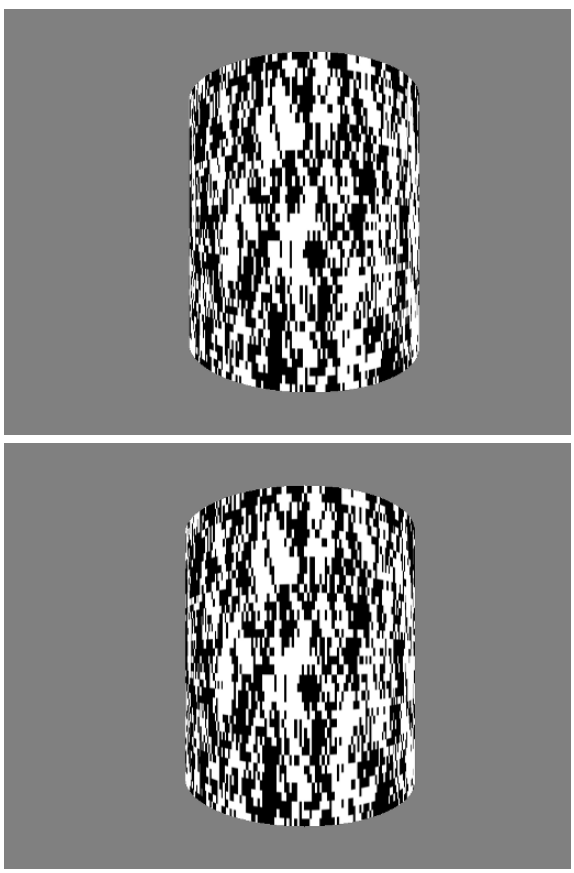


Figure 6: Stereoscopic pair for the synthetic cylinder.

Table 1: Euclidean error obtained with cylinder images for various tests.

Euclidean error		
Method	**Disparity range**	
	Scale=0	**Scale**=1
kz2	0.24	0.24
Corr+StereoFlow	0.22	0.21
kz2+StereoFlow	0.20	0.19

In the table 1 we compare the solution for two different configurations: In the first one the pyramidal scheme is reduced to only one scale (Scale = 0) and in the second one we use two scales (Scale = 1). If we compare them, we may conclude that the use of the pyramidal approach improves the solutions for both methods (correlation + stereoFlow and kz2 + stereoFlow) and that the result for kz2 + stereoFlow is still better than using the correlation–based method.

In Figures 8, 9 and 10, we see the visual results for the synthetic cylinder obtained for each method. Looking at the figures the result obtained with the latter is smoother and more accurate. All the experimental results are improved with the combined method and in most cases the improvement is greater than a 16% for the euclidean error for the same scale.

We have also tested our method using the synthetic corridor pairs to compare its accuracy with different initial approximations. Corridor pair was generated by a ray tracing program, the ground truth is in float precision. We used three pairs for the corridor, the original and other two contaminated by noise of variance 10, 100. In these tests we compare not only the precision of the methods each other, and also the stability of their results for noisy images.

In the Table 2 we show the results obtained (by the combination of $kz2$, $Corr + SF$ and $kz2 + SF$). Analizing these results we see that the $correlation +$

Figure 7: Ideal disparity map for the cylinder images.

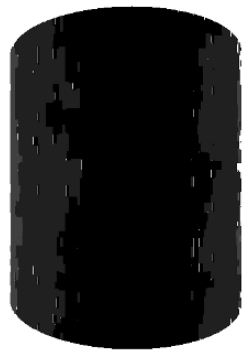


Figure 8: Disparity map obtained through kz2.

Figure 9: Disparity map obtained through correlation + stereoFlow.

Figure 10: Disparity map obtained through kz2 + stereoFlow.

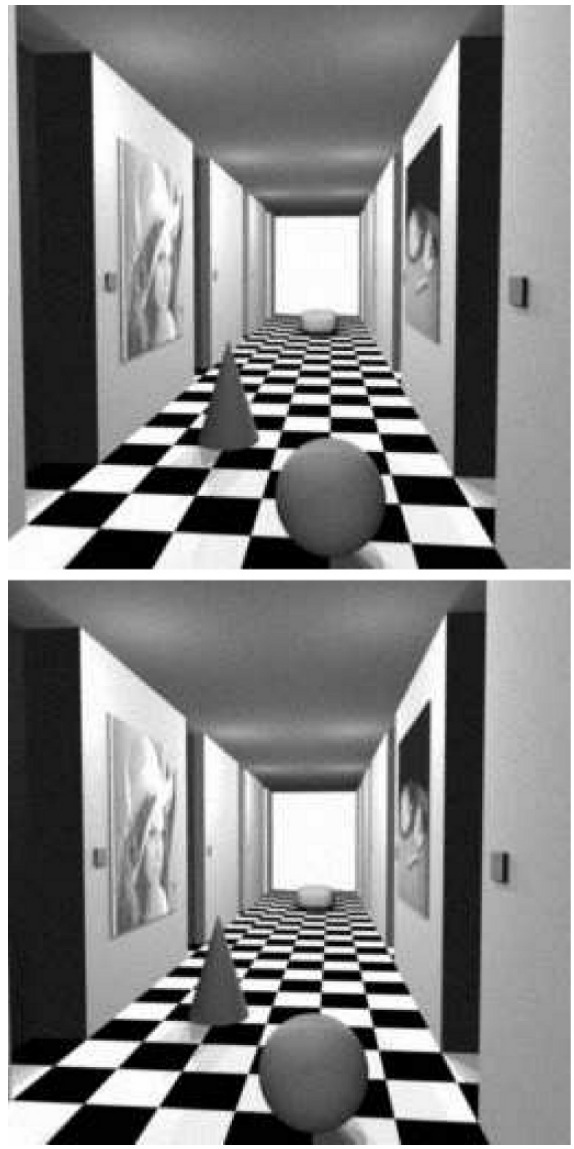


Figure 11: Stereoscopic pair for the Corridor sequence contaminated by Noise of Variance 10.

$stereoFlow$ method gives better results than the kz2 method and that the $kz2 + stereoFlow$ improves the solution of the correlation–based one. The kz2 results are very accuracy when we test kz2 using the original images of the corridor (without noise). However, when we add noise to this pair the results have not been as well as kz2+SF.

In Figures 14, 15 and 16 we see the visual results for the corridor sequence (corridor pair contaminated by noise of variance 10, Figure 11) obtained for each method. All the experimental results are improved with the combined method and in most cases the improvement is greater than a 26% for the euclidean error.

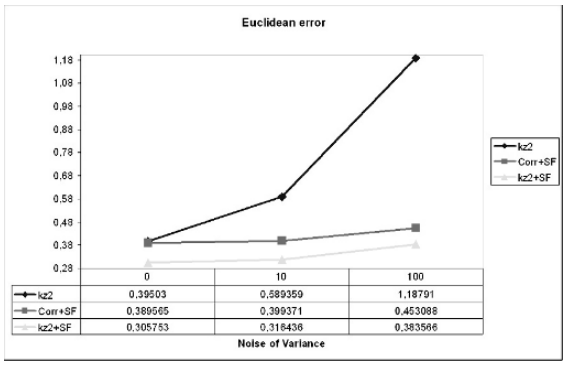


Figure 12: Euclidean error obtained in the corridor pairs (original, noise of variance 10 and 100).

In Figure 12, we can see the stability of these three methods. Kz2 method is very sensitive for noisy images. StereoFlow is very stable as kz2 method as correlation based technique.

6 CONCLUSIONS

In this work we have combined two different techniques on disparity maps estimation in order to obtain more accurate and reliable solutions. We have used a pixel precision method based on graph–cuts as initialization for another method based on PDEs. The latter depends on an initial approximation which is supported by the former one. The solution we obtain is in float precision and the accuracy is considerably improved. We have compared the combination of the PDE and graph–cuts with the combination of the PDE

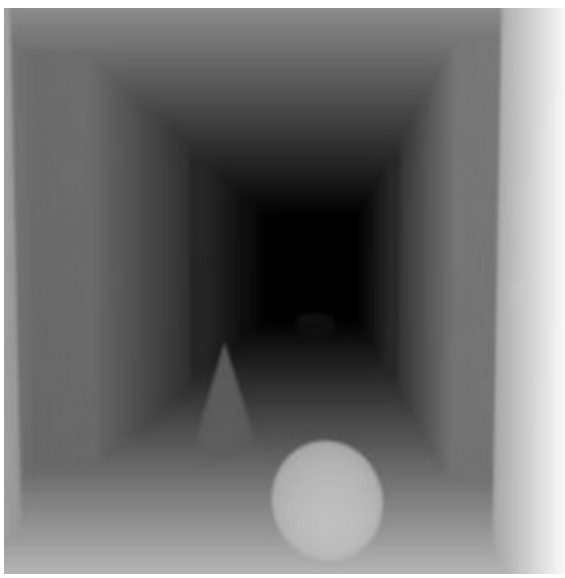


Figure 13: Ideal disparity map for the Corridor pair.

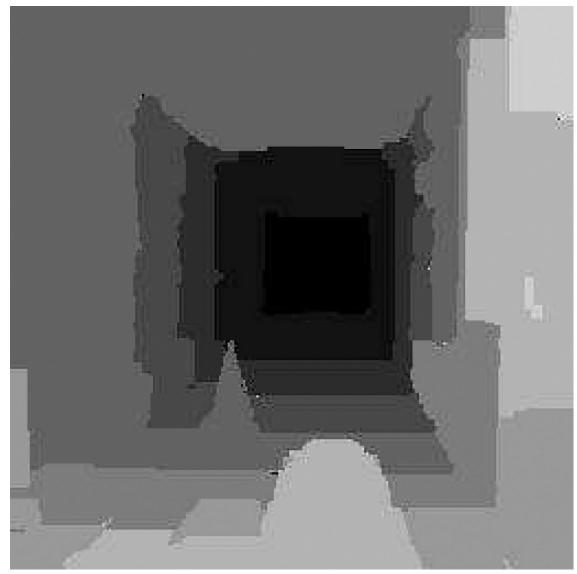


Figure 14: Disparity map obtained through kz2.

Figure 15: Disparity map obtained through correlation + stereoFlow.

Table 2: Result of our method applied to the Stereo pair in Figure 11

Euclidean error		
Method	**Disparity range**	**% improvements**
kz2	0.5893	-86.25%
Corr+SF	0.3993 (2 scales)	-26.21%
kz2+SF	0.3164 (0 scales)	+0.0%

Figure 16: Disparity map obtained through kz2 + stereoFlow.

and a correlation–based method. We may conclude that the use of the kz2 at the first stage provides better results than the correlation method.

ACKNOWLEDGEMENTS

This paper has been partly funded by the Spanish Ministry of Science and Technology and FEDER through the research project TIC2003-08957.

REFERENCES

L. Alvarez, R. Deriche, J. Sánchez and J. Weickert. Dense Disparity Map Estimation Respecting Image Discontinuities: A PDE and Scale-Space Based Approach. Journal of Visual Communication and Image Representation, 13, pp. 3-21. 2002.

L. Alvarez J. Weickert and J. Sánchez. Reliable Estimation of Dense Optical Flow Fields with Large Displacements. International Journal of Computer Vision, 39, pp. 41-56, 2000.

V. Kolmogorov and R. Zabih, Computing Visual Correspondence with Occlusions using Graph Cuts. In International Conference on Computer Vision (ICCV), July 2001.

Y. Boykov and V. Kolmogorov, An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. In IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), September 2004.

PRECISE DEAD-RECKONING FOR MOBILE ROBOTS USING MULTIPLE OPTICAL MOUSE SENSORS

Daisuke Sekimori
Akashi National College of Technology
Akashi, Hyogo 674-8501, Japan
sekimori@akashi.ac.jp

Fumio Miyazaki
Graduate School of Engineering Science, Osaka University
Toyonaka, Osaka 560-8531, Japan
miyazaki@me.es.osaka-u.ac.jp

Keywords: Dead-reckoning, mobile robot, optical mouse sensor.

Abstract: In this paper, in order to develop an accurate localization for mobile robots, we propose a dead-reckoning system based on increments of the robot movements read directly from the floor using optical mouse sensors. The movements of two axes are measurable with an optical mouse sensor. Therefore, in order to calculate a robot's deviation of position and orientation, it is necessary to attach two optical mouse sensors in the robot. However, it is also assumed that a sensor cannot read the movements correctly due to the condition of the floor, the shaking of the robot, etc. To solve this problem, we arrange multiple optical mouse sensors around the robot and compare sensor values. By selecting reliable sensor values, accurate dead-reckoning is realized. Finally, we verify the effectiveness of this algorithm through several experiments with an actual robot.

1 INTRODUCTION

For a mobile robot to move around autonomously, it is necessary for it to possess the ability to estimate its position and orientation. The localization of mobile robots is roughly divided into those using internal sensors and those using external sensors. The method using internal sensors is known as dead-reckoning, mainly, and estimates position by measuring and accumulating the rotation of the wheel with the rotary encoder, etc. Dead-reckoning is a convenient estimating method using only internal sensors. However, the accuracy of estimation decreases as the movement becomes longer since the errors of the transformations and wheel slippage accumulate. On the other hand, the method using external sensors estimates the position by measuring positions of a landmark in the environment with a vision sensor or a range sensor. Some error is always caused by resolution or the noise of the sensor; accumulated errors are not caused as such by dead-reckoning. Therefore, because both methods have their respective merits and demerits, the two methods are often used together (Cox, 1989) (Watanabe and Yuta, 1990) (Chenavier and Crowley, 1992).

In the case of the estimation method using both dead-reckoning and an external sensor, it is advantageous to improve the accuracy of dead-reckoning. As for the reason, in general, many of the external sensors are expensive, and processing is very complex. Moreover, estimation methods using external sensors need to have a previously installed landmark in the environment. By improving the accuracy of dead-reckoning and reducing the part that depends on the method using the external sensor, the hardware and software costs of the robot can be decreased, and the time needed to install a landmark can be omitted.

In this paper, in order to develop an accurate localizaion for mobile robots, we propose a dead-reckoning system based on increments of the robot movements read directly from the floor using optical mouse sensors (Fujimoto et al., 2002). The movements of two axes are measurable with an optical mouse sensor. Therefore, in order to calculate a robot's deviation of position and orientation, it is necessary to attach two optical mouse sensors in the robot (Tobe et al., 2004) (Singh and Waldron, 2004) (Cooney et al., 2004). However, it is also expected that a sensor cannot read the movements correctly due to the condition of the floor, the shaking of the robot, etc. To solve this problem, we arrange multiple optical mouse sensors around the robot and compare sensor values. By selecting reliable sensor values, accurate dead-reckoning is achieved.

In Section 2, we explain the optical mouse sensor. In Section 3, we describe the algorithm of dead-reckoning based on optical mouse sensors. Finally,

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 145–151

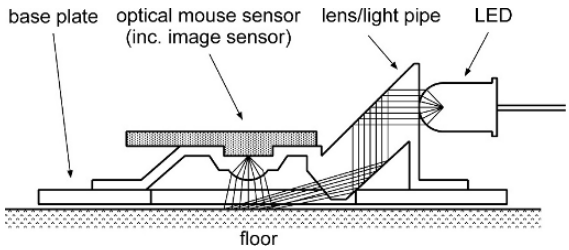


Figure 1: Structure of optical mouse sensor.

Table 1: Specifications of the optical mouse sensor (Agilent Technologies, HDNS-2051).

resolution	800 counts/inch
max speed	14 inch/sec
scanning frequency	2300 Hz
power supply	5 volts

in Section 4, we verify the effectiveness of this algorithm through several experiments with an actual robot.

2 OPTICAL MOUSE SENSOR

An optical mouse sensor is built into an optical mouse for personal computers, and measures non-contact movements. It is maintenance free and not influenced by floor friction.

The principle of the optical mouse sensor is that the installed small image sensor reads the change in the image information on the floor and the optical mouse sensor measures movement. The structure of the optical mouse sensor is shown in Figure 1. An optical mouse sensor takes a floor picture irradiated by a LED through a lens, with the image sensor located on the sensor undersurface. Changes in the pictures taken are processed within the sensor and transformed into distance information. Finally the sensor outputs a two phase pulse from the ports. The main specifications of the optical mouse sensor(Agilent Technologies, HDNS-2051) used in our research are shown in Table 1.

3 DEAD-RECKONING BASED ON OPTICAL MOUSE SENSORS

This section describes the basic equation for dead-reckoning based on optical mouse sensors and the comparison method for increasing the reliability of mouse sensor values. In this method, the mobility range of the robot is limited to the floor whereby the optical mouse sensor can initially measure the movement.

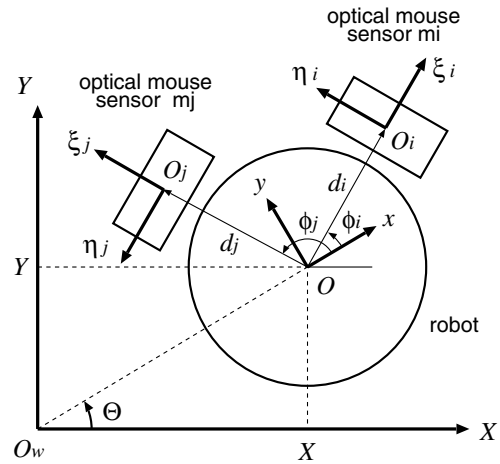


Figure 2: Configuration of robot and optical mouse sensors.

3.1 Basic Equation

Movement of the direction of two axes is measurable by one optical mouse sensor. Therefore, movement (translation and rotation) of the robot which moves through a plane is calculable by using two optical mouse sensors.

Firstly, a robot and two optical mouse sensors m_i, m_j are arranged as shown in Figure 2. The world coordinate system $(O_w - XY)$ is placed on the floor, the robot coordinate system $(O - xy)$ is placed on the robot center, and coordinate systems $(O_i - \xi_i\eta_i)$, $(O_j - \xi_j\eta_j)$ are put on the center of two optical mouse sensors m_i, m_j attached to the robot. However, axes x_{m_i}, x_{m_j} of each sensor are located in a radial direction from the robot center [1]. The positions of each sensor in terms of the robot coordinate system are expressed by $[d_i \cos\phi_i, d_i \sin\phi_i]^T$, $[d_j \cos\phi_j, d_j \sin\phi_j]^T$. The relation movement $[\Delta\xi_i, \Delta\eta_i]^T$, $[\Delta\xi_j, \Delta\eta_j]^T$ measured by each sensor and movement $[\Delta x, \Delta y, \Delta\theta]^T$ of the robot center is expressed as follows:

$$\begin{bmatrix} C\phi_i & -S\phi_i \\ S\phi_i & C\phi_i \end{bmatrix} \begin{bmatrix} \Delta\xi_i \\ \Delta\eta_i \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + \Delta\theta \begin{bmatrix} -d_i S\phi_i \\ d_i C\phi_i \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} C\phi_j & -S\phi_j \\ S\phi_j & C\phi_j \end{bmatrix} \begin{bmatrix} \Delta\xi_j \\ \Delta\eta_j \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + \Delta\theta \begin{bmatrix} -d_j S\phi_j \\ d_j C\phi_j \end{bmatrix} \quad (2)$$

[1] It is our goal to decrease the number of parameters used for this method, and the basic equation can be derived regardless of ξ_i, ξ_j axial direction of each sensor.

Here, $S\phi_*$ and $C\phi_*$ mean $\sin\phi_*$ and $\cos\phi_*$ respectively, and use this notation as follows. Moreover, upper formulas are arranged as follows:

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{a} \tag{3}$$

$$\boldsymbol{u} = [\Delta x, \Delta y, \Delta\theta]^T,$$

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & -d_i S\phi_i \\ 0 & 1 & d_i C\phi_i \\ 1 & 0 & -d_j S\phi_j \\ 0 & 1 & d_j C\phi_j \end{bmatrix},$$

$$\boldsymbol{a} = \begin{bmatrix} \Delta\xi_i C\phi_i - \Delta\eta_i S\phi_i \\ \Delta\xi_i S\phi_i + \Delta\eta_i C\phi_i \\ \Delta\xi_j C\phi_j - \Delta\eta_j S\phi_j \\ \Delta\xi_j S\phi_j + \Delta\eta_j C\phi_j \end{bmatrix}$$

Here, elements of matrix $\boldsymbol{A}$ and vector $\boldsymbol{a}$ are replaced with A_{pq} and $a_p (p = 1, 2, 3, 4; q = 1, 2, 3)$ respectively. Furthermore, the squared error E_{ij} of movements is defined as follows:

$$E_{ij} = \sum_{p=1}^{4} (A_{p1}\Delta x + A_{p2}\Delta y + A_{p3}\Delta\theta - a_p)^2 \tag{4}$$

The movement $\boldsymbol{u} = [\Delta x, \Delta y, \Delta\theta]^T$ that has the minimum square error E_{ij} is determined by using the following equation.

$$\boldsymbol{u} = \boldsymbol{A}^{-}\boldsymbol{a} \tag{5}$$

Here, matrix $\boldsymbol{A}^{-}$ means a pseudo-inverse matrix of $\boldsymbol{A}$.

After movement $\boldsymbol{u}$ of the robot can be determined, dead-reckoning is computed using the following equation and robot position $[X_t, Y_t, \Theta_t]^T$ in terms of the world coordinate system is determined. In addition, $[X_{t-1}, Y_{t-1}, \Theta_{t-1}]^T$ expresses the position at a pre-measurement point.

$$\begin{bmatrix} X_t \\ Y_t \\ \Theta_t \end{bmatrix} = \begin{bmatrix} X_{t-1} + \Delta x\, C\Theta_{t-1} - \Delta y\, S\Theta_{t-1} \\ Y_{t-1} + \Delta x\, S\Theta_{t-1} + \Delta y\, C\Theta_{t-1} \\ \Theta_{t-1} + \Delta\theta \end{bmatrix} \tag{6}$$

3.2 Comparison of Values of Optical Mouse Sensors

Robot movements may be incorrectly measured by the optical mouse sensor due to robot speed, robot shaking, the condition of the floor, etc. When errors arise in only one optical mouse sensor between two optical mouse sensors (since the squared error E_{ij} in (4) will be large), error is detectable by supervising the value of E_{ij}. However, when an error arises in both of two mouse sensors, there is no corroboration to which the value of E_{ij} becomes large. That is, error is undetectable when only supervising the value of E_{ij}. Thus, we proposed a method of computing robot movements by comparison of the optical mouse sensor values and by selecting reliable sensor values.

The number of optical mouse sensors is N, the squared errors E_{ij} $(i = 1 \cdots N, j = 1 \cdots N (i \neq j))$ of all optical mouse sensor values are calculated. Then, threshold E_{th} of E_{ij} is decided, and accuracy of a measurement value is evaluated by the following equation.

$$r_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} \delta_{ij}, \quad \delta_{ij} = \begin{cases} 1 & (E_{ij} \leq E_{th}) \\ 0 & (E_{ij} > E_{th}) \end{cases} \tag{7}$$

Here, r_i expresses the reliability of optical mouse sensor m_i. This reliability is computed to each optical mouse sensor, and optical mouse sensors $m_\alpha, m_\beta, \cdots$ with high reliability are elected using threshold r_{th}. And the following equation is derived using those values.

$$\boldsymbol{B}\boldsymbol{u} = \boldsymbol{b} \tag{8}$$

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 & -d_\alpha S\phi_\alpha \\ 0 & 1 & d_\alpha C\phi_\alpha \\ 1 & 0 & -d_\beta S\phi_\beta \\ 0 & 1 & d_\beta C\phi_\beta \\ : & : & : \\ : & : & : \end{bmatrix},$$

$$\boldsymbol{b} = \begin{bmatrix} \Delta\xi_\alpha C\phi_\alpha - \Delta\eta_\alpha S\phi_\alpha \\ \Delta\xi_\alpha S\phi_\alpha + \Delta\eta_\alpha C\phi_\alpha \\ \Delta\xi_\beta C\phi_\beta - \Delta\eta_\beta S\phi_\beta \\ \Delta\xi_\beta S\phi_\beta + \Delta\eta_\beta C\phi_\beta \\ : \\ : \end{bmatrix}$$

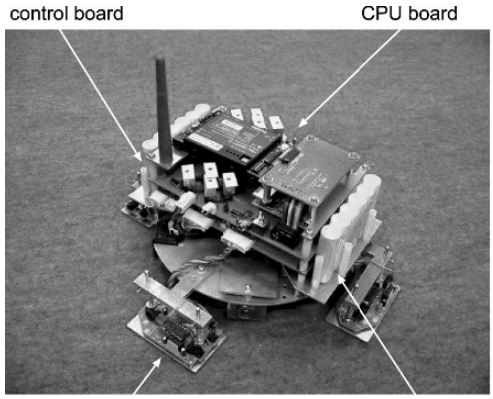
A movement $\boldsymbol{u}$ of the robot is calculated by using the following equation.

$$\boldsymbol{u} = \boldsymbol{B}^{-}\boldsymbol{b} \tag{9}$$

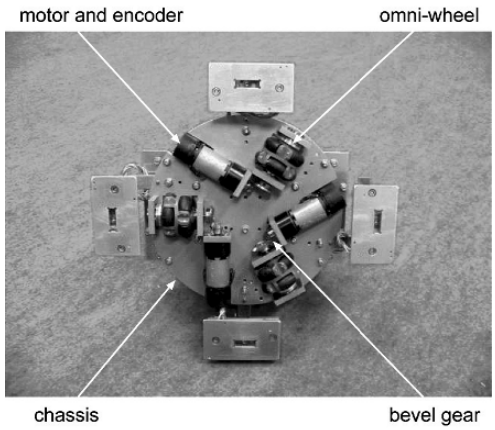
In addition, when two or more sets of optical mouse sensor values with high reliability do not exist, movement of the robot is computed based on wheel rotation.

4 EXPERIMENTS

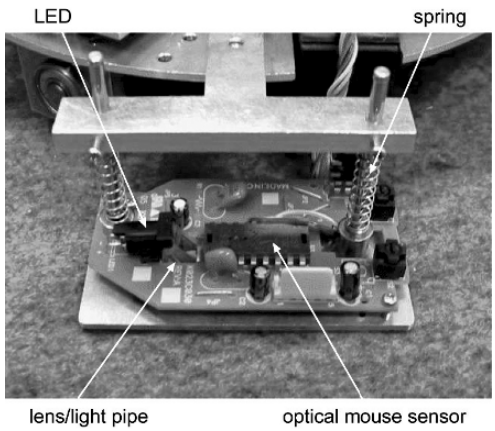
In order to evaluate our methods, experiments were executed using our robot. Firstly, we explain the system configuration of the robot. After that, we show the results of self-localization using dead-reckoning based on optical mouse sensors. Finally, we make one evaluation of our method by reporting on the results of the integration of the global camera information and the dead-reckoning value using the Kalman Filter.



(a) Overall view of mobile robot



(b) Bottom view of mobile robot



(c) Sensor unit

Figure 3: Robot equipped with omni-directional mechanism and optical mouse sensors.

4.1 System Configuration

The robot we have been developing is shown in Figure 3, and its control flow is shown in Figure 4.

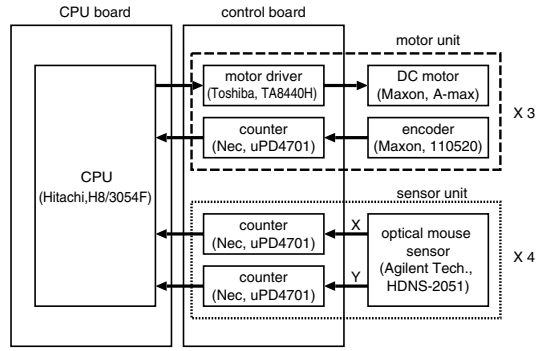


Figure 4: Control flows of the robot.

The robot has an omni-directional mobile mechanism driven by three omni-directional wheels. Four optical mouse sensors are attached around the robot. And in order that an optical mouse sensor may stably scan a floor, the sensor unit is forced onto the floor by springs. Moreover, the CPU board and control board are mounted onto the robot. And they control driving motors and count the pulse from optical mouse sensors. The main specifications of the robot are shown in Table 2.

4.2 Dead-Reckoning Based on Optical Mouse Sensors

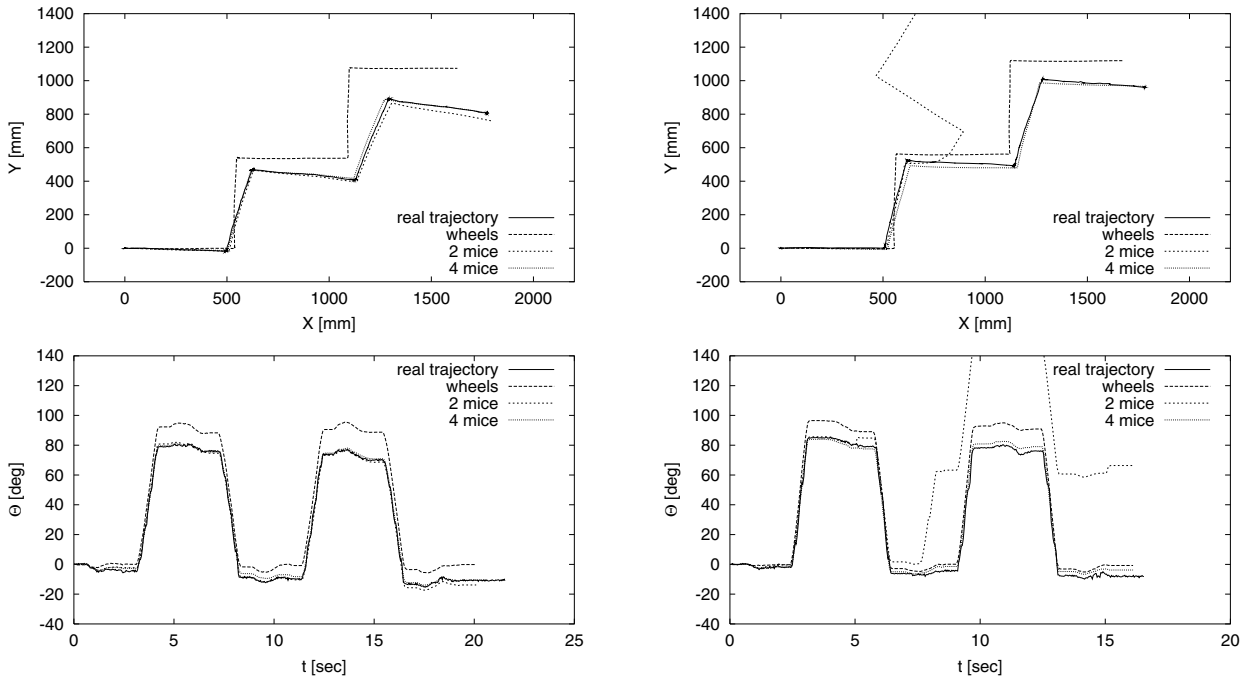
We used two robot speeds: (a) v=300[mm/s], ω=1.82[rad/s] and (b) v=500[mm/s], ω=3.03[rad/s] in the experiments. Speed (a) is slower than the maximum measurement speed of the optical mouse sensor(see Table 1), and speed (b) is faster. We determined speed (b) to be the general maximum speed of an indoor mobile robot. The motion path of the robot is shown in Table 3. The floor is covered with the felt

Table 2: Specifications of the mobile robot.

height	120 [mm]
width	262 [mm]
weight	2 [kg]
max speed	1000 [mm/s]

Table 3: Planned path cartesian coordinates.

	1	**2**	**3**	**4**	**5**
X [mm]	0	500	500	500	500
Y [mm]	0	0	0	500	500
Θ [rad]	0	0	$\pi/2$	$\pi/2$	0
	6	**7**	**8**	**9**	**10**
X [mm]	1000	1000	1000	1000	1500
Y [mm]	500	500	1000	1000	1000
Θ [rad]	0	$\pi/2$	$\pi/2$	0	0



(a) v=300[mm/s], ω=1.82[rad/s]

(b) v=500[mm/s], ω=3.03[rad/s]

Figure 5: Self-localization based on dead-reckoning.

mat used in the RoboCup small size league competitions. Moreover, in order to measure a robot's real trajectory, a camera is installed on the ceiling.

The comparison result of the real trajectory and dead-reckoning values based on wheels, two optical mouse sensors, and four optical mouse sensors are shown in Figure 5. As a result, when a robot speed is (a), even if the dead-reckoning value based on the wheels greatly differs from the real trajectory, two dead-reckoning values based on the optical mouse sensors are mostly in agreement with the real trajectory. On the other hand, when robot speed is (b), the dead-reckoning value based on the wheels differs greatly from the real trajectory as well as in the case of speed (a). The method based on two optical mouse sensors caused erroneous measurements during movement, and a large error has arisen in the dead-reckoning value. On the other hand, the method based on four optical mouse sensors has carried out position estimation with a small error, since a comparison between optical mouse sensors was performed correctly.

Table 4: Errors in 10 dead-reckonings.

(a) v = 300 [mm/s], ω = 1.82 [rad/s]

type	average of the maximum error	
	translation [mm]	orientation [deg]
wheels	191.499	14.970
2 mice	44.516	4.918
4 mice	38.011	4.607

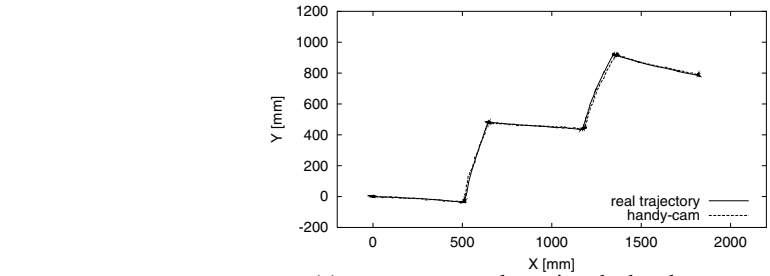
(b) v = 500 [mm/s], ω = 3.03 [rad/s]

type	average of the maximum error	
	translation [mm]	orientation [deg]
wheels	239.397	19.264
2 mice	627.237	40.638
4 mice	61.593	8.588

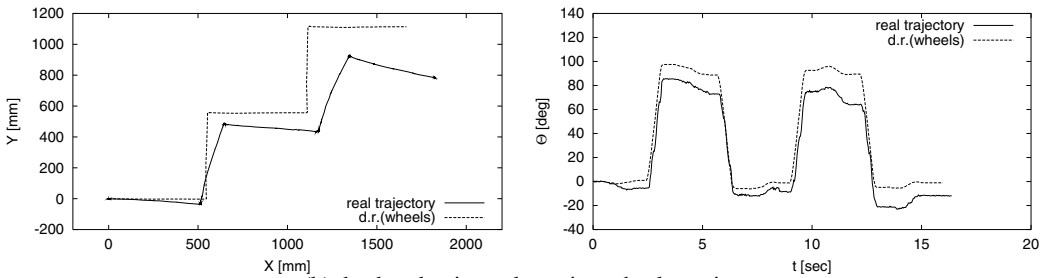
Moreover, we verified the dead-reckoning measurements ten times under the same condition. The average of the maximum error of the estimated value and the measured value in the movement is shown in Table 4. As a result, in the ten dead-reckoning measurements, results similar to the above-mentioned are obtained, and the stability of our method can be confirmed.

4.3 Integration of Global Camera Information and Dead-Reckoning Value

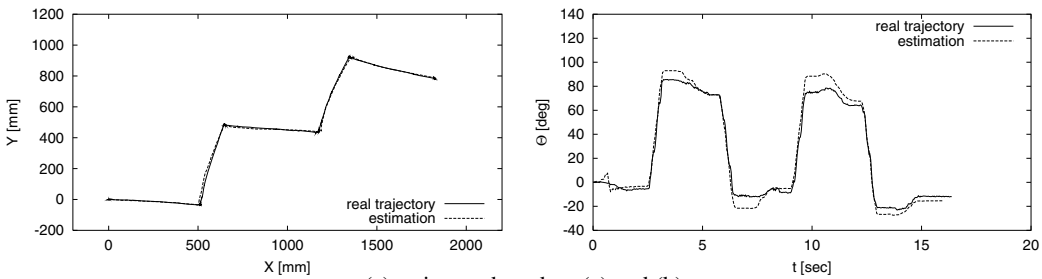
Using another evaluation method, we report on the results of integration of the robot position via global camera and dead-reckoning value using the Kalman Filter. The handy-cam installed in the upper part of



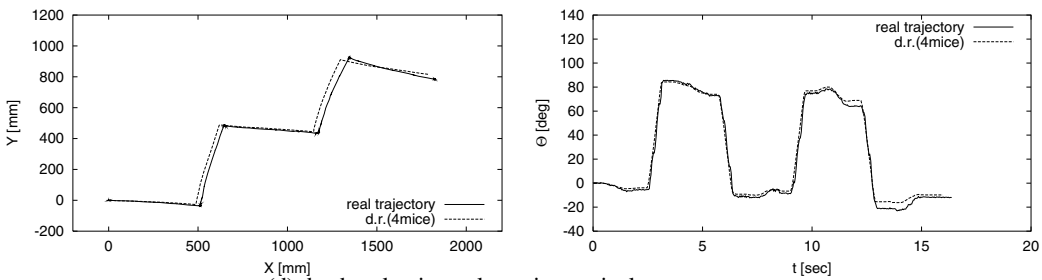
(a) measurement value using the handy-cam



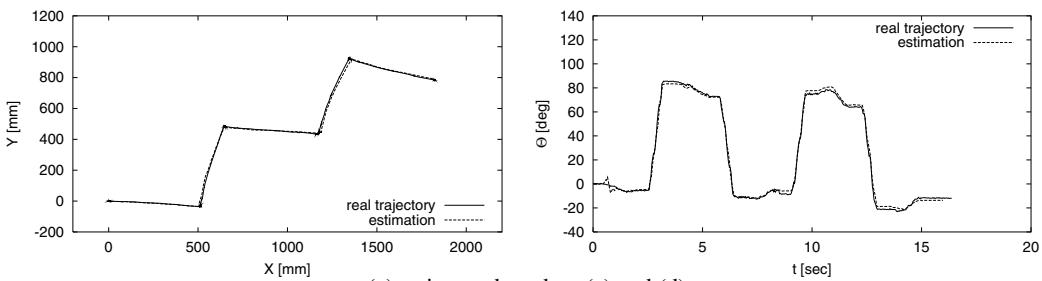
(b) dead-reckoning value using wheel rotation



(c) estimates based on (a) and (b)



(d) dead-reckoning value using optical mouse sensors



(e) estimates based on (a) and (d)

Figure 6: Self-localization based on Integration of global camera information and dead-reckoning value(v=500[mm/s], ω=3.03[rad/s]).

the room is used as the global camera (A separate camera is used for measuring). The global camera measures only the robot position information (orientation information is not included) for the sake of convenience. Though the extended Kalman Filter is used for integrating the two values, its details are omitted. We used v=500[mm/s] and ω=3.03[rad/s] as the robot speed in the experiments.

Figure 6 shows the results of (a) measurement value using the handy-cam, (b) dead-reckoning value using wheel rotation, (c) estimates based on (a) and (b), (d) dead-reckoning value using optical mouse sensors, and (e) estimates based on (a) and (c). As a result, in the case of (c), even if estimates of the position are mostly in agreement with the real trajectory, a large error has arisen in the estimates of orientation. On the other hand, in the case of (e), estimates of both position and orientation are mostly in agreement with the real trajectory. In this experiment, since the orientation is not included in the information from the global camera, the accuracy of the estimates of orientation tends to worsen compared with the estimates of position. However, by using optical mouse sensors, accurate dead-reckoning can be realized, and consequently, not only a position but also an orientation is realizable with sufficient accuracy.

5 CONCLUSION

In this paper, we proposed the method of accurate dead-reckoning by measuring the movement of a robot directly from the floor with optical mouse sensors. By comparing and selecting sensor values from the multiple optical sensors, reliable dead-reckoning was realized. Through several verification checks with the actual robot, we confirmed that our dead-reckoning can be realized accurately and with stability compared with the method based on wheel rotation. In addition, we showed that the accuracy of estimation was greatly improved by using only simple global camera information.

This method of measuring the movement of the robot with optical mouse sensors is limited to the indoor environment. Though the system becomes large scale in an outdoor environment, it is also possible to measure the movement of the robot by taking images of the ground surface with multiple CCD cameras as well as optical mouse sensors. When CCD cameras are used for the measurement, our method can be introduced without the big alterations.

In future work, we will develop one sensor unit including multiple optical sensors, and install this sensor unit in various robots.

REFERENCES

Acroname Inc. Omni-directional poly roller wheel. *http://www.acroname.com/index.html*.

Agilent Technologies Inc. ADNS-2051 optical mouse sensor data sheet. *http://www.agilent.com/semiconductors*.

Chenavier, F. and Crowley, J. L. (1992). Position estimation for a mobile robot using vision and odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'92)*, pages 2588–2593.

Cooney, J. A., Xu, W. L., and Bright, G. (2004). Visual dead-reckoning for motion control of a mecanum-wheeled mobile robot mecanum-wheeled mobile robot. *Mechatronics*, 14:623–637.

Cox, I. J. (1989). Blanche: position estimation for an autonomous robot vehicle. In *Proceedings of the IEEE/RSJ International workshop on Robots and Systems (IROS'89)*, pages 432–439.

Fujimoto, R., Enomoto, M., Sekimori, D., Masutani, Y., and Miyazaki, F. (2002). Dead reckoning for mobile robots using optical mouse sensor (in Japanese). In *Proceedings of the JSME Conference on Robotics and Mechatronics (ROBOMEC'02)*, pages 1A1–G01.

RoboCup official web site. *http://www.robocup.org/*.

Singh, S. P. N. and Waldron, K. J. (2004). Design and evaluation of an integrated planar localization method for desktop robotics. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'04)*, pages 1109–1114.

Tobe, N., Fuchiwaki, O., Misaki, D., Usuda, T., and Aoyama, H. (2004). Precise navigation for piezo based versatile miniature robot with self-gauging detector. In *Proceedings of the 1st International Conference on Positioning Technology (ICPT'04)*.

Watanabe, Y. and Yuta, S. (1990). Position estimation of mobile robots with internal and external sensors using uncertainty evolution technique. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'90)*, pages 2011–2016.

IMAGE BINARISATION USING THE EXTENDED KALMAN FILTER

Alexandra Bartolo[1], Tracey Cassar[1], Kenneth P. Camilleri[1], Simon G. Fabri[2] and Jonathan C. Borg[3]

[1]*Department of Electronic Systems Engineering, University of Malta*
[2]*Department of Electrical Power and Control Engineering, University of Malta*
[3]*Department of Manufacturing Engineering, University of Malta*

[abbart, trcass, kpcami, sgfabr, jjborg]@eng.um.edu.mt

Keywords: Feature Extraction, Image Processing, CAD.

Abstract: Form design is frequently carried out through paper sketches of the designer's mental model of an object. To improve the time it takes from solution concept to production it would therefore be beneficial if paper-based sketches can be automatically interpreted for importation into three-dimensional geometric computer aided design (CAD) systems. This however requires image pre-processing before initiating the automated interpretation of the drawing. This paper proposes a novel application of the Extended Kalman Filter to guide the binarisation process, thus achieving suitable and automatic classification between image foreground and background.

1 INTRODUCTION

Line drawing interpretation systems are used in engineering design as an interface between engineering or architectural drawings and computer-aided design (CAD) tools (Ablameyko and Pridmore, 2000). Using similar principles, sketch recognition systems are being developed since it is acknowledged that designers can express their ideas more naturally by means of sketches (Roth-Koch, 2000). Recent developments of such recognition systems have focused on online sketches obtained by means of PDAs or tablet PCs. Since such systems are online, the interpretation system has additional information about the drawing, for example, pen position and velocity. However, these systems lack the portability and flexibility of paper (Farrugia et al., 2004). In order to achieve this flexibility, the images must be pre-processed, such that line data may be extracted from the static image. Binarisation is one such process, which compensates for the noise introduced by the digitizing system. This work proposes the use of the Extended Kalman Filter (EKF) to guide the binarisation of images as a step towards the automation of the sketch recognition process which provides the necessary data to control rapid prototyping and manufacturing equipment. The proposed method improves the binarisation of poor quality images, whilst reducing the complexity of the threshold selection process. This paper is divided as follows: Section 2 gives a brief review of binarisation techniques, Section 3 introduces the EKF and illustrates how this filter may be used to identify a suitable threshold for binarisation. This is followed by the results in Section 4 and conclusions in Section 5.

2 BINARISATION TECHNIQUES

Binarisation is the process by which grey levels within an image are classified as either foreground or background (Ablameyko and Pridmore, 2000). The selection of a suitable binarisation technique is dependent on the type and quality of the images being used (Bulen and Mehmet, 2004). Since the sketched line drawing interpretation system that is being developed is expected to process images from the field, it will typically process poor quality images, digitized using low-resolution devices, such as a camera-phone. This requires a more detailed and local analysis of the pixel distributions in order to select a suitable threshold for pixel classification. This section describes five established binarisation techniques originally proposed for line drawings and text images.

Palumbo and Guliano (Yang Y. and Yan H., 2000) use a fixed 9×9 window to evaluate the class of each pixel within the image. The pixel value is determined according to the five 3×3 local pixels within the 9×9 window centered on the pixel in consideration.

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 153–162.

An initial user-defined threshold is used to determine the pixels which definitely belong to the background whilst the remaining pixels are classified using a different label assignment rule requiring the specification of three additional user defined parameters. Determining the values of the user-defined parameters is not straightforward since they cannot be deduced from the image properties.

In Niblack's algorithm (Bulen and Mehmet, 2004), the user does not need to define the classification threshold since this is evaluated according to the mean and standard deviation of each pixel's neighbours. However, the user is required to specify the size of a window W from which the pixel's neighbours are taken. As with Palumbo and Guliano's method, the selected threshold is applied to each individual pixel in the image. Thus, the classification of each pixel $w(x, y)$ may be modelled as follows:

$$w(x,y) \rightarrow \begin{cases} w_f & , \quad p(x,y) < T(x,y) \\ w_b & , \quad p(x,y) \geq T(x,y) \end{cases} \qquad (1)$$

where $T(x, y) = \mu(x, y) + k \times \sigma(x, y)$, $\mu(x, y)$ is the mean grey level of the pixels within the window, $\sigma(x, y)$ is the standard deviation of these pixels and k is a user defined parameter. Therefore, Niblack's algorithm requires two user defined parameters, namely the window size W and k. The window size determines the number of pixels from which the mean $\mu(x, y)$ and standard deviation $\sigma(x, y)$ are evaluated, and so it should reflect the quality of the image background on which prior knowledge is unavailable. The value of k is used to adjust the amount of print object boundary that is taken as part of the foreground, and is therefore dependent on the quality of the drawn line which is also an unknown quantity.

Eikvil's method (Trier O. D. and Jain A. K., 2000) is based on the established global binarisation technique developed by Otsu (Gonzalez R. and Woods R. E., 2002). As with Palumbo and Guliano, Eikvil's classification requires the specification of two window sizes. However, in this case the two windows W_L and W_S of size L and S respectively, are concentric, with W_L being the larger window. The pixels within W_L are temporarily classified into two clusters by Otsu's threshold. The mean of these two clusters is evaluated and their difference is compared to a parameter k which determines whether there is sufficient contrast between the two clusters. This indicates the effectiveness of Otsu's threshold on the selected region. Thus, if the difference between the two means is larger than k, the pixels within the smaller window W_S are classified according to Otsu's threshold. Otherwise, the pixels are assigned to the class whose label is closest to the mean grey level within the smaller window W_S. Thus, unlike the previous two methods, this algorithm does not classify single pixels, but the group of pixels located in the smaller window W_S. This method requires the specification of three user defined parameters, of which S and L define window sizes, whilst k determines the thresholding method applied to the smaller window. The size of the smaller window W_S may be set to 3 which defines the smallest window centered on a pixel. However, the remaining parameters must be specified according to the particular image properties.

Kamel and Zhao's logical adaptive technique (Kamel M. and Zhao A., 1993) compares the grey level of the pixel in consideration with eight local averages in a pixel neighbourhood of size $(2SW + 1) \times (2SW + 1)$ where SW represents the stroke width of the line drawing. A comparison operator is derived from these averages and is used to determine the class of the pixel in consideration. The algorithm requires two user defined parameters, namely the stroke width SW and an initial threshold T which is used to evaluate the required comparison operator. Yang and Yan (Yang Y. and Yan H., 2000) proposed a method by which the two parameters SW and T are calculated adaptively. However, the adaptive evaluation of the parameter T requires another parameter α. Yang and Yan (Yang Y. and Yan H., 2000) specify a range of values of α for which suitable values of T may be obtained.

Brensen's method (Bulen and Mehmet, 2004) may either classify a single pixel or a group of pixels simultaneously according to the contrast present within a selected window. The window's contrast is defined as $C(x, y) = Z_{max} - Z_{min}$, where Z_{max} and Z_{min} are the maximum and minimum grey levels within the window. If this contrast is smaller than a predefined value k, the pixels within the window belong to the same class, and the entire window may be assigned to a single class. However, if the contrast C is sufficiently large, then the pixels within that window belong to two different classes. Since the window has high contrast, a simple threshold based on the average grey level may be used to classify the pixels within this window. Thus, the threshold T is defined as $T(x, y) = \frac{1}{2} \times (Z_{max} + Z_{min})$. This method requires the specification of parameter k which may be evaluated adaptively using the method proposed in (Bartolo A. et al., 2004)

2.1 Drawbacks

Although the above methods may yield results of considerably good quality, the classification process requires that a classifying criterion is evaluated for each pixel in the image. Furthermore, these algorithms require the specification of some parameter, such as a window size in order to evaluate the threshold. Although suggested values are specified for some algorithms, better results are obtained after fine-tuning the parameter to the characteristics of the image under

test. Thus the performance of these methods is susceptible to image conditions. Methods for the adaptive evaluation for Brensen's and Kamel & Zhao's methods have been proposed, but these require considerable computational times, which slow down the product prototyping process. In this paper, we attempt to overcome these problems by modelling the sketch as a trajectory being tracked in time.

3 LINE TRACKING

A sketched drawing may be considered as a number of lines which interact at junctions or corners, from which two or three dimensional shapes may be perceived. These lines may be considered as distinct entities, which can be described independently by some mathematical model. In this paper, two mathematical models which describe the position of a point on a line and its intensity are used to enable line tracking. Each line stroke is modelled as a trajectory propagating with a velocity v along a line subtending an angle θ with the horizontal axis, thus the position of the trajectory at a time instant $k+1$ is given by Eq. (2), where x_1 and x_2 are the vertical and horizontal coordinates on a plane.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + v \begin{bmatrix} \sin\theta \\ \cos\theta \end{bmatrix} \quad (2)$$

Given a static, offline image, the velocity of propagation is irrelevant and may be assumed to be unity, thus making the (x_1, x_2) coordinates of the line dependent only on the direction θ. The bilinear interpolation relationship given by Eq. (3) (Gonzalez R. and Woods R. E., 2002), can be used to describe the intensity of a pixel in terms of its (x_1, x_2) coordinates,

$$z(k) = Ax_1(k) + Bx_2(k) + Cx_1(k)x_2(k) + D \quad (3)$$

where A, B, C and D are interpolation coefficients derived from the four neighbours of a pixel in the image with coordinates (x_1, x_2).

In this way, each stroke in a given drawing may be considered as a process modelled by Eq. (2). From the original image, measurements of pixel intensities, given by Eq. (3), can be obtained. By considering the (x_1, x_2) coordinates of each line as the states of a dynamic system and the intensity values to be measurements obtained from noisy sensors, Kalman filter theory (Maybeck P. S. , 1982) may be used to estimate the system states and hence the coordinates of points on each line.

Since the relation between pixel intensity and pixel position given by Eq. (3) is not linear, the Extended Kalman Filter (EKF) was adopted. The EKF linearizes the state estimation around the current estimate by using the partial derivatives of the process and measurement functions to compute estimates even when non-linearities are present. The EKF essentially assumes that the process is linear around the current state (Maybeck P. S. , 1982).

3.1 The Extended Kalman Filter

In general, the EKF addresses the problem of estimating the state $\mathbf{x}$ of a discrete-time process modelled by nonlinear state-space equations of the general form:

$$\mathbf{x}(k+1) = \mathbf{f}(k, \mathbf{u}(k), \mathbf{x}(k)) + \mathbf{w}(k) \quad (4)$$
$$\mathbf{z}(k) = \mathbf{h}(k,\ \mathbf{x}(k)) + \mathbf{v}(k) \quad (5)$$

where $\mathbf{w}$ represents the model noise and $\mathbf{v}$ the measurement noise, which are zero-mean, Gaussian noise sequences of covariance Q and R respectively, $\mathbf{u}$ is a known input, $\mathbf{z}$ is a measured output and $\mathbf{f}$ and $\mathbf{h}$ are general non-linear functions. In our line sketching application, the line generator model given by Eq. (2) is cast into the state space form of Eq. (4) to give Eq. (6). Similarly, Eq. (3) which represents the intensity model, is cast into the state space form of Eq. (5) to give Eq. (7).

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \theta(k) + \mathbf{w}(k) \quad (6)$$
$$\mathbf{z}(k) = \mathbf{h}(k,\ \mathbf{x}(k)) + \mathbf{v}(k) \quad (7)$$

where the state vector

$$\mathbf{x} \equiv \left[x_{11}, x_{12}, \cdots, x_{n1}, x_{n2}\right]^T$$

represents pen positions on n line strokes in the image,

$$\theta(k) \equiv \left[\sin(\theta_1), \cos(\theta_1), \cdots, \sin(\theta_n), \cos(\theta_n)\right]^T$$

represents the orientation of the lines. The intensity of the trajectories at time k is given by $\mathbf{z}$, which may be written as:

$$\begin{aligned} z_1(k) &= A_1x_{11}(k) + B_1x_{12}(k) + \\ &\quad C_1x_{11}(k)x_{12}(k) + D_1 + v_1(k) \\ z_2(k) &= A_2x_{21}(k) + B_2x_{22}(k) + \\ &\quad C_2x_{21}(k)x_{22}(k) + D_2 + v_2(k) \\ &\vdots \\ z_n(k) &= A_nx_{n1}(k) + B_nx_{n2}(k) + \\ &\quad C_nx_{n1}(k)x_{n2}(k) + D_n + v_n(k) \end{aligned}$$

where A_i, B_i, C_i and D_i are known constants obtained from the bilinear interpolation, given by Eq. (3).

Comparison between Eqs. (4) and (6) gives $\mathbf{f}(k, \mathbf{u}(k), \mathbf{x}(k)) = \mathbf{x}(k) + \theta(k)$, and $\mathbf{h}(k, \mathbf{x}(k))$ is

given by Eq. (3). Process noise $\mathbf{w}(k)$ models any deviations from the ideal line stroke, and measurement noise $\mathbf{v}(k)$ corresponds to noise affecting the intensity measurements of the image.

Based on the state space model given by Eqs. (6) and (7), the EKF algorithm is used to find an estimate $\hat{\mathbf{x}}(k|k)$ to the actual state $\mathbf{x}(k)$, based upon intensity measurements $\mathbf{z}(k)$ as follows (Maybeck P. S. , 1982): The Kalman Gain is defined as:

$$\mathbf{K}(k) = \frac{\mathbf{P}(k|k-1)\nabla^T_{\mathbf{h}(\hat{\mathbf{x}}(k|k-1))}}{\nabla_{\mathbf{h}(\hat{\mathbf{x}}(k|k-1))}\mathbf{P}(k|k-1)\nabla^T_{\mathbf{h}(\hat{\mathbf{x}}(k|k-1))} + R} \tag{8}$$

where

$$\nabla_{\mathbf{h}(\mathbf{x})} \equiv \begin{pmatrix} \frac{\partial \mathbf{h}_1}{\partial x_{11}} & \frac{\partial \mathbf{h}_1}{\partial x_{12}} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{\partial \mathbf{h}_n}{\partial x_{n1}} & \frac{\partial \mathbf{h}_n}{\partial x_{n2}} \end{pmatrix}$$

represents the rate of change in intensity in the vertical and horizontal directions for each line stroke.
The state estimate is obtained by:

$$\begin{aligned} \hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \\ \mathbf{K}(k)\big[\mathbf{z}(k) - \mathbf{h}(k, \hat{\mathbf{x}}(k|k-1))\big] \end{aligned} \tag{9}$$

where $\mathbf{z}(k)$ represents the true intensity measured from the image and $\mathbf{h}(k, \hat{\mathbf{x}}(k|k-1))$ is given by Eq. (3) evaluated at $\mathbf{x}(k) = \hat{\mathbf{x}}(k|k-1)$. The error Covariance $\mathbf{P}(k|k)$ is given by:

$$\mathbf{P}(k|k) = [\mathbf{I} - \mathbf{K}(k)\nabla_{\mathbf{h}(\hat{\mathbf{x}}(k|k-1))}]\mathbf{P}(k|k-1) \tag{10}$$

where the covariance prediction is:

$$\mathbf{P}(k+1|k) = J_{\mathbf{f}(\hat{\mathbf{x}}(k|k))}P(k|k)J^T_{\mathbf{f}(\hat{\mathbf{x}}(k|k))} + Q \tag{11}$$

where $J_{\mathbf{f}(\mathbf{x})}$ is the Jacobian matrix $\left[\frac{\partial f_i}{\partial x_j}\right]$, which in this case is equivalent to the identity matrix from Eq. (6). The state estimate prediction at time $k+1$ is:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{f}(k, \mathbf{u}(k), \hat{\mathbf{x}}(k|k)) \tag{12}$$

The EKF Eqs. (8) to (12) will recursively compute the state estimate $\hat{\mathbf{x}}(k|k)$ for each iterate k, given initial estimates $\hat{\mathbf{x}}(0|-1) = \mathbf{x}_0$ and initial covariance $\mathbf{P}(0|-1) = \mathbf{P}_0$ which represents the initial covariance of the error, reflecting the initial uncertainty of the estimates.

3.2 Application to Binarisation

The EKF equations described above, give the possibility of locating points on a number of lines that form part of the drawn object or objects in an image. In this way, the EKF helps to discriminate between the image foreground and background by selecting those pixels which are located on lines and are therefore part of the image foreground. The intensity of the pixels along the tracked trajectory provides information about the grey level intensities of the pixels forming part of the line drawing. The mean grey level intensity μ_t, and standard deviation σ_t of the tracked pixels can therefore be used to approximate the grey-level intensity of the sketched object, and hence guide the binarisation process. Since line pixels are darker than the background, the image pixels are classified by comparison to a threshold $T = \mu_t + n \times \sigma_t$,where n is a constant which defines the tolerance to grey-level variations along the tracked line. Pixels whose intensities are less than T may be classified as foreground line pixels whilst the remaining pixels may be classified as background pixels.

3.3 Implementation of the EKF

The implementation of the EKF requires suitable starting points to initialize the line tracking process. The position of the lines in a static image are unknown, however, it may be assumed that part of the image will be located towards the center of the image. Thus, two scans along the horizontal and vertical centerlines of the image are performed. The derivative of the grey level intensities of pixels lying along this line is considered as shown in Figure 1. The presence of a line is indicated by a negative to positive peak transition along the tracking direction which correspond to the background-foreground and foreground-background transitions associated with the edges of the line stroke. Thus, the zero-crossing between two such peaks may be considered as a suitable starting for line tracking. Since a horizontal and a vertical scan are carried out, the EKF can be initialized with at least two starting points.

Since sketches are not necessarily built from straight lines, but may have curves or lines that exhibit a change in orientation, the sketched strokes are

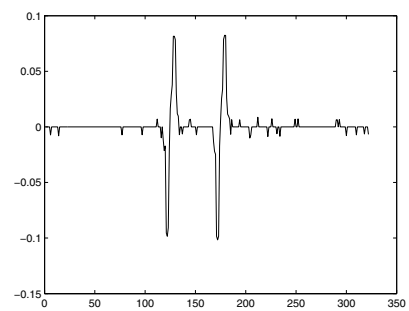


Figure 1: The derivative of a horizontal row in an image. The two $-ve$ to $+ve$ peak transitions indicate the presence of two line strokes.

modelled as piece-wise linear segments. This implies that the line stroke may be built from a number of short straight line segments which may be represented by the state space model shown in Eq. (6), thus requiring the evaluation of the line orientation $\theta(k)$ at each iteration k. This may be obtained by using Sobel edge response (Gonzalez R. and Woods R. E., 2002), which gives the magnitude response and the orientation of the pixels within the image. For edge pixels, the magnitude response is highest and the pixel orientation corresponds to the orientation of the line. Thus, for each state $\hat{\mathbf{x}}$, the closest edge pixel pair are located and their orientation is used as an approximation for the line direction $\theta(k)$.

The state estimates given by Eqs. (9) require the evaluation of the intensity of the image at instant k. This may be obtained by searching for the darkest grey level within a distance of one unit from the current pixel position in the direction of $\theta + \Delta\theta$ where the $\Delta\theta$ term is used to allow for deviations in the line direction. Since the line strokes are expected to be smooth, any line deviation from θ is accommodated by seeking the darkest pixels in the cardinal directions that enclose θ.

Since the EKF assumes that the model used is linear around the current state, large deviations from the line stroke would cause the filter to diverge. For this reason, it is required to terminate the tracking before the filter diverges. An indication that the filter is diverging may be obtained by comparing the grey level intensity of the tracked point with the grey level intensity of background pixels. A measure of the background intensity may be obtained by taking the mean grey-level m_s of a sample of pixels located at a distance d perpendicular to the line direction, where d should be greater than the stroke width of the sketched lines. A point located on an image line will have a grey level m_p which is less than the mean grey level m_s of the sample pixels. Thus, divergence is indicated when the mean grey level m_s of the sampled pixels is less than or equal to the grey level intensity m_p of the tracked point thus indicating that the point is no longer on the sketched line and has moved to a background region. This criterion also detects when the tracking point arrives at the end of a line and is therefore also used as a criterion to terminate the tracking process once the end of the line is reached.

The image digitization process introduces some degree of noise to the image such that adjacent pixels will have variations in their grey level intensity even though they belong to the same class. This will introduce errors in the evaluation of the EKF starting points since the derivative of an image row or column will also reflect these intensity variations. For this reason the image is low pass filtered using a 3×3 mean filter (Gonzalez R. and Woods R. E., 2002). This will reduce the effect of the grey level variations in adjacent pixels making the transition between image class more prominent in the row and column derivatives.

4 TESTING AND RESULTS

The proposed algorithm was tested under various conditions and compared to the performance of the other established methods discussed in Section 2. This section shows the results obtained for a number of sample grey level images, whose grey levels are in the range [0, 256]. The visual results obtained by Kamel and Zhao's method are also shown in order to allow a visual comparison. Kamel and Zhao's method has been chosen as this has the lowest number of user defined parameters and thus offers the highest degree of automation.

The ability of the EKF to track multiple lines was first tested using images that exhibited a low noise component such as that shown in Figure 2. This shows an example where the EKF tracks eleven segments, corresponding to seven starting points detected from a horizontal scan and four segments detected from a vertical scan of the image.

The algorithm was then tested under varying degrees of measurement noise. This was introduced by adding zero-mean Gaussian noise to the image. Figure 3 illustrates a ground truth image in which 5.7% of the pixels are foreground pixels, whilst the remaining 94.3% are background pixels. The corresponding noisy image is shown in Figure 3(b). The noise added has a standard-deviation of 36 grey-levels, which in this case corresponds to a signal-to-noise ratio (SNR) of 12.8dB. The results obtained by the EKF algorithm and Kamel and Zhao's algorithm are shown in Figure 3(c) and 3(d) respectively. Further results are given in Table 1. These show that the results obtained

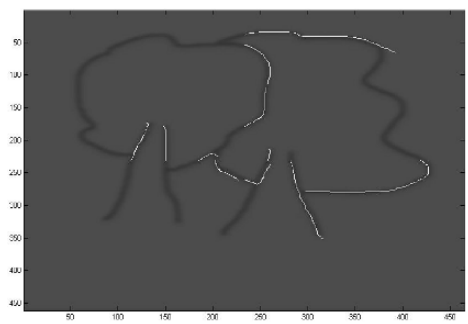
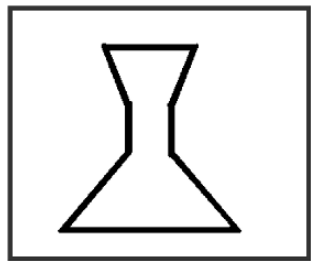
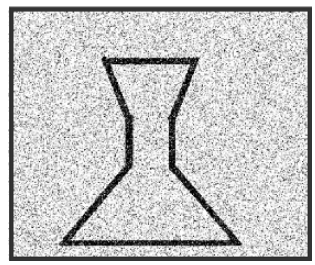


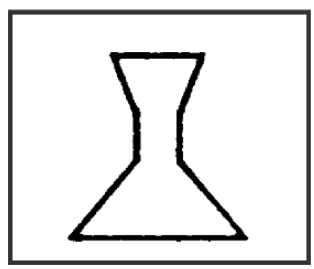
Figure 2: Illustrating the tracking paths generated by the EKF. White line segments indicate the tracked path, whilst the darker lines indicate the image line strokes. This example shows the tracking of 11 segments.



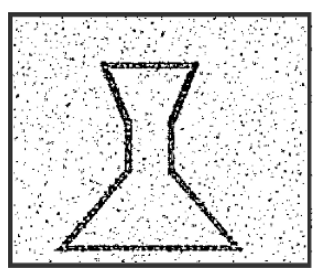
(a)



(b)



(c)



(d)

Figure 3: Illustrating the performance of the EKF binarisation under noise. Figure (a) shows a ground truth image and Figure (b) shows the corresponding image with added Gaussian noise resulting in an SNR of 15.5dB. Figures (c) and (d) show the binary result obtained by the EKF method and Kamel & Zhao's method respectively.

Table 1: Comparison of percentage pixel error of the EKF algorithm and Kamel and Zhao's algroithm under different noise conditions for the image shown in Figure 3(a). f→b indicates the percentage foreground misclassification and b→f the percentage background misclassification.

	% pixel error			
	EKF		Kamel & Zhao	
SNR(dB)	f → b	b → f	f → b	b → f
18.08	7.1	0.53	17.9	0.79
16.20	14	0.07	26	1.19
14.54	12.2	0.18	35.8	1.59
12.79	2.9	0.97	44.7	1.9

by the EKF algorithm have lower foreground misclassifications and background misclassifications in comparison to the results obtained by Kamel and Zhao's method. This indicates that the EKF algorithm gives a better performance than that of Kamel and Zhao under noisy conditions.

Figure 4 illustrates three images used to test the algorithm. These include multiple lines and curves, which illustrate that the line tracking process may effectively track such images. Four lines were tracked from Figure 4(a–i), whilst seven lines were tracked in Figure 4(a–ii) and Figure 4(a–iii). Using this tracking procedure, the number of sampled pixels as a ratio of the total number of pixels is 24%, 36% and 27% respectively. This indicates that the thresholding decision is based on a small number of pixels, which however, correspond to pixels directly related to the image foreground. The binary result obtained for these images is illustrated in Figure 4(b i–iii). The results obtained may be compared with those illustrated in Figure 4(c i–iii), which are obtained by using Kamel and Zhao's algorithm after *manually* determining the most suitable values for α. These results show that the proposed EKF algorithm gives results whose quality is comparable to those given by other binarisation techniques. Furthermore, Table 2 shows

Table 2: Comparison of computational times with the algorithms discussed in Section 2 for the images shown in Figure 4.

	Computational Time (s)		
Image	4(a)	4(b)	4(c)
EKF	57.3	26.3	55.9
Eikvil	75.2	73.7	75.4
Brensen	160	139.5	137.9
Kamel	78.6	78.8	70.9
Niblack	30.7	30.5	39.8
Palumbo	12.5	9.7	13

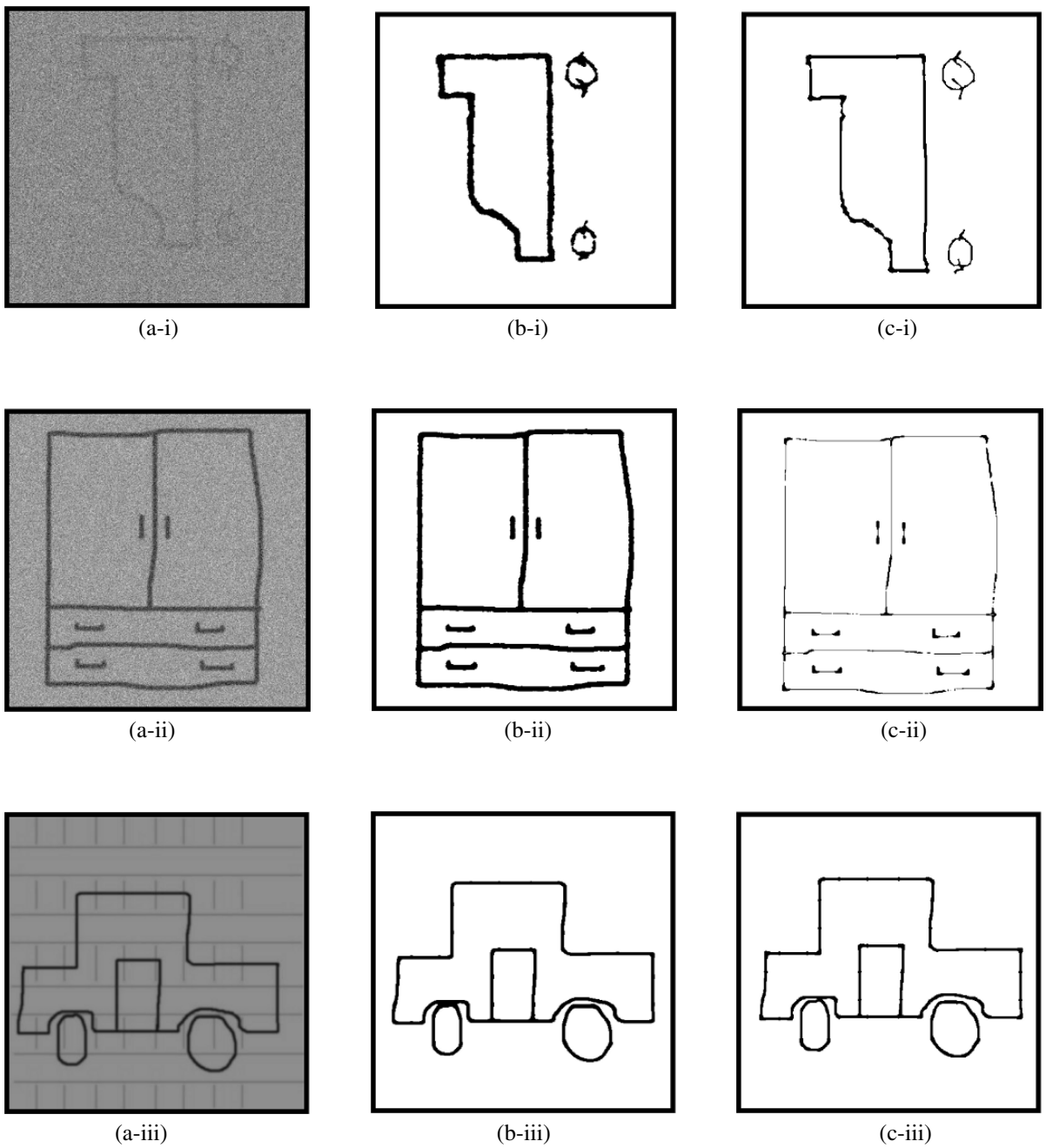
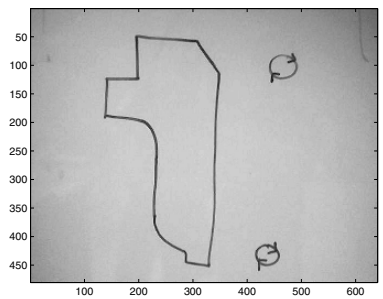
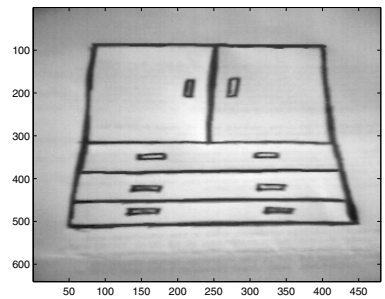


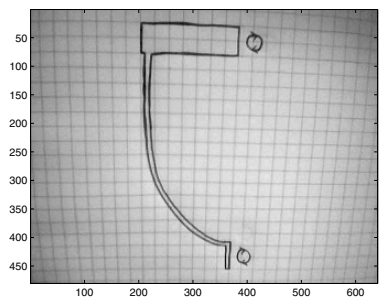
Figure 4: Sample test images. Images (a i) and (a ii) illustrate images that have a homogenous background. The dynamic range of these images is 80 and 160 respectively. Image (a iii) is an example of an image having background artefacts. Images (b i - iii) are the results obtained after binarising the image using the proposed algorithm. These can be compared with Images (c i - iii) which show the results obtained by Kamel and Zhao's algorithm after *manually* setting the value of α to 0.2, 0.4 and 0.1 respectively.



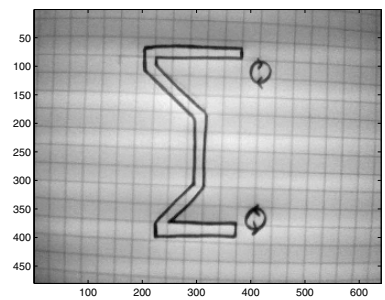
(a)



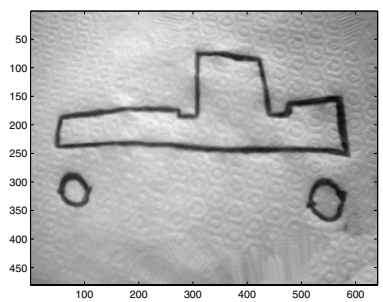
(b)



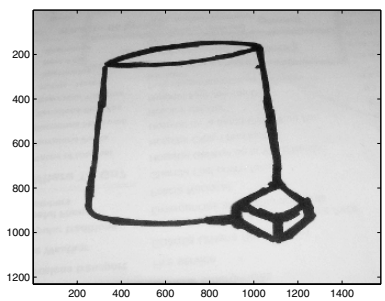
(c)



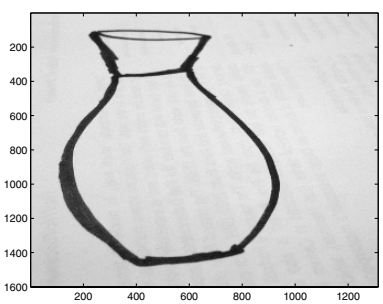
(d)



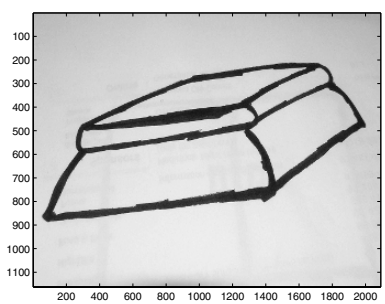
(e)



(f)



(g)



(h)

Figure 5: A sample of the images used to test the EKF algorithm. Figures (a–e) were captured at a resolution of 96dpi using a cameraphone, whilst Figures (f–h) were captured with a digital camera. Figures (a) and (b) are examples of images drawn on plain, white background, Figures (c–e) show images drawn on textured background whilst Figure (f–h) are examples of images with variable line strokes.

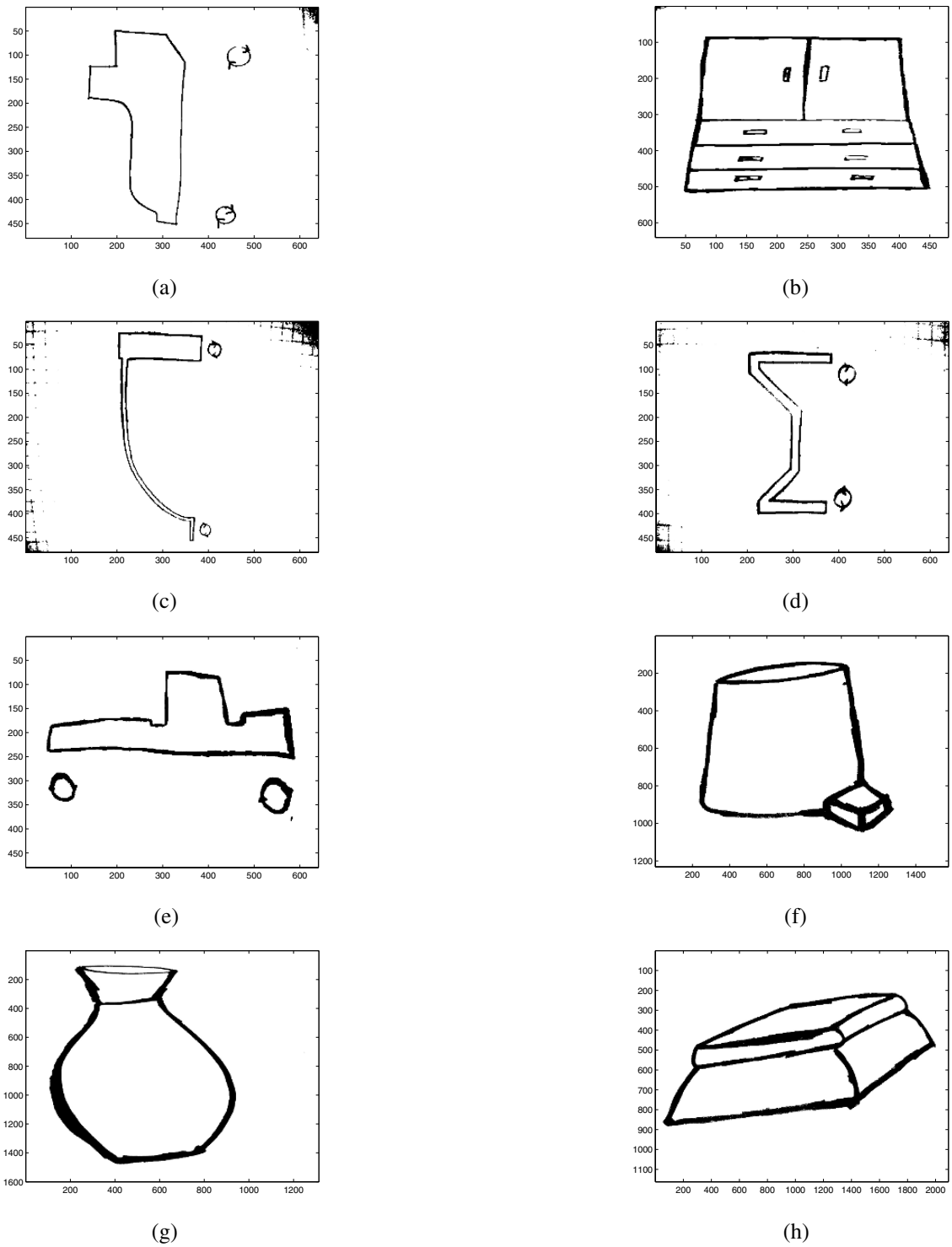


Figure 6: Results obtained by the EKF binarisation algorithm for the test images shown in Figure 5.

that the proposed binarisation process requires lower computational times than Brensen's and Kamel & Zhao's methods, for which adaptive parameter evaluation was applied. Although Niblack's, and Palombo and Guiliano's methods show lower computational times, this does not include the time required to find suitable parameters for each image, because, these are not adaptive methods.

The algorithm was also tested on images captured by a cameraphone at a resolution of 96 dpi. A sample of these images is shown in Figure 5(a–e) where Figure 5(a) and (b) show images drawn on plain white paper, Figure 5(c) and (d) show images drawn on graph paper, whilst Figure 5(e) shows an image drawn on textured tissue paper. The images shown in Figures 5(f–h) were captured using a higher resolution digital camera. These images were drawn on low quality paper and are examples of images which have variable line strokes. Note that the cameraphone captures the light reflections on the image such that the resulting digital images display variable grey-level intensities along the background. This is particularly evident in the image corners of Figures 5(a–e), where the grey-level intensity of the background is comparable to that of the foreground. The performance of the EKF algorithm is comparable to that of the algorithms discussed in Section 2 but the proposed EKF algorithm requires smaller computational times. Figure 6 shows how the algorithm correctly identifies the foreground pixels. Of particular importance is the distinction made between the image foreground and the textured backgrounds in Figures 6(c–e), which allows the extraction of the object from interfering backgrounds. Some misclassification of the background pixels in the corner regions occurs for images in Figure 6(a), (c) and (d). This is due to the fact that the image is being thresholded with a global threshold. In these images, the EKF tracked pixels which, although still part of the foreground, had a relatively high grey level intensity. This results in a higher valued threshold, which will classify the darker background regions as foreground. Correct classification is also obtained for images having variable stroke widths, which shows that the proposed algorithm is independent of stroke width. This contrasts with other algorithms, where the user is required to specify a window size that balances the thin and thick stroke widths.

5 CONCLUSION

The proposed EKF binarization method has been shown to yield good quality results that are comparable to those obtained by existing methods. Further work is being carried out in order to apply the algorithm to local image regions, resulting in a number of local thresholds rather than a single global threshold. This will reduce the effect of global thresholding, thus further improving the results obtained.

The proposed method offers a higher degree of automation than the other binarization techniques discussed since no user defined parameters are required. This helps to improve the rapid prototyping process of the sketched line drawing, which is the main, long term objective of this work.

ACKNOWLEDGEMENTS

This work has been mainly supported by Grant 73604 of the University of Malta.

REFERENCES

Ablameyko S. and Pridmore T. (2000). *Machine Interpretation of Line Drawing Images*. Springer-Verlag.

Bartolo A., Camilleri K., Borg J., and Farrugia P. (2004). Adaptation of Brensen's Thresholding Algorithm for Sketched Line Drawings. *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pages 81–90.

Bulen S. and Mehmet S. (2004). Image Thresholding Techniques - A Survey over Categories. *Journal of Electronic Imaging*, 13:146–165.

Farrugia P., Borg J., Camilleri K., Spiteri C., and Bartolo A. (2004). A Cameraphone-Based Approach for the Generation of 3D Models from Paper Sketches. *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pages 34–42.

Gonzalez R. and Woods R. E. (2002). *Digital Image Processing*. Prentice Hall.

Kamel M. and Zhao A. (1993). Extraction of Binary Character/Graphics from Grey Level Document Images. *CVGIP: Graphical Models and Image Processing*, 55(3):203–217.

Maybeck P. S. (1982). *Stochastic Models, Estimation and Control, Volume 2*. Academic Press.

Roth-Koch S. (2000). Generating CAD Models from Sketches. *Proceedings of the IFIP WG5.2 Geometric Modeling: Fundamentals and Applications*, pages 207–219.

Trier O. D. and Jain A. K. (2000). Goal directed evaluation of binarisation methods. *Workshop on Perfromance versus Methodology in Computer Vision*, 17(3):209–217.

Yang Y. and Yan H. (2000). An Adaptive Logical Method for Binarisation of Degraded Document Images. *Journal of the Pattern Recognition Society*, 33:787–807.

LOWER LIMB PROSTHESIS: FINAL PROTOTYPE RELEASE AND CONTROL SETTING METHODOLOGIES

Vicentini Federico, Canina Marita, Rovetta Alberto
Robotics Laboratory, Mechanical Department - Politecnico di Milano – Itlay
Via Bonardi 9, 20133 Milano - Italy
{federico.vicentini,marita.canina,alberto.rovetta}@polimi.it

Keywords: Lower limb prosthesis, biorobotics, human-machine interface, control, step analysis.

Abstract: The current research activity on prostheses project at the Robotics Laboratory (Mechanics Department, Politecnico di Milano) is carried on in cooperation with Centro Protesi INAIL and STMicroelectronics. The team is both innovative and interesting, owing to the fact that it not only involves a range of specialists but also gives rise to interdisciplinary aspects. They are absolutely essential in project dealing with such complex issues. This Mechanic-Leg project, called Hermes, is an original solution in the field of prosthesis. Main aim of this research is the prototyping of a new kind of mechanical lower limb with an electronic control. The device, resorting to innovatory mechanical and electronic solutions, allows the controller to modify the type of step, passing from a slow to a fast walk, in an easy and intuitive way, taking care of patient's requirements. The Hermes M-Leg cost is comparable to the actual commercial non electronic controlled artificial knees. The distinguishing features of Hermes M-Leg project are an higher awareness in innovative aspects related to medical/biological/engineering research. Then, a pervasive use of cutting-edge technology (electronics, IT, material-related technologies, etc.). The controller architecture is built upon a low memory processing features. The hard analysis and test activity help to model the algorithm for step control. The adaptive behaviour is mostly due to an effective experience in testing and software tuning in cooperation with patients and clinical staff.

1 INTRODUCTION

A *prosthetic system,* totally replacing a lost human body part (thereby ensuring the functionality of a specific physiological system), should act as a fully interconnected part which the person is able to interact with. The prosthesis designer takes into particular account the man-device interface. This is done to satisfy the patient-driven requirements and to project a suitable prosthesis. According to the assessment, the design concept play a major role in evaluating the prosthetic system.

The prostheses technological evolution has begun in sixties. The availability of advanced technologies coming from the automotive and aerospace industry, allowed to develop more comfortable, resistant and light materials. The prosthesis weight limitation is mandatory to reduce the user tiring out and to allow a longer use during the day. In the 80s, new materials were introduced provided with similar mechanical resistance but lower density compared to previous ones. The miniaturization of components is actually fundamental in the design system; it allows to reduce the prosthesis overall weight. So that many global requirements have arisen from technology development to user comfort. First, an electronic controlled prosthesis must give stability to the patient, support his/her weight and make his/her movements easy. Therefore, it is very important to minimize energy supplied by the patient and to fit a natural limb behaviour. Finally, the device should be adaptive and self learning.

This paper is addressed to the methodology in device optimization from several points of view. The final release prototype developed mixes up design, mechanics and software issues due to long experience in prosthesis field. An accurate analysis is carried on about all design process, and it will be presented as well as a short description of produced device.

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 163–173.

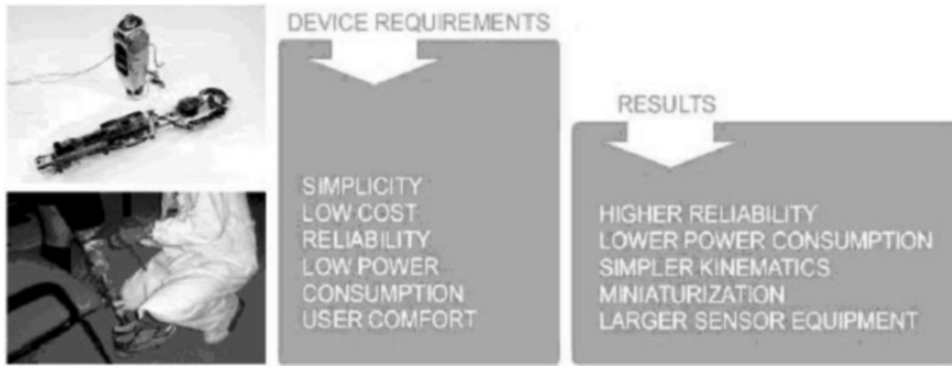


Figure 1: General design requirements.

2 DESIGN METHOD OF THE M-LEG SYSTEM

Accurate design of a lower limb prostheses requires analysis aimed at defining shapes, materials and usability. Analysis issues are related to general requirements coming from both experience and commitment outlines. One of the major goals in the research work is to design a prosthesis equipped with some device for storing energy. Then the system should also enable an amputee to perform almost the same type of (even complex) movements as those performed with a natural limb. Another objective of this project is the development of the mechanical structure strongly oriented to criteria of maximum reliability. So two criteria were identified for carrying on an effective design process: a functional and a structural criterion. When the design was in progress, however, a hard reduction in sizes was crucial. It aimed to reduce overall prosthesis weight and get the best compactness compared to requirements and constraints. On the other hand, the prosthesis must satisfy two different requirements about the de-ambulation. The first is stability in both static and dynamic conditions. Most geometries are drawn in relation to steady loads and robust load cycle response. The second requirement is related to specific leg and foot trajectory, so it requires a variable linkage. Naturally in the M-Leg prosthesis both requirements are satisfied within the same system. A prosthetic device for a thigh amputee must allow general de-ambulation conditions where each movement situation, i.e. walking, climbing stairs, sitting, running, shows different kinematical and dynamic characteristics. The prosthetic mechanism must be designed for flexible efficiency in all of them. In the prostheses design over the last few years two phenomena have come to light:

- The exponential development of electronics applied to prostheses;
- The availability of wide range typology of devices - from those equipped with only a spring to those with hydraulic circuits.

These two elements are closely connected; in fact, both are linked to the fast progress in electronics, to the consequent cost reduction, to increased processing and information storage performance and, most of all, to the increasing convergence of mechanics and electronics. They must be managed in an innovative way, encoding a methodology starting, for instance, from the design approach.

The knowledge in creation processes, analysis methods and design procedures allows a team to use not only the most suitable technology but also a methodological approach for complex problems solving. This process involves different stages of iterated validation. Researchers and designers fundamental task lies in the ideas of "materialization" through effective methods, pursuing fast and competitive product for market. Even in research field, effective tools are to be developed in order to gain large yield in innovative applications. The decision to begin a new system in the high tech devices sector involves the undertaking of a process that is generally not only long and expensive in procedural terms but also in cognitive terms. The first step is the design phase. It includes an initial project plan involving:

- analysis of the available technologies even not immediately suitable for requirements to be accomplished;

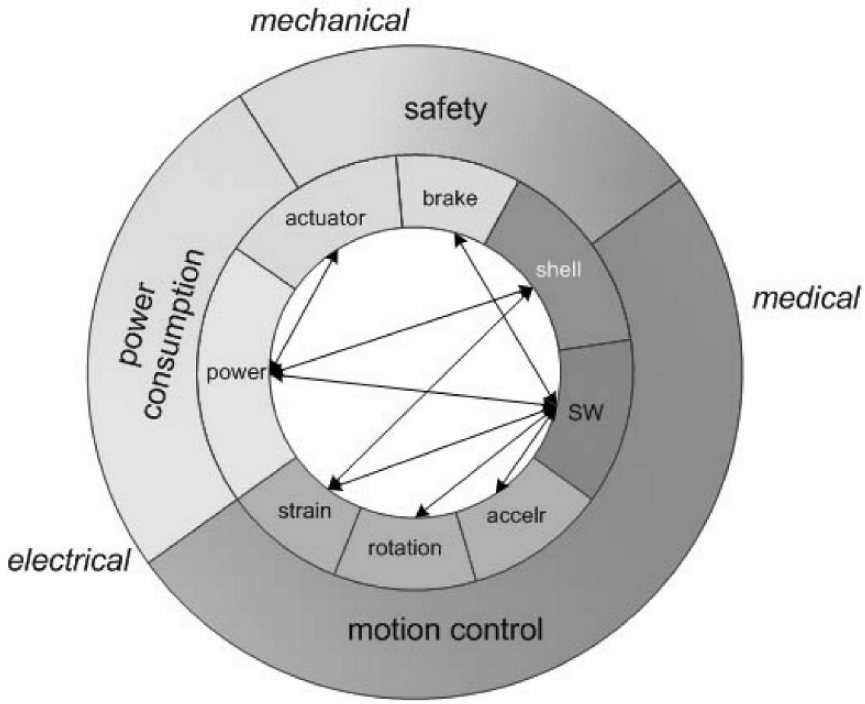


Figure 2: Requirements and components relationships.

- selection of technological tools to be used;
- choice of materials and validation protocol;
- realization of prototypes;
- analysis of available clinical data, both tested with patient and from literature;
- assessment and reformulation of specifications;

As a result, a structure of relationships among overall requirements leads towards project statements. This procedure may be codified and shared among the project team. Such procedure has been applied in last activity and final prototype production. In the next section it will be described the analysis methodologies transfer to project details and components.

3 ACTUAL RELEASE OF ARTIFICIAL LIMB PROSTHESIS. FROM DESIGN METHOD TO COMPONENT DESIGN

Long trained prototypes and improvement of skills in limb prosthesis development have drawn into final release of the artificial knee. This is due to a complete review of previous releases and to the fulfilling of special requirements in every detail. Conceptual scheme in Figure 2 depicts the relationships between project requirements, functional features of the device and related hardware components. The scheme helps the team to manage the design effort. The design action is particularly devoted to clinical and safety aspects. Moreover, from a technical point of view, the motion control involves the largest part of hardware components. Many efforts are given in accomplishing an artificial device behaviour as natural as possible. Both these high level requirements are interconnected in the software features for control. For this reason, the control coding represents the final largest activity. Many tests and control concepts are developed on the basis of measurements and sensors acquisition. In section 4 it will be described the general architecture and the tests carried for that. Finally, another large efforts in developing the final release is due to sensor equipment. It has been enriched from the last release and many components have been re-engineered.

Both safety and motion regulation are due to a brake system. Figure 3 depicts the global requirements, most of which are related to a robust and reliable braking system. The requirements

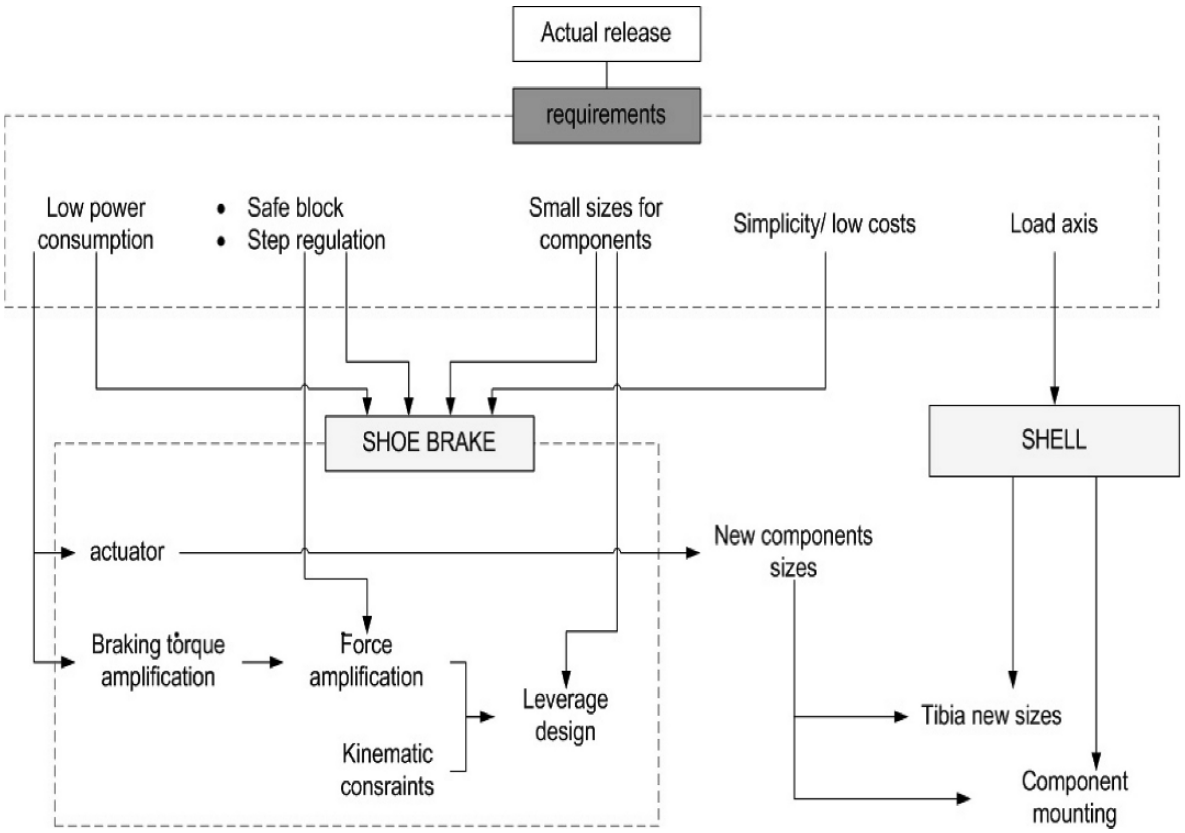


Figure 3: Brake system requirements analysis.

analysis in the scheme makes the general requirements of Figure 2 deeper. It shows the features of core components to develop or re-design. Design choices in the whole mechanical equipment are taken upon this analysis. The produced prototype works on combined active and resistive principles. The device is not designed to autonomously provide energy for walking. For this reason a constant force spring is used to store elastic energy, during hip backward flexion. This energy, coming from user stump, is given back to the artificial limb by the spring itself. The step regulation, during both the walking and other conditions, is due to a shoe brake action. It is mounted inside the artificial knee and provides reactive force and safety. Therefore the brake system represents the largest design effort from a mechanical point of view. In analysis scheme, the choices about the braking technology, the auxiliary mechanisms and the mounting structure are made on the basis of a well planned design process.

The mechanical system is designed to be functional to control action. The control itself, however, is submitted to the same general requirements of Figure 1. In particular low power consumption, simplicity and low costs are the main features to achieve in software development. The electronic control must be reliable and effective using a limited amount of memory and processing performances. The design challenge is related to optimization of control action by the available hardware. For this reason the device is equipped with several sensors which supply information to recognize artificial limb dynamic. Finally, the design aspects are very important for the compactness of the device. It must be stand alone, reliable and safe. The design contribution is shown in hardware components definition, in the structure and shape of the device in order to let the user feel comfortable. The user must be supported and facilitate during all motion situations, allowing flexibility of movements and stability.

As a result a very compact device is developed and tested. In order to fulfil many requirements, the electro-mechanical brake is used as regulation system. Moreover, it's mounted onto the main knee

Figure 4: Final release. Design, compactness and test.

joint with the housing of both sticking parts belonging to different rotating parts. Functional kinematical components are all mounted around the knee hinge, like the elastic spring around the knee joint axis. M-Leg is a semi-passive prosthesis because of partial potential energy accumulation. The particular shape of the spring makes it very easy to control. The overall compactness is evident from the electronic equipment above anything else. The micro-controller by STMicroelectronics allows data acquisition, signal conditioning and output generation; it is a very miniaturized equipment and provides the correct execution of the control algorithm. It must be pointed out that, under many aspects, the innovatory criteria applied along all the phases of the development are original in solutions, ever used before in none of the existing prosthesis.

4 CONTROL STATEMENTS METHODOLOGY

The control target lies in device adaptation to different dynamic conditions of user motion. The device must follow the behaviour of a natural leg and give a good mechanical response to user needs in equilibrium and mobility. It must do this in real time mode. Information coming from sensors input are very important to outline the current situation. The update of signal reading and output elaboration allow the artificial knee to supply the right action. The input channels are knee joint rotation, stress on lower leg structure and upper segment acceleration. The signals are provided by common strain gauges for compression and bending, and micromachined sensors for inertial parameters. The knee rotational speed is calculated by the rotation angle derivative. A calibration session has been done before using such signal. The kinematics of knee joint is single-centre, i.e. it has only one centre of rotation, and an external reference is used to check the linearity of sensor response. This procedure is required because the potentiometer is not mounted directly on rotation axes, but its connected to displaced integral shaft.

The software design comes after the acquisition session of the whole sensor equipment. This is necessary to find out the recursive patterns in step evolution and, in parallel, in signal records. The

pattern recognition phase is especially done for walking conditions. This is the case of major content in regulation statements. The walking shows periodicity of profile during the step cycle. But it's marked by a large variability. This is added to noise and variability of input signals. As a result it happens to be very useful to have several sensor available for step condition clustering. It's only by all signals comparison that the walking behaviour can be recognized. The preliminary phase for control algorithm design is to recognize periodical pattern at a reference conditions and to use the brake action without any regulation. (The fundamental feature of an electronic controlled device is the real time adaptation to different conditions).

The walking pattern recognition is the result of an accurate analysis on the acquisition of final release device. The analysis is based upon the long trained experience in step recognition during the many years limb prosthesis development. That experience proves to be very useful now that the input signals for control are related to final device and very reliable.

In this section the preliminary study of control logic is discussed. First the signal acquisition from sensor is described. The records are used to define the signal recurrent patterns. Pattern are the basis on which a real time recognition and regulation algorithm is able to work. Then a number of states are defined in order to cluster the recorded patterns. This phase is very important because it is the modelling approach for software architecture. The states definition allows to set the transitions between states and the pattern related to transitions. Then the control unit is turned on, but only for simple constant impulse. This working mode is used during explorative tests in order to map the relationship between step velocity, sensors information and braking effect.

The motion analysis starts from walking. The framework for signal interpretation is the natural walking. the topics is well known and several studies have been done in Biomechanics in last decades. Many techniques let experts to measure biometric parameters, such as rotations, angles, segment position and so on. Literature data are very important in finding out related pattern in records from an artificial device. Such data set the *walking cycle* or *step* as the complete motion of both lower limbs between two following resting upon ground by the same foot. The step can be split into two main phases: the stance phase, when a foot is touching the ground and the body weight is diversely leaning on that foot, and the swing phase, when the same foot is lifted from the ground and flies straightforward. The stance starts from the heel rest. Then the foot sole rolls as long as the toe leaves the ground. In that moment the swing starts till the next heel ground touch.

The whole cycle is made up of 60% of stance and 40% of swing. The symmetry and periodicity of walking may induce to give the same duration to both the phases, but for a small amount of cycle both the feet are resting on the ground. This is counted in stance. There are two short interval of simultaneous foot resting within the cycle, each counting the 10%. Both stance and swing phases can be divided into sub-phases. This is due to better understanding the step dynamic and recognizing it in signals records.
The stance phase is formed by five sequences.

1. initial contact: this is very critical in the step dynamic. The safety of standing on the artificial limb depends on this moment for the largest part. The firmness of the artificial limb must be comparable to natural one, both for safety and for self confidence in motion.
2. first double touch (10% of walking cycle): the body weight is pushed forward lifting the rear foot heel and lowering the front foot toe. In this phase the weight is passing from a leg to the other and it can be easily detected by the stress on the device structure.
3. half touch (20%): from the lifting of the rear foot toe to the lifting of the front foot heel. During this phase the rear foot passes the front one. The weight rests on a single limb.
4. final touch (20%): starts from the resting foot heel lifting and goes up to the finish of the other limb swing phase. This is a very complex and slight movement to detect. The touching limb shows a flexion and a waving pattern affected by large variability.
5. second double touch (10%): inverted compared to the first.

The swing phase is formed by three sequences:

1. swing start (10%): starts from the lifting of foot toe. The limb gets a backward acceleration.
2. half swing (15%): the knee flexion reaches the maximum extension. The sub-phase ends at the touching heel lifting. It's hard to recognize because of the large

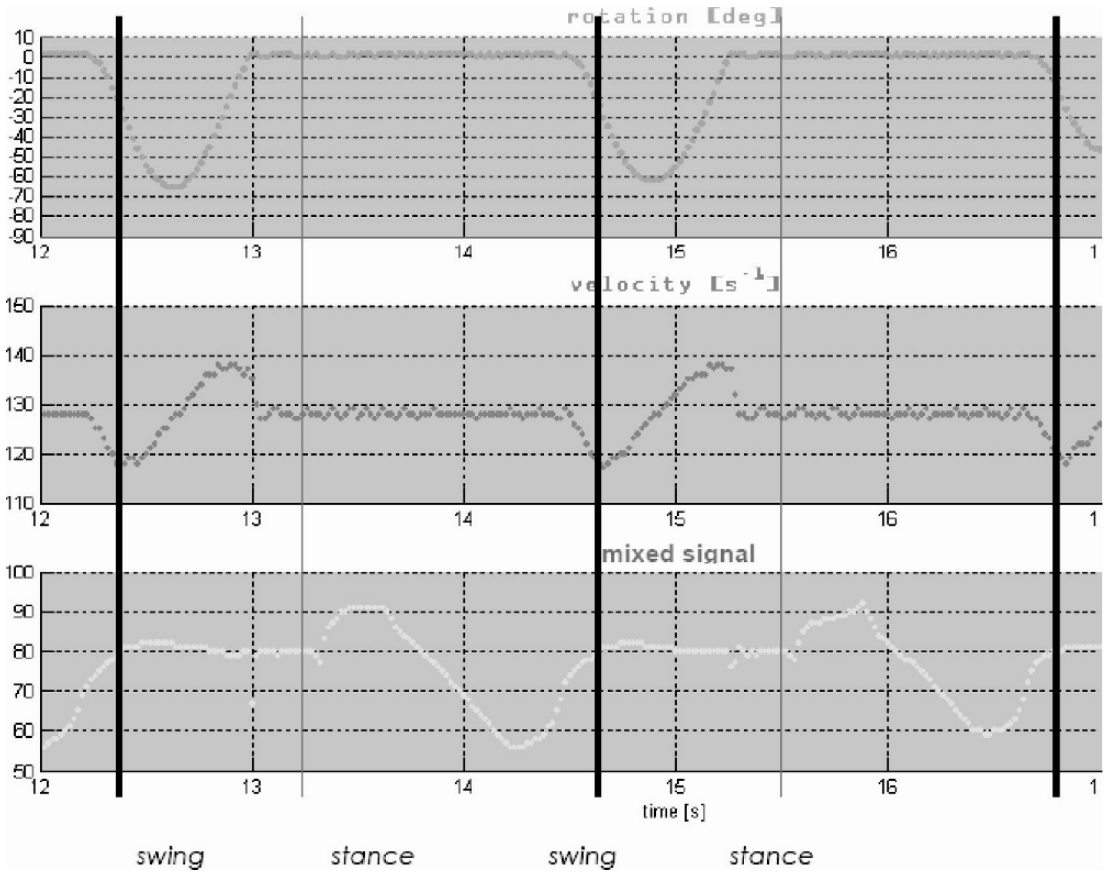


Figure 5: Acquired signals from sensors equipment.

variability of simultaneous values of data related to maximum extension and heel lifting.

3. end of swing (15%): starts from the touching heel lifting till the flying heel touching. The limb decelerates in order to prepare the following ground touch.

The sensors signal records are analyzed in comparison with the natural walking cycle definition. This is useful for modelling the step behaviour and building the software architecture. In real time control is not so important to recognize and define the whole set of sub-phases.

But there are some critical transitions that must be detected and accomplished. In particular, it's necessary to react to the swing end before the ground contact, the flexion-extension of flying limb and the whole weight resting upon only one limb for stability. The signal used to detect the limb behaviour are the rotation angle, its derivative, a mixed signal coming from an arrangement of compression and bending signals. These signals are given by a double Wheatstone bridge.

First, such signal are acquired in a test session with the brake system turned off. The black lines in Figure 5 mark the limits of each walking cycle. The red ones mark the stance and swing phases. The rotation angle is reported in indirect degrees size. This is due to the potentiometer return shaft. The relationship between the knee rotation and the potentiometer shaft angle has an amplifying factor due to the different diameters of return mechanism. The calibration set up guarantee the linear ratio between knee rotation and potentiometer angle. The mixed signal size is reported in percentage of weight. The signal gets the contribution of two types of stress signal, so the size is not directly related to an absolute value. The signal used for first analysis are collected from a standard step succession. The user is invited to walk as naturally as usual trying to keep the speed constant. The resulting signals are

averaged among several walking acquisition on the basis of a predetermined reference point. The output pattern are very regular and predictable of standard pattern.

The rotation signal shows the typical pattern, no hyperextension is supposed to be detected and the knee flexion happens to be short compared to step cycle. This is due to the reduction of swing percentage in prosthesis users. The flying phase of the artificial limb is faster than the natural one.

One of the objective in device control is to allow the user walk as much naturally as possible. This means to arrange the symmetry of walking cycle between both the limbs. The zero axes crossing in velocity, related to changing versus of rotation belongs to a narrow distribution, and the average value of maximum knee angle recorded is 45°. It's smaller than natural value because of the shorter duration of knee flexion. The lower leg has not enough time to accelerate and reach larger values. The region of rotation related to maximum extension is one of the most interesting in regulation of rotation range. The reason lies in complete absence of direct control by the user. The user has non chance to control the artificial limb backward flexion. This is over the impulsive energy give to the hip at the start of swing phase.

The mixed signal is very useful to evaluate the swing-stance transition and the stance sub-phases. The mean non-scaled value is related to absence of weight. It corresponds to swing phase when the rotation is active and for a while after the complete knee extension just before touching the ground. After the heel touches the ground an increase in signal due to compression stress is gathered. The peak is short and not marked because of the step velocity. The body weight, in fact, is thrown straightforward passing the vertical axis of the device. In this way the torque due to bending changes sign and decreasing the value below the mean. The absolute value is larger the compression phase one because of the longer leverage for torque and the duration of rolling on the foot sole. In the final release the two contribution are separately evaluated.

From this first analysis two main output are available: the states definition and the transition average values. These issues are fundamental in setting the architecture model for control software and in software requirements statements. These features are related to transitions, so the braking action coming from control regulation must fulfil the needs of the user shown through the signal record.
From the detected pattern it can be assumed:

1. a brake action is required between swing and stance in order to guarantee the safety and stability in touching the ground. The brake must be on as long as the weight is passed across the vertical axis.
2. the velocity at the end of swing phase drops very quickly. This is due to initial

Figure 6: Acquisition tests at constant

acceleration in forward rotation due to spring elastic force and the mechanical block at the complete extension, 0°. This provokes a stroke to the device transmitted to the socket and, finally, to the hip and the backbone of the user. The return rotation must be decelerated before the end of its range.

3. other requirements could be revealed by a finer analysis of device behaviour with the brake turned on. An important feed back is given by the user, pointing some features he may be consider useful or comfortable.

These requirements come from pattern first analysis and must be added to general ones dealing with emergency management, safe standing upon the device with the whole or partial body weight, different motion situations. In particular sitting down and climbing stairs are test routine run in order to achieve typical data. The methodology is the same about different shaped patterns.

The control model is thought to be implemented through a Finite State Machine (FSM). It's a traditional tool to describe formal requirements and relationships between defined states. It's not the control algorithm structure but the ideal framework of transition management. Such tool is quite powerful in setting states and transitions, is fit for limited amount of memory of processor and can be managed by several people inside the multidisciplinary team. A first prototype of FSM is implemented to turn on the brake system in detected and required points. The initial rules are based only on pattern analysis. This feature allows the tester to run some experiments for mapping the dynamic relationship between braking and step conditions.

Test regulation in control logic is assigned to fixed velocity/braking position ratio. It's obviously a simplification because this ratio changes during the walking. But for in lab test on leg simulator this is very useful. It helps to check the right brake activation due to sensor record and software regulation. The dynamic step regulation must adapt the braking action to the velocity and rotation angle. The dynamics of brake achieve effective resistive torque as a function of velocity, angle and time delay in impulse transmission. The larger the velocity, the larger must be the angle of activation or, in other words, the advance in getting the speed reduction. This set of relationships must be fitted among an empirical data setting and collection. The experiments take the first step towards the adaptive control required to electronic controlled prosthesis. They are carried by a specific tool of calibration and tuning described below. It's used to change regulation parameters both for initial test and for customization of stand alone final release. For such reasons a Calibration and customization tool has been developed.

Initial setting must be run before using the device. The controller sets internal parameters on the basis of user features. The main quantity to measure is the user weight. As usual procedure, the user

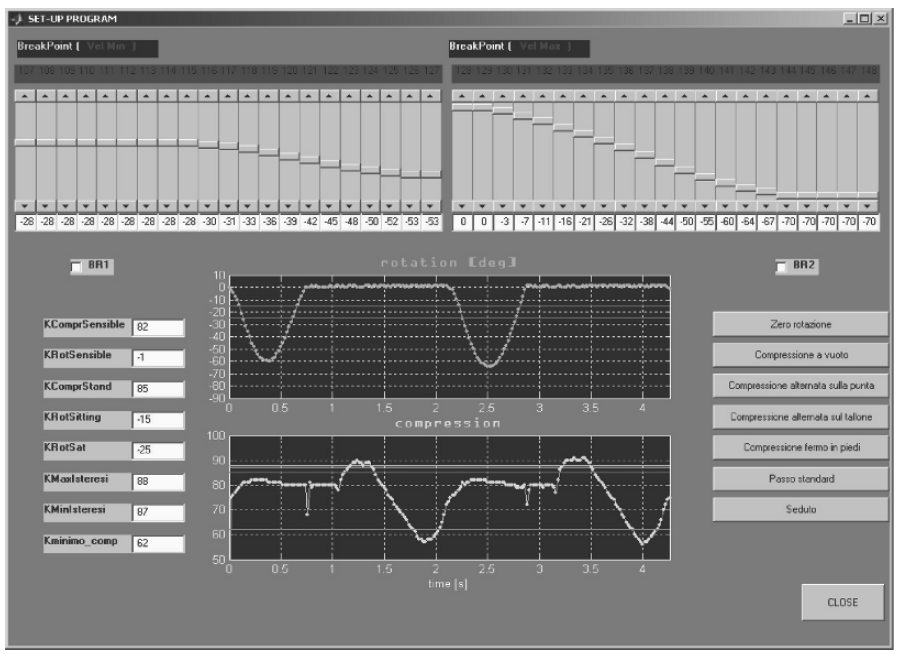


Figure 7: Tuning, setting and customization tool.

stands on the device for a while. The device records a large amount of values of stress on its structure. The duration of record may last from six to twenty seconds. This allows the user to feel free to stand in natural way. In such way the weight distribution on both legs, the natural one and the artificial one, has the chance to vary in a wide range of usual conditions.

The setting tool has been developed for fine tuning the recorded parameters. The operator can set thresholds or constants related to step regulation. The fine tuning of thresholds is not so usual because of the initial auto-setting procedure. It's very useful to change comparison values inside control software. This is done to check particular features of device behaviour. For instance, several tests were run for understanding the right shape of velocity table curve during the software development. The user was forced to experience the same brake activation response for the whole walk. In that way the step time and step percentage profile were forced to be tightly constant. The user was so forced to walk at fixed velocity. The user was helped in do this by walking on a *tapis roulant* so that he could slightly feel the unnatural step regulation. The amount of tests was collected varying walking velocity and brake activation position.

5 CONCLUSIONS

A final prosthesis prototype is the result of a long design process. Experience and skills are supported now by coded methodologies and analysis tool. This process starts from a design approach leading towards details optimization. It's important to underline the methodology contribution to several re-engineering stages. By means of final release a large development in direct signal acquisition and testing became possible. The proposed methodology for step analysis was done with the constant help and experience supply of patients and INAIL staff. The presented methodology is basic for the further FSM development with self-learning and adaptive features.

ACKNOWLEDGEMENTS

The project has been developed in cooperation with INAIL National Prosthesis Centre – Budrio (Bologna, Italy) and STMicroelectronics. Special thanks for research contribution in topics discussed in this paper to G.Verni (INAIL), to B. Murari and F.Pasolini (STM), to S. Della Santina, C.Abertelli, C.Pintaudi and C.Maglie for hw and sw development and to M.Andorno for design (Politecnico). A fundamental contribution in testing and feedback by C. Rinaldi.

REFERENCES

AAVV, *Ausili e ortesi in medicina, Vol. 3*, Editrice Ricerca Medica (Na) 1998

AAVV, *Biomechanics of the musculo-skeletal system (II edition)*, Wiley 1999

AAVV, *Otto Bock Manuale Protesi, Protesi per arto inferior,* SCHIELE & SCHON 1988

Canina, M., Verni, G., Valentini, P., *Gamba "intelligente" cambia il passo secondo il terreno*, in La Repubblica Salute anno 7 n. 286, 20 settembre 2001

Canina M. et al., *Innovative system for the accumulation of energy of the step in a limb prosthesis,* accepted at 11th World Cong. Mech. Mach. Science, China, August 18–21, 2003.

Canina M., Vicentini F., Rovetta A., Innovative Design, Development And Prototyping Of Knee Prosthesis, *in Proceedings of ROBTEP 2004, Automation – Robotics in theory and practice, 2004*

Canina M., Vicentini F., Rovetta A., Innovative Wide Sensors Integration for Smart Bio-robotic Prosthesis Control, *in Proceedings of RAAD'04, 13th International Workshop on Robotics, 2004*

Dornig A., Le molle, CLUP 1973

Hugh Herr, *Presentation highlights: Prosthetic and orthotic limbs*, J. Rehabilitation Res. & Dev. Vol. 39 No. 3 (supplement) pp. 11–12, 2002

Ju, M. S., Yang, Y. F. and Hsuch, T.C., *Development of actively controlled electro-hydraulic above-knee prosthesis,* Proc. Romansy 10/ the 10th CISM-IFToMM symposium, theory and practice of robots and manipulators, Spring-Verlag Ed., pp. 367–372, 1995

Kapandji, *Fisiologia articolare* – Soc Editrice DEMI-Roma 1974

Kim, J. H., Oh, J. H., 2001, *Development of an above knee prosthesis using MR damper and leg simulator,* Proc. Conf. Rob. Aut., Seoul, 1998

Nam P. Suh, *The principles of Design*, Oxford University Press 1990

Nigg, B. M., Herzog, W., *Biomechanics of the musculo-skeletal system*, 2nd edition, John Wiley and Sons, England, 1999

Peeraer L. et al., *Development of EMG-based mode and intent recognition algorithms for a computer-controlled above-knee prosthesis,* J. Biomed. Eng. Vol. 12, May, pp. 178–182, 1990

Popovic, D. et al., *Optimal control for the active above knee prosthesis*, J. Biomed. Eng. Vol. 19, pp. 131–150, 1991

Rovetta, A., Wen, X., *Biorobotic in a new artificial leg*, IEEE Int. Symp. on Int. Rob. and Manipulators, Krakow, 1990

Rovetta, M. Canina, P. Allara, G. Campa, S. Della Santina, *Biorobotic design criteria for Innovative Limb Prosthesis*, Mechanika – 2001, Proceedings of the International Conference, Kaunas, Lituania, April 2001

Rovetta A., Canina M., Campa G., Della Santina S., *Biorobotic design criteria for Innovative Limb Prosthesis*, 9th International Symposium on Intelligent Robotic Systems, SIRS'2001, Toulouse, France, 18–20 July 2001

Rovetta A., Canina M., Allara P., Campa G., Della Santina S., *Biorobotic design criteria for Innovative Limb Prosthesis*, Icar 2001, International Conference on Advanced Robotics, Budapest, 22–26 August 2001

Slavica, J., Tamara, J., Vladimir, G., Dejan, P., *Three machine learning techniques for automatic determination of rules to control locomotion,* IEEE Trans. Biomedical Eng., Vol. 46, No. 3, pp. 300–310, 1999

Winter, D. A., *Biomechanics and motor control of human movement*, Wiley-Interscience Publication, 2nd edition, 1990

Zlatnik, D., *"Intelligently controlled above knee prosthesis"*, the 4th Int. conf. on motion and vibration control

DIRECT GRADIENT-BASED REINFORCEMENT LEARNING FOR ROBOT BEHAVIOR LEARNING

Andres El-Fakdi, Marc Carreras and Pere Ridao
Institute of Informatics and Applications, University of Girona, Politecnica 4, Campus Montilivi, 17071 Girona, Spain
aelfakdi@eia.udg.es, marcc@eia.udg.es, pere@eia.udg.es

Keywords: Robot Learning, Autonomous robots.

Abstract: Autonomous Underwater Vehicles (AUV) represent a challenging control problem with complex, noisy, dynamics. Nowadays, not only the continuous scientific advances in underwater robotics but the increasing number of sub sea missions and its complexity ask for an automatization of submarine processes. This paper proposes a high-level control system for solving the action selection problem of an autonomous robot. The system is characterized by the use of Reinforcement Learning Direct Policy Search methods (RLDPS) for learning the internal state/action mapping of some behaviors. We demonstrate its feasibility with simulated experiments using the model of our underwater robot URIS in a target following task.

1 INTRODUCTION

A commonly used methodology in robot learning is Reinforcement Learning (RL) (Sutton and Barto, 1998). In RL, an agent tries to maximize a scalar evaluation (reward or punishment) obtained as a result of its interaction with the environment. The goal of a RL system is to find an optimal policy which maps the state of the environment to an action which in turn will maximize the accumulated future rewards. Most RL techniques are based on *Finite Markov Decision Processes* (FMDP) causing finite state and action spaces. The main advantage of RL is that it does not use any knowledge database, so the learner is not told what to do as occurs in most forms of machine learning, but instead must discover actions yield the most reward by trying them. Therefore, this class of learning is suitable for online robot learning. The main disadvantages are a long convergence time and the lack of generalization among continuous variables.

In order to solve such problems, most of RL applications require the use of generalizing function approximators such artificial neural-networks (ANNs), instance-based methods or decision-trees. As a result, many RL-based control systems have been applied to robotics over the past decade. In (Smart and Kaelbling, 2000), an instance-based learning algorithm was applied to a real robot in a corridor-following task. For the same task, in (Hernandez and Mahadevan, 2000) a hierarchical memory-based RL was proposed.

The dominant approach has been the value-function approach, and although it has demonstrated to work well in many applications, it has several limitations, too. Function approximator methods in "value-only" RL algorithms may present convergence problems, if the state-space is not completely observable (POMDP), small changes in the value function can cause big changes in the policy (Bertsekas and Tsitsiklis, 1996).

Over the past few years, studies have shown that approximating directly a policy can be easier than working with value functions, and better results can be obtained (Sutton et al., 2000) (Anderson, 2000). Instead of approximating a value function, new methodologies approximate a policy using an independent function approximator with its own parameters, trying to maximize the expected reward. Examples of direct policy methods are the REINFORCE algorithm (Williams, 1992), the direct-gradient algorithm (Baxter and Bartlett, 2000) and certain variants of the actor-critic framework (Konda and Tsitsiklis, 2003). Some direct policy search methodologies have achieved good practical results. Applications to autonomous helicopter flight (Bagnell and Schneider, 2001), optimization of robot locomotion movements (Kohl and Stone, 2004) and robot weightlifting task (Rosenstein and Barto, 2001) are some examples.

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 175–182.

The advantages of policy methods against value-function based methods are various. The main advantage is that using a function approximator to represent the policy directly solves the generalization problem. A problem for which the policy is easier to represent should be solved using policy algorithms (Anderson, 2000). Working this way should represent a decrease in the computational complexity and, for learning control systems which operate in the physical world, the reduction in time-consuming would be notorious. Furthermore, learning systems should be designed to explicitly account for the resulting violations of the Markov property. Studies have shown that stochastic policy-only methods can obtain better results when working in POMDP than those ones obtained with deterministic value-function methods (Singh et al., 1994). On the other hand, as dis-advantage, policy gradient estimators used in these algorithms may have large variance, so these methods learn much more slower than RL algorithms using a value function (Marbach and Tsitsiklis, 2000) (Sutton et al., 2000) (Konda and Tsitsiklis, 2003) and they can converge to local optima of the expected reward (Meuleau et al., 2001).

In this paper we propose an on-line direct policy search algorithm based on Baxter and Bartlett's direct-gradient algorithm OLPOMDP (Baxter and Bartlett, 1999) applied to a real learning control system in which a simulated model of the AUV URIS (Ridao et al., 2004) navigates a two-dimensional world. The policy is represented by a neural network whose input is a representation of the state, whose output is action selection probabilities, and whose weights are the policy parameters. The proposed method is based on a stochastic gradient descent with respect to the policy parameter space, it does not need a model of the environment to be given and it is incremental, requiring only a constant amount of computation step. The objective of the agent is to compute a stochastic policy (Singh et al., 1994), which assigns a probability over each action. Results obtained in simulation show the viability of the algorithm in a real-time system.

The structure of the paper is as follows. In section II the direct-policy search algorithm is detailed. In section III a description of all the elements that affect our problem (the world, the robot and the controller) are commented. The simulated experiment description and the results obtained are included in section IV and finally, some conclusions and further work are included in section V.

2 THE RLDPS ALGORITHM

A partially observable Markov decision process (POMDP) consists of a state space *S*, an observation space *Y* and a control space *U*. For each state $i \in S$ there is a deterministic reward *r(i)*. As mentioned before, the algorithm applied is designed to work on-line so at every time step, the learner (our vehicle) will be given an observation of the state and, according to the policy followed at that moment, it will generate a control action. As a result, the learner will be driven to another state and will receive a reward associated to this new state. This reward will allow us to update the controller's parameters that define the policy followed at every iteration, resulting in a final policy considered to be optimal or closer to optimal. The algorithm procedure is summarized in Table 1.

Table 1: Algorithm: Baxter & Bartlett's OLPOMDP.

1: Given:
- $T > 0$
- Initial parameter values $\theta_0 \in \mathbb{R}^K$
- Arbitrary starting state i_0

2: Set $z_0 = 0$ ($z_0 \in \mathbb{R}^K$)

3: **for** $t = 0$ to T **do**

4: Observe state y_t

5: Generate control action u_t according to current policy $\mu(\theta, y_t)$

6: Observe the reward obtained $r(i_{t+1})$

7: $$z_{t+1} = \beta z_t + \frac{\nabla \mu_{u_t}(\theta, y_t)}{\mu_{u_t}(\theta, y_t)}$$

8: $$\theta_{t+1} = \theta_t + \alpha r(i_{t+1}) z_{t+1}$$

9: **end for**

The algorithm works as follows: having initialized the parameters vector θ_0, the initial state i_0 and the gradient $z_0 = 0$, the learning procedure will be iterated T times. At every iteration, the parameters gradient z_t will be updated. According to the immediate reward received $r(i_{t+1})$, the new gradient vector z_{t+1} and a fixed learning paramenter α, the new paramenter vector θ_{t+1} can be calculated. The current policy μ_t is directly modified by the new parameters becoming a new policy μ_{t+1} that will be followed next iteration, getting closer, as $t \to T$ to a final policy μ_T that represents a correct solution of the problem.

In order to clarify the steps taken, the next lines will relate the update parameter procedure of the algorithm closely. The controller uses a neural network as a function approximator that generates a stochastic policy. Its weights are the policy parameters that are updated on-line every time step. The accuracy of the approximation is controlled by the parameter $\beta \in [0,1)$.

The first step in the weight update procedure is to compute the ratio:

$$\frac{\nabla \mu_{u_t}(\theta, y_t)}{\mu_{u_t}(\theta, y_t)} \tag{1}$$

for every weight of the network. In AANs like the one used in the algorithm the expression defined in step 7 of Table 1 can be rewritten as:

$$z_{t+1} = \beta z_t + \delta_t y_t \tag{2}$$

At any step time *t*, the term z_t represents the estimated gradient of the reinforcement sum with respect to the network's layer weights. In addition, δ_t refers to the local gradient associated to a single neuron of the ANN and it is multiplied by the input to that neuron y_t. In order to compute these gradients, we evaluate the soft-max distribution for each possible future state exponentiating the real-valued ANN outputs $\{o_1, ..., o_n\}$ being *n* the number of neurons of the output layer (Aberdeen, 2003).

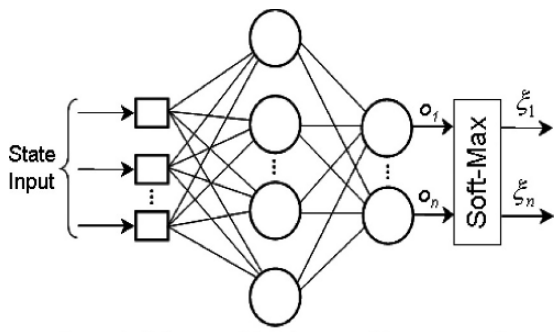


Figure 1: Schema of the ANN architecture used.

After applying the soft-max function, the outputs of the neural network give a weighting, $\xi_j \in (0,1)$ to each of the vehicle's thrust combinations. Finally, the probability of the i^{th} thrust combination is then given by:

$$\Pr_i = \frac{\exp(o_i)}{\sum_{z=1}^{n} \exp(o_z)} \tag{3}$$

Actions have been labeled with the associated thrust combination, and they are chosen at random from this probability distribution.

Once we have computed the output distribution over the possible control actions, next step is to calculate the gradient for the action chosen by applying the chain rule; the whole expression is implemented similarly to *error back propagation* (Haykin, 1999). Before computing the gradient, the error on the neurons of the output layer must be calculated. This error is given by expression (4).

$$e_j = d_j - \Pr_j \tag{4}$$

The desired output d_j will be equal to 1 if the action selected was o_j and 0 otherwise (see Figure 2).

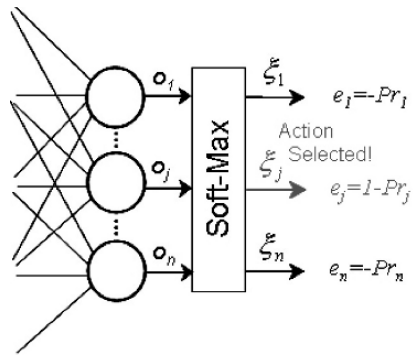


Figure 2: Soft-Max error computation for every output.

With the soft-max output error calculation completed, next phase consists in computing the gradient at the output of the ANN and back propagate it to the rest of the neurons of the hidden layers. For a local neuron *j* located in the output layer we may express the local gradient for neuron *j* as:

$$\delta_j^o = e_j \cdot \varphi_j'(o_j) \tag{5}$$

Where e_j is the soft-max error at the output of neuron *j*, $\varphi_j'(o_j)$ corresponds to the derivative of the activation function associated with that neuron and o_j is the function signal at the output for that neuron. So we do not back propagate the gradient of an error measure, but instead we back propagate the soft-max gradient of this error. Therefore, for a neuron j located in a hidden layer the local gradient is defined as follows:

$$\delta_j^h = \varphi_j'(o_j) \sum_k \delta_k w_{kj} \tag{6}$$

When computing the gradient of a hidden-layer neuron, the previously obtained gradient of the following layers must be back propagated. In (6) the term $\varphi_j'(o_j)$ represents de derivative of the activation function associated to that neuron, o_j is the function signal at the output for that neuron and finally the summation term includes the different gradients of the following neurons back propagated by multiplying each gradient to its corresponding weighting (see Figure 3).

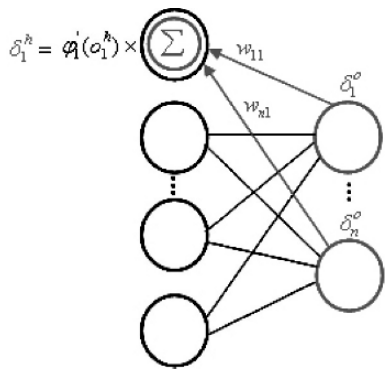


Figure 3: Gradient computation for a hidden-layer neuron.

Having all local gradients of the all neurons calculated, the expression in (2) can be obtained and finally, the old parameters are updated following the expression:

$$\theta_{t+1} = \theta_t + \gamma r(i_{t+1}) z_{t+1} \tag{7}$$

The vector of parameters θ_t represents the network weights to be updated, $r(i_{t+1})$ is the reward given to the learner at every time step, z_{t+1} describes the estimated gradients mentioned before and at last we have γ as the learning rate of the RLDPS algorithm.

3 CASE TO STUDY: TARGET FOLLOWING

The following lines are going to describe the different elements that take place in our problem. First, the simulated world will be detailed, in a second place we will present the underwater vehicle URIS and its model used in our simulation. At last, a description of the neural-network controller is presented.

3.1 The World

As mentioned before, the problem deals with the simulated model of the AUV URIS navigating a two-dimensional world constrained in a plane region without boundaries. The vehicle can be controlled in two degrees of freedom (DOFs), surge (X movement) and yaw (rotation respect z-axis) by applying 4 different control actions: a force in either the positive or negative surge direction, and another force in either the positive or negative yaw rotation. The simulated robot was given a reward of 0 if the vehicle reaches the objective position (if the robot enters inside a circle of 1 unit radius, the target is considered reached) and a reward equal to -1 in all other states. To encourage the controller to learn to navigate the robot to the target independently of the starting state, the AUV position was reset every 50 (simulated) seconds to a random location in x and y between [-20, 20], and at the same time target position was set to a random location within the same boundaries. The sample time is set to 0.1 seconds.

3.2 URIS AUV Description

The Autonomous Underwater Vehicle URIS (Figure 4) is an experimental robot developed at the University of Girona with the aim of building a small-sized UUV. The hull is composed of a stainless steel sphere with a diameter of 350mm, designed to withstand pressures of 4 atmospheres (30m. depth).

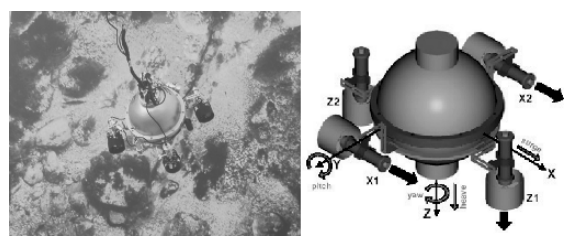


Figure 4: (Left) URIS in experimental test. (Right) Robot reference frame.

The experiments carried out use the mathematical model of URIS computed by means of parameter identification methods (Ridao et al., 2004). The whole model has been adapted to the problem so the hydrodinamic equation of motion of an underwater vehicle with 6 DOFs (Fossen, 1994))has been uncoupled and reduced to modellate a robot with two DOFs. Let us consider the dynamic equation for the surge and yaw DOFs:

$$\dot{u} = \underbrace{\frac{X}{(m-X_{\dot{u}})}}_{\gamma} - \underbrace{\frac{X_u}{(m-X_{\dot{u}})}}_{\alpha} \cdot u - \underbrace{\frac{X_{u|u|}|u|}{(m-X_{\dot{u}})}}_{\beta} \cdot u + \underbrace{\frac{\tau_p}{(m-X_{\dot{u}})}}_{\delta} \tag{8}$$

$$\dot{r} = \underbrace{\frac{N}{(m-N_{\dot{r}})}}_{\gamma} - \underbrace{\frac{N_r}{(m-N_{\dot{r}})}}_{\alpha} \cdot r - \underbrace{\frac{N_{r|r|}|r|}{(m-N_{\dot{r}})}}_{\beta} \cdot r + \underbrace{\frac{\tau_p}{(m-N_{\dot{r}})}}_{\delta} \quad (9)$$

Then, due to identification procedure (Ridao et al., 2004), expressions in (8) and (9) can be rewritten as follows:

$$\dot{v}_x = \alpha_x v_x + \beta_x v_x |v_x| + \gamma_x \tau_x + \delta_x \quad (10)$$

$$\dot{v}_\psi = \alpha_\psi v_\psi + \beta_\psi v_\psi |v_\psi| + \gamma_\psi \tau_\psi + \delta_\psi \quad (11)$$

Where $\dot{v}_x$ and $\dot{v}_\psi$ represent de acceleration in both surge and yaw DOFs, v_x is the linear velocity in surge and v_ψ is the angular velocity in yaw DOF. The force and torque excerted by the thrusters in both DOFs are indicated as τ_x and τ_ψ. The model parameters for both DOFs are stated as follows: α and β coeficients refer to the linear and the quadratic damping forces, γ represent a mass coeficient and the bias term is introduced by δ. The identified parameters values of the model are indicated in Table 2.

Table 2: URIS Model Parameters for Surge and Yaw.

	α	β	γ	δ
Units	$\left(\frac{N \cdot s}{Kg \cdot m}\right)$	$\left(\frac{N \cdot s^2}{Kg \cdot m^2}\right)$	Kg^{-1}	$\left(\frac{N}{Kg}\right)$
Surge	-0.3222	0	0.0184	0.0012
Yaw	1.2426	0	0.5173	-0.050

3.3 The Controller

A one-hidden-layer neural-network with 4 input nodes, 3 hidden nodes and 4 output nodes has been used to generate a stochastic policy. One of the inputs corresponds to the distance between the vehicle and the target location, another one represents the yaw difference between the vehicle's current heading and the desired heading to reach the objective position. The other two inputs represent the derivatives of the distance and yaw difference at the current time-step. Each hidden and output layer has the usual additional bias term. The activation function used for the neurons of the hidden layer is the hyperbolic tangent type (12, Figure 5), while the output layer nodes are linear. The four output neurons have been exponentiated and normalized as explained in section 2 to produce a probability distribution. Control actions are selected at random from this distribution.

$$\tanh(z) = \frac{\sinh(z)}{\cosh(z)} \quad (12)$$

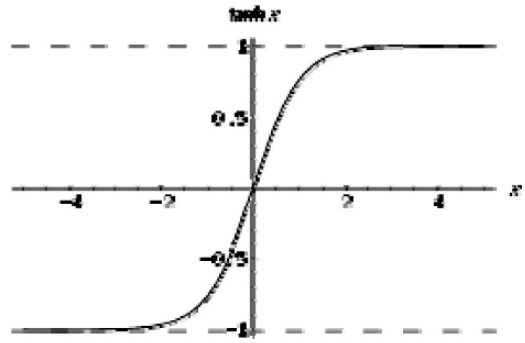


Figure 5: The hyperbolic tangent function.

4 SIMULATED RESULTS

The controller was trained, as commmented in section 3, in an episodic task. Robot and target positions are reseted every 50 seconds so the total amount of reward per episode percieved varies depending on the episode. Even though the results presented have been obtained as explained in section 3, in order to clarify the graphical results of time convergence of the algorithm, for the plots below some constrains have been applied to the simulator: Target initial position is fixed to (0,0) and robot initial location has been set to four random locations, $x = \pm 20$ and $y = \pm 20$, therefore, the total amount per episode when converged to minima will be the same.

The number of episodes to be done has been set to 100.000. For every episode, the total amount of reward percieved is calculated. Figure 6 represents the performance of the neural-network vehicle controller as a function of the number of episodes, when trained using OLPOMDP. The episodes have been averaged over bins of 50 episodes. The experiment has been repeated in 100 independent runs, and the results presented are a mean over these runs.

The simulated experiments have been repeated and compared for different values of α and β.

For $\alpha = 0.000001$:

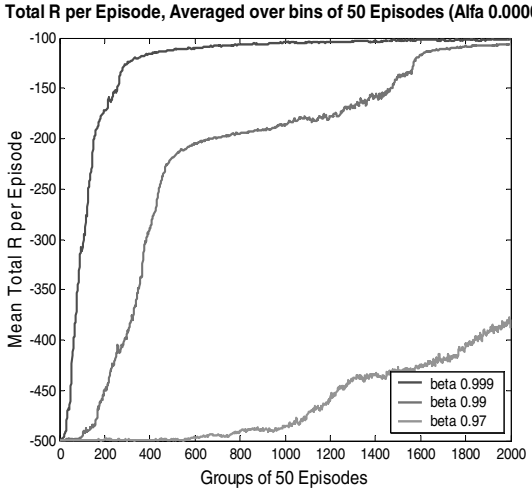


Figure 6: Performance of the neural-network puck controller as a function of the number of episodes. Performance estimates were generated by simulating 100.000 episodes, and averaging them over bins of 50 episodes. Process repeated in 100 independent runs. The results are a mean of these runs. Fixed $\alpha = 0.000001$, for different values of $\beta = 0.999$, $\beta = 0.99$ and $\beta = 0.97$.

For $\alpha = 0.00001$:

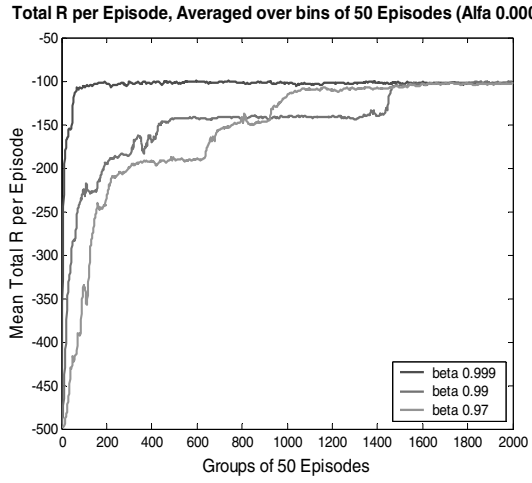


Figure 7: Performance of the neural-network puck controller as a function of the number of episodes. Performance estimates were generatedby simulating 100.000 episodes, and averaging them over bins of 50 episodes. Process repeated in 100 independent runs. The results are a mean of these runs. Fixed $\alpha = 0.00001$, for different values of $\beta = 0.999$, $\beta = 0.99$ and $\beta = 0.97$.

For $\alpha = 0.0001$:

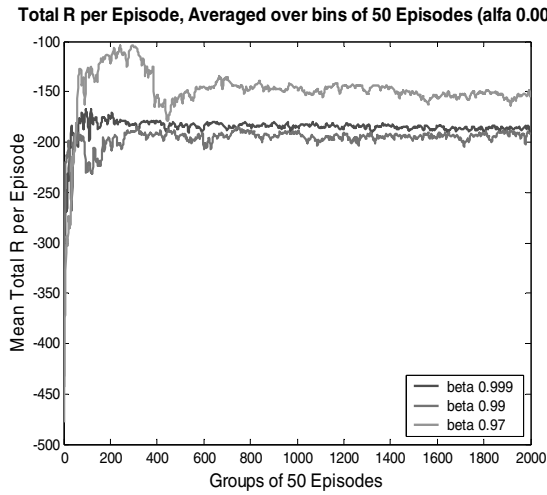


Figure 8: Performance of the neural-network puck controller as a function of the number of episodes. Performance estimates were generatedby simulating 100.000 episodes, and averaging them over bins of 50 episodes. Process repeated in 100 independent runs. The results are a mean of these runs. Fixed $\alpha = 0.0001$, for different values of $\beta = 0.999$, $\beta = 0.99$ and $\beta = 0.97$.

As it can bee apreciated in the figure above (see Figure 7), the optimal performance (within the neural network controller used here) is around -100 for this simulated problem, due to the fact that the puck and target locations are reset every 50 seconds and for this reason the vehicle must be away from target a fraction of the time. The best results are obtained when $\alpha = 0.00001$ and $\beta = 0.999$, see Figure 7.

Figure 9 represents the behavior of the trained robot controller. For the purpose of the illustration, only target location has been reseted to random location, not the robot location.

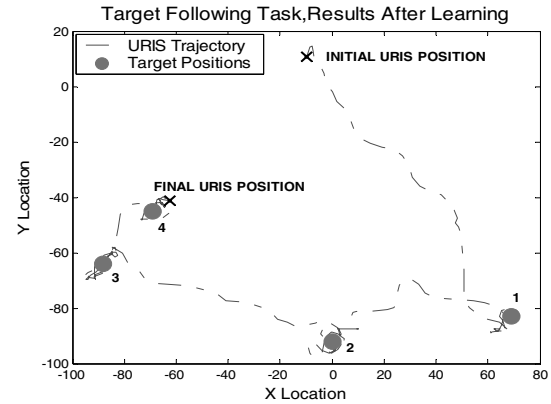


Figure 9: Behavior of a trained robot controller, results of target following task asfter learning period is completed.

5 CONCLUSIONS

An on-line direct policy search algorithm for AUV control based on Baxter and Bartlett's direct-gradient algorithm OLPOMDP has been proposed. The method has been applied to a real learning control system in which a simulated model of the AUV URIS navigates a two-dimensional world in a target following task. The policy is represented by a neural network whose input is a representation of the state, whose output is action selection probabilities, and whose weights are the policy parameters. The objective of the agent was to compute a stochastic policy, which assigns a probability over each of the four possible control actions.

Results obtained confirm some of the ideas presented in Section 1. The algorithm is easier to implement compared with other RL methodologies like value function algorithms and it represents a considerable reduction of the computational time of the algorithm. On the other side, simulated results show a poor speed of convergence towards minimal solution.

In order to validate the performance of the method proposed, future experiments are centered on obtaining empirical results: the algorithm must be tested on real URIS in a real environment. Previous investigations carried on in our laboratory with RL value functions methods with the same prototype URIS (Carreras et al., 2003) will allow us to compare both results. At the same time, the work is focused in the development of a methodology to decrease the convergence time of the RLDPS algorithm.

ACKNOWLEDGEMENTS

This research was esponsored by the spanish commission MCYT (DPI2001-2311-C03-01). I would like to give my special thanks to Mr. Douglas Alexander Aberdeen of the Australian National University for his help.

REFERENCES

D.A., Aberdeen, Policy Gradient Algorithms for Partially Observable Markov Decision Processes, PhD Thesis, Australian National University, 2003.

C. Anderson, "Approximating a policy can be easier than approximating a value function" *Computer Science Technical Report*, CS-00-101, February 10, 2000.

J.A. Bagnell and J.G. Schneider, "Autonomous Helicopter Control using Reinforcement Learning Policy Search Methods", in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, Korea, 2001.

D.P. Bertsekas and J.N. Tsitsiklis, Neuro-Dynamic Programming. Athena Scientific, 1996.

J. Baxter and P.L. Bartlett, "Direct gradient-based reinforcement learning I: Gradient estimation algorithms" Technical Report. Australian National University, 1999.

J. Baxter and P.L. Bartlett, "Direct gradient-based reinforcement learning" IEEE *International Symposium on Circuits and Systems*, May 28–31, Geneva, Switzerland, 2000.

M. Carreras, P. Ridao and A. El-Fakdi, "Semi-Online Neural-Q-Learning for Real-Time Robot Learning", in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, USA, 2003.

T.I., Fossen, *Guidance and Control of Ocean Vehicles*, John Wiley and Sons, New York, USA, 1994.

S. Haykin, *Neural Networks, a comprehensive foundation*, Prentice-Hall, Upper Saddle River, New Jersey, USA, 1999.

N. Hernandez and S. Mahadevan, "Hierarchical memory-based reinforcement learning", *Fifteenth International Conference on Neural Information Processing Systems*, Denver, USA, 2000.

N. Kohl and P. Stone, "Policy Gradient Reinforcement Learning for Fast Quadrupedal Locomotion", in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

V.R. Konda and J.N. Tsitsiklis, "On actor-critic algorithms", in SIAM Journal on Control and Optimization, vol. 42, no. 4, pp. 1143–1166, 2003.

P. Marbach and J.N. Tsitsiklis, "Gradient-based optimization of Markov reward processes: Practical Variants", Center for Communications Systems Research, University of Cambridge, Tech. Rep,. March 2000.

N. Meuleau, L. Peshkin and K. Kim, "Exploration in gradient-based reinforcement learning", Technical report AI Memo 2001–003, April 3, 2001.

P. Ridao, A. Tiano, A. El-Fakdi, M. Carreras and A. Zirilli, "On the identification of non-linear models of unmanned underwater vehicles" in Control Engineering Practice, vol. 12, pp. 1483–1499, 2004.

M.T. Rosenstein and A.G. Barto, "Robot Weightlifting by Direct Policy Search", in Proceedings of the International Joint Conference on Artificial Intelligence, 2001.

S.P. Singh, T. Jaakkola and M.I. Jordan, "Learning without state-estimation in partially observable Markovian decision processes", in *Proceedings of the 11th International Conference on Machine Learning*, pp. 284–292, 1994.

W.D. Smart and L.P Kaelbling, "Practical reinforcement learning in continuous spaces", *International Conference on Machine Learning*, 2000.

R. Sutton and A. Barto, *Reinforcement Learning, an Introduction*. MIT Press, 1998.
R. Sutton, D. McAllester, S. Singh and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation" in Advances in Neural Information Processing Systems 12, pp. 1057–1063, MIT Press, 2000.
R. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning" in Machine Learning, 8, pp. 229–256, 1992.

PART 3

Signal Processing, Systems Modeling and Control

PERFORMANCE ANALYSIS OF TIMED EVENT GRAPHS WITH MULTIPLIERS USING (Min, +) ALGEBRA

Samir Hamaci, Jean-Louis Boimond and Sébastien Lahaye
LISA
62 avenue Notre Dame du Lac - Angers, France
[hamaci, boimond,lahaye]@istia.univ-angers.fr

Keywords: Timed event graphs with multipliers, (min,+) algebra, linearization, cycle time.

Abstract: We are interested in the performance evaluation of timed event graphs with multipliers. The dynamical equation modelling such graphs are nonlinear in (min,+) algebra. This nonlinearity is due to multipliers and prevents from applying usual performance analysis results. As an alternative, we propose a linearization method in (min,+) algebra of timed event graphs with multipliers. From the obtained linear model, we deduce the cycle time of these graphs. Lower and upper linear approximated models are proposed when linearization condition is not satisfied.

1 INTRODUCTION

Timed event graphs (TEG's) are well adapted to model synchronization phenomena occuring in discrete event systems (Murata, 1989). Their behavior can be modelled by recurrent linear equations in (min, +) algebra (Baccelli et al., 1992). When the size of the model becomes very significant, techniques of analysis developed for these graphs reach their limits. A possible alternative consists in using timed event graphs with multipliers, denoted TEGM's. Indeed, the use of multipliers associated with arcs is natural to model a large number of systems, for example, when the achievement of a specific task requires several units of a same resource, or when an assembly operation requires several units of a same part.

To our knowledge, few works deal with the performance analysis of TEGM's. In fact, in the most of works the proposed solution is to transform the TEGM into an ordinary TEG, which allows the use of well-known methods of performances analysis.

In (Munier, 1993) the initial TEGM is the object of an operation of expansion. Unfortunately, this expansion can lead to a model of significant size, which does not depend only on the initial structure of the TEGM, but also on initial marking. With this method, the system transformation proposed under *single* server semantics hypothesis, or in (Nakamura and Silva, 1999) under *infinite* server semantics hypothesis, leads to a TEG with $|\theta|$ transitions ($|\theta|$ is the 1-norm of the elementary T-semiflow of the corresponding TEGM).

Another linearization method was proposed in (Trouillet et al., 2002) when each elementary circuit of graph contains at least one *normalized* transition (*i.e.,* a transition for which its corresponding elementary T-semiflow component is equal to one). This method increases the number of transitions. Inspired by this work, a linearization method without increasing the number of transition was proposed in (Hamaci et al., 2004).

A calculation method of cycle time of a TEGM is proposed in (Chao et al., 1993) but under restrictive conditions on initial marking.

The weights on the arcs of a TEGM are nonlinearly modelled in (min, +) algebra. Based on works given in (Cohen et al., 1998), we propose a new method of linearization without increasing the number of transition from the graph. The obtained (min, +) linear model allows to evaluate the performance of these graphs. According to initial marking, these performances are evaluated in an exact or approached way.

This article is organized as follows. Some concepts on TEGM's and their functioning are recalled in Section 2. The method of linearization is presented in Section 3. From the equivalent, or approached, TEG of a TEGM, we deduce the cycle time in the Section 4.

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 185–190

2 RECURRENT EQUATIONS OF TEGM's

We assume that the reader is familiar with the structure, firing rules, and basic properties of Petri nets, see (Murata, 1989) for more details.

Consider a Petri net defined as a valued bipartite graph given by a five-tuple (P, T, M, m, τ) in which:

- P and T represent the finite set of *places*, and *transitions* respectively;
- A *multiplier M* is associated with each *arc*. Given $q \in T$ and $p \in P$, the multiplier M_{pq} (respectively, M_{qp}) specifies the weight (in $\mathbb{N}$) of the arc from transition *q* to place *p* (respectively, from place *p* to transition *q*). A zero value for M codes an absence of arc;
- With each place are associated an *initial marking* (m_p assigns an initial number of tokens (in $\mathbb{N}$) in place *P*) and a *holding time* (τ_p gives the minimal time (in $\mathbb{N}$) a token must spend in place *p* before it can contribute to the enabling of its downstream transitions).

We denote by ${}^\bullet q$ (resp., $q^\bullet$) the set of places upstream (resp., downstream) transition q. Similarly, ${}^\bullet p$ (resp., $p^\bullet$) denotes the set of transitions upstream (resp., downstream) place p.

An *event graph* is a Petri net whose each place has exactly one upstream and one downstream transition.

We denote W the incidence matrix of a Petri net. A vector $\theta \in \mathbb{N}^T$ such that $\theta \neq 0$ and $W\theta = 0$ is a T-semiflow. A T-semiflow θ has a minimal support *iff* there exists no other T-semiflow, θ', such that $\{q \in T \mid \theta'(q) > 0\} \subset \{q \in T \mid \theta(q) > 0\}$.

A vector $Y \in \mathbb{N}^P$ such that $Y \neq 0$ et $Y^t W = 0$ is a P-semiflow.

In the rest of the paper we assume that TEGM's are *consistent* (*i.e.*, there exists a T-semiflow θ covering all transitions : $\|\theta\| = T$) and are *conservative* (*i.e.*, there exists a P-semiflow Y covering all places: $\|Y\| = P$).

Remark 1 We disregard without loss of generality *firing times* associated with transitions of a TEG because they can always be transformed into holding times on places (Baccelli et al., 1992, §2.5).

With each transition *q* is associated a *counter variable*, denoted $n_q : n_q$ is an increasing map from $\mathbb{R}$ to $\mathbb{Z} \cup \{+\infty\}$, $t \mapsto n_q(t)$ which denotes the cumulated number of firings of transition *q* up to time *t*.

In the following, we assume that counter variables satisfy the *earliest firing* rule, *i.e.*, a transition q fires as soon as all its upstream places $\{p \in {}^\bullet q\}$ contain enough tokens (M_{qp}) having spent at least τ_p units of time in place p. When the transition q fires, it consumes M_{qp} tokens in each upstream place p and produces $M_{p'q}$ tokens in each downstream place $p' \in q^\bullet$.

Assertion 1 The counter variable n_q of a TEGM (under the earliest firing rule) satisfies the following *transition to transition* equation:

$$n_q(t) = \min_{p \in {}^\bullet q,\, q' \in {}^\bullet p} \lfloor M_{qp}^{-1}(m_p + M_{pq'} n_{q'}(t - \tau_p)) \rfloor. \quad (1)$$

Let us note the presence of inferior integer part to preserve integrity of Eq. (1). In general, a transition q may have several upstream transitions ($\{q' \in {}^{\bullet\bullet} q\}$) which implies that its associated counter variable is given by the *min* of *transition to transition* equations obtained for each upstream transition.

Example 1

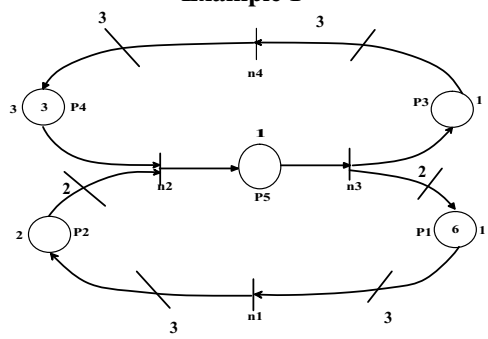


Figure 1: A TEGM.

The counter variables associated with transitions of TEGM depicted in the Figure 1 satisfy the next system equations:

$$\begin{cases} n_1(t) &= \lfloor \frac{6+2n_3(t-1)}{3} \rfloor, \\ n_2(t) &= \min(\lfloor \frac{3n_1(t-2)}{2} \rfloor, 3 + 3n_4(t-3)), \\ n_3(t) &= n_2(t-1), \\ n_4(t) &= \lfloor \frac{n_3(t-1)}{3} \rfloor. \end{cases}$$

In the case of ordinary TEG's, the *transition to transition* equation given in Eq. (1) becomes:

$$x_q(t) = \min_{p \in {}^\bullet q,\, q' \in {}^\bullet p} (m_p + x_{q'}(t - \tau_p)). \quad (2)$$

This equation is linear in the algebraic structure called (*min*, +) algebra. This structure, denoted $\mathbb{Z}_{\min}$, is defined as the set $\mathbb{Z} \cup \{+\infty\}$, equipped with the *min* as additive law (denoted $\oplus$) and with the usual addition as multiplicative law (denoted $\otimes$). The neutral element of the law $\oplus$ (resp., $\otimes$) is denoted $\varepsilon = +\infty$ (resp., $e = 0$). More generally, the (*min*, +) algebra is a *dioid* (Baccelli et al., 1992).

A *dioid* $(\mathcal{D}, \oplus, \otimes)$ is a semiring in which $\oplus$ is *idempotent* ($\forall a,\ a \oplus a = a$). Neutral elements of $\oplus$ and $\otimes$ are denoted ε and *e* respectively.

From the Eq.(2) obtained for each transition, one can express a TEG as the following recursive matrix equation:

$$x(t) = A \otimes x(t-1), \qquad (3)$$

where A is a square matrix with coefficient in $\mathbb{Z}_{\min}$, and $x(t)$ is the vector of the counter variables associated with transitions of the graph. See (Baccelli et al., 1992) for more details on the representation of TEG's in the dioid $\mathbb{Z}_{\min}$.

3 LINEARIZATION OF TEGM'S

A TEGM is *linearizable* if there exists a change of variable $n_q(t) = \theta_q x_q(t)$ such that $x_q(t)$ satisfies a (*min*,+) linear recurrent equation knowing that:

- $n_q(t)$ is the counter associated with transition q of TEGM,
- θ_q is the component of T-semiflow associated with transition q ($\theta_q \in \mathbb{N}^*$).

Proposition 1 A TEGM is *linearizable* if

$$\forall q \in T, \forall p \in^\bullet q, \quad \lfloor \frac{m_p}{M_{qp}} \rfloor \in \theta_q \mathbb{N}. \qquad (4)$$

Proof: According to assertion (1), we have for each transition q of a TEGM:

$n_q(t) = \min_{p \in^\bullet q,\, q' \in^\bullet p} \lfloor M_{qp}^{-1}(m_p + M_{pq'} n_{q'}(t-\tau_p)) \rfloor$.

Using the change of variable $n_q(t) = \theta_q x_q(t)$, and by distributivity of the multiplication with respect to the *min* operator, we have:

$x_q(t) = \min_{p \in^\bullet q,\, q' \in^\bullet p} \frac{1}{\theta_q} \lfloor (\frac{m_p}{M_{qp}} + \frac{M_{pq'}}{M_{qp}} n_{q'}(t-\tau_p)) \rfloor$.

From relation $\frac{\theta_q}{M_{pq'}} = \frac{\theta_{q'}}{M_{qp}}$, obtained for consistent and conservative TEGM (see Munier, 1993), we have

$x_q(t) = \min_{p \in^\bullet q,\, q' \in^\bullet p} \frac{1}{\theta_q} \lfloor (\frac{m_p}{M_{qp}} + \frac{\theta_q}{\theta_{q'}} n_{q'}(t-\tau_p)) \rfloor$,

i.e.,

$x_q(t) = \min_{p \in^\bullet q,\, q' \in^\bullet p} \frac{1}{\theta_q} \lfloor (\frac{m_p}{M_{qp}} + \theta_q x_{q'}(t-\tau_p)) \rfloor$.

Because $\theta_q x_{q'}(t-\tau_p) \in \mathbb{N}$, we finally obtain

$$x_q(t) = \min_{p \in^\bullet q,\, q' \in^\bullet p} (\frac{1}{\theta_q} \lfloor \frac{m_p}{M_{qp}} \rfloor + x_{q'}(t-\tau_p)), \qquad (5)$$

which corresponds to a (*min*,+) linear recurrent equation. ■

More generally, if the condition (4) is satisfied for each transition, the Eq. (1) can be expressed as a (*min*,+) linear recurrent equation.

Remarks:

- Let us define an equivalence class of initial markings for the equivalence relation $m' \equiv m'' \Leftrightarrow \forall q \in T, \forall p \in^\bullet q, \quad \lfloor \frac{m'_p}{M_{qp}} \rfloor = \lfloor \frac{m''_p}{M_{qp}} \rfloor$.

  We can notice that all initial markings of a same equivalence class generate the same firing times behavior of transitions and give the same (*min*,+) model.

- In (Cohen et al., 1998), the authors propose a linearization method through a similar diagonal change of counter variables for fluid TEGMs (*i.e.,* where initial marking and multipliers can take real values). Moreover, they state in Prop. IV.6 that the behavior of a TEGM coincides (in $\mathbb{N}$) with that of its fluid version if $\forall q \in T, \forall p \in^\bullet q, \frac{m_p}{M_{qp}} \in \theta_q \mathbb{N}$. Thus, under this condition, it is possible to linearize a TEGM by considering its fluid version . However, the required condition is more restrictive than the condition (4).

When the condition (4) is not satisfied, we define two linear approximated models of the TEGM by considering a greater (resp., smaller) initial marking.

Definition 1 The upper (resp., lower) linear model is obtained by a minimal addition (resp., removal) of initial tokens in the TEGM, in order to satisfy the linearization condition (4) for each initial marking.

In other words, in each place p for which $\lfloor \frac{m_p}{M_{qp}} \rfloor \notin \theta_q \mathbb{N}$, we add $\overline{m}_p$ (resp., remove $\underline{m}_p$) initial tokens until the linearization condition is checked.

We denote $\overline{x}(t)$ (resp., $\underline{x}(t)$) the state vector of the TEG obtained from the approximate linearization by addition (resp., removal) of tokens in the TEGM.

We have

$$\overline{x}_q(t) = \min_{p \in^\bullet q,\, q' \in^\bullet p} (\frac{1}{\theta_q} \lfloor \frac{(m_p + \overline{m}_p)}{M_{qp}} \rfloor + \overline{x}_{q'}(t-\tau_p)), \qquad (6)$$

where $\overline{m}_p$ is the minimum number of tokens added in the place p such that $\lfloor \frac{m_p + \overline{m}_p}{M_{qp}} \rfloor \in \theta_q \mathbb{N}$.

and

$$\underline{x}_q(t) = \min_{p \in^\bullet q,\, q' \in^\bullet p} (\frac{1}{\theta_q} \lfloor \frac{(m_p - \underline{m}_p)}{M_{qp}} \rfloor + \underline{x}_{q'}(t-\tau_p)), \qquad (7)$$

where $\underline{m}_p$ is the minimum number of tokens removed in the place p such that $\lfloor \frac{m_p - \underline{m}_p}{M_{qp}} \rfloor \in \theta_q \mathbb{N}$.

We have:

$\forall q, \quad \theta_q \underline{x}_q(t) = \underline{n}_q(t) \leq n_q(t) \leq \overline{n}_q(t) = \theta_q \overline{x}_q(t)$.

Example 2 The TEGM depicted in Fig.1 admits the elementary T-semiflow: $\theta = (2, 3, 3, 1)$.

For initial marking M(0)=(6,0,0,3,0), we easily verify that initial marking of each place satisfies the linearization condition (4), which means that TEGM is linearizable.

Using the change of variables $n_i(t) = \theta_i x_i(t)$ and thanks to Eq.(5), we obtain the following linear model:
$$\begin{cases} x_1(t) &= 1 + x_3(t-1), \\ x_2(t) &= \min(1 + x_4(t-3), x_1(t-2)), \\ x_3(t) &= x_2(t-1), \\ x_4(t) &= x_3(t-1). \end{cases}$$
These equations correspond to the TEG depicted in Figure 2.

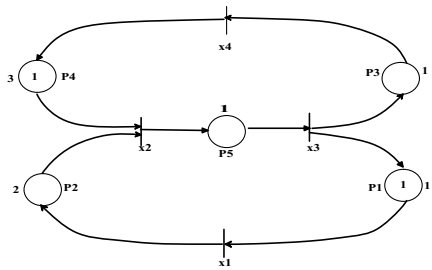


Figure 2: TEG obtained by the linearization of the TEGM of the Figure 1.

For initial marking M(0)=(6,0,0,4,0), we can note that the place P4 does not satisfy the linearization condition.

Thanks to Eqs (6) and (7), we obtain respectively :
$$\begin{cases} \overline{x}_1(t) &= 1 + \overline{x}_3(t-1), \\ \overline{x}_2(t) &= \min(2 + \overline{x}_4(t-3), \overline{x}_1(t-2)), \\ \overline{x}_3(t) &= \overline{x}_2(t-1), \\ \overline{x}_4(t) &= \overline{x}_3(t-1), \end{cases}$$
and
$$\begin{cases} \underline{x}_1(t) &= 1 + \underline{x}_3(t-1), \\ \underline{x}_2(t) &= \min(1 + \underline{x}_4(t-3), \underline{x}_1(t-2)), \\ \underline{x}_3(t) &= \underline{x}_2(t-1), \\ \underline{x}_4(t) &= \underline{x}_3(t-1). \end{cases}$$

The evolution of the counter $n_2(t)$ is depicted in Figure 3 and is such that $\underline{n}_2(t) \leq n_2(t) \leq \overline{n}_2(t)$.

4 PERFORMANCE EVALUATION

4.1 Elements of Performance Evaluation for TEG

We recall main results characterizing an ordinary TEG's modelled in the dioid $\mathbb{Z}_{\min}$ (Baccelli et al., 1992), (Gaubert, 1992).

Definition 2 (Irreducible matrix) A matrix A is said *irreducible* if for any pair *(i,j)*, there is an integer m such that $(A^m)_{ij} \neq \varepsilon$.

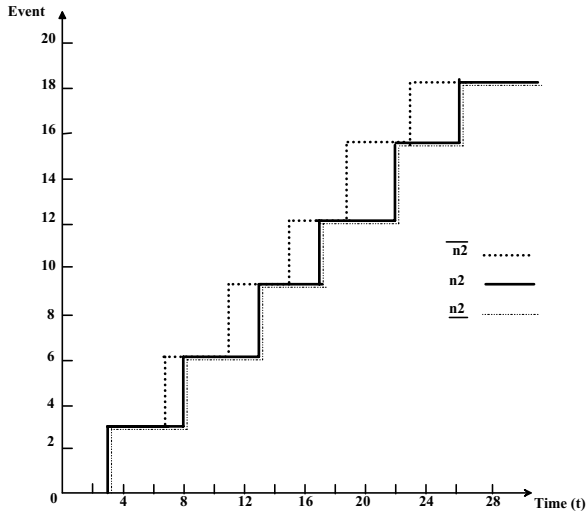


Figure 3: The evolution of the counter variables $\underline{n}_2, n_2$ and $\overline{n}_2$.

Theorem 1 Let A be a square matrix with coefficient in $\mathbb{Z}_{\min}$. The following assertions are equivalent:

- Matrix A is irreducible,
- The TEG associated with matrix A is strongly connected.

One calls *eigenvalue* and *eigenvector* of a matrix A with coefficients in $\mathbb{Z}_{\min}$, the scalar λ and the vector υ such as:
$$A \otimes \upsilon = \lambda \otimes \upsilon.$$

When the initial vector $x(0)$ of matrix Eq. (3) is equal to an eigenvector of matrix A, the TEG reaches a periodic regime from the initial state.

Theorem 2 Let A be a square matrix with coefficients in $\mathbb{Z}_{\min}$. If A is irreducible, or equivalently, if the associated TEG is strongly connected, then there is a single eigenvalue denoted λ. The eigenvalue can be calculated in the following way:
$$\lambda = \bigoplus_{j=1}^{n} (\bigoplus_{i=1}^{n} (A^j)_{ii})^{\frac{1}{j}}. \tag{8}$$

Regarding the TEG, λ corresponds to the firing rate identical for each transition. This eigenvalue λ can be directly deduced from the TEG by
$$\lambda = \min_{c \in C} \frac{M(c)}{T(c)}, \tag{9}$$
where:

- C is the set of elementary circuits of the TEG.
- *T(c)* is the sum of holding times in circuit c.
- *M(c)* is the number of tokens in circuit c.

It is possible that several eigenvectors can be associated with the only eigenvalue of an irreductible matrix.

Definition 3 Let A be an irreducible matrix of eigenvalue λ. One defines the matrix denoted A_λ by

$$A_\lambda = \lambda^{-1} \otimes A.$$

Theorem 3 (Gondran and Minoux, 1977) Let A be an irreducible matrix of eigenvalue λ. The *j*-th column of the matrix A_λ^+, denoted $(A_\lambda^+)_j$, is an eigenvector of A if it satisfies the following equality:

$$(A_\lambda^+)_j = A_\lambda \otimes (A_\lambda^+)_j. \quad (10)$$

4.2 Elements of Performance Evaluation for TEGM's

In the case of TEGM's, the firing rate, denoted λ_{m_q}, is not identical for all transitions. It is defined as follows:

$$\lambda_{m_q} = \frac{\theta_q}{TC_m}, \quad (11)$$

where θ_q is the component of the T-semiflow associated with transition q, and TC_m is average the cycle time of the TEGM.

The average cycle time of a TEGM can be defined as follows :

Definition 4 (Sauer, 2003) The average cycle time, TC_m, of a TEGM is the average time to fire once the T-semiflow under the earliest firing rule (*i.e.*, transitions are fired as soon as possible) from the initial marking.

The firing rate λ_{m_q} of a linearizable TEGM can be calculated from the (*min*, +) linear model by:

$$\lambda_{m_q} = \theta_q \lambda \quad (12)$$

where λ is the eigenvalue of the equivalent (*min*,+) linear model. This result is a direct consequence of the linearization proprety.

In the case where we have an approximate linearization, we obtain

$$\underline{\lambda}_{m_q} \leq \lambda_{m_q} \leq \overline{\lambda}_{m_q},$$

where $\overline{\lambda}_{m_q}$(resp., $\underline{\lambda}_{m_q}$) is the firing rate of the transition q obtained by using the upper (resp., lower) approximated linear model of the TEGM.

When components of the eigenvector, associated with the TEG obtained by linearization, are integer values, the initial conditions vector of TEGM, denoted υ_m (which allow to reach the periodic regime from the initial state) can be deduced by:

$$\upsilon_m = (\theta_1 x_1(0), ..., \theta_n x_n(0)) \quad (13)$$

where $x(0)$ is the eigenvector of the TEG.

Example 3 One determines the firing rate for each transition of the TEGM of Figure 1 from the (*min*, +) linear model.

For $M(0) = (6, 0, 0, 3, 0)$, the initial marking of each place verifies the linearization condition. This TEGM is linearizable.

Thanks to Eq.(8), the production rate of the TEG obtained after linearization is equal to $\frac{1}{5}$. From Eq.(12), we deduce the firing rate of each transition: $\lambda_{m_1} = \frac{2}{5}$, $\lambda_{m_2} = \frac{3}{5}$, $\lambda_{m_3} = \frac{3}{5}$, $\lambda_{m_4} = \frac{1}{5}$.

Thanks to Eq.(11), we deduce that $TC_m = 5$.

For initial marking M(0)=(6,0,0,4,0), we have two linear approximated (*min*, +) models.

In the case where we add two tokens in the place P4 in the TEGM, we obtain a TEG with λ is equal to $\frac{1}{4}$.

Thanks to Eq.(12), we deduce the firing rate of each transition :

$$\overline{\lambda}_{m_1} = \frac{3}{4}, \overline{\lambda}_{m_2} = \frac{2}{4}, \overline{\lambda}_{m_3} = \frac{2}{4}, \overline{\lambda}_{m_4} = \frac{1}{4}.$$

Thanks to Eq. (11), we deduce $\overline{TC}_m = 4$.

In the case where we remove one token in the place P4, we obtain a TEG with λ is equal to $\frac{1}{5}$.

Thanks to Eq.(12), we deduce the firing rate of each transition :

$$\underline{\lambda}_{m_1} = \frac{3}{5}, \underline{\lambda}_{m_2} = \frac{2}{5}, \underline{\lambda}_{m_3} = \frac{2}{5}, \underline{\lambda}_{m_4} = \frac{1}{5}.$$

Thanks to Eq.(11), we deduce $\underline{TC}_m = 5$.

Finally, for M(0)=(6,0,0,4,0), we obtain:

$$4 \leq TC_m \leq 5$$
$$\frac{3}{5} \leq \lambda_{m_1} \leq \frac{3}{4}, \quad \frac{2}{5} \leq \lambda_{m_2} \leq \frac{2}{4}, \quad \frac{2}{5} \leq \lambda_{m_3} \leq \frac{2}{4},$$
$$\frac{1}{5} \leq \lambda_{m_4} \leq \frac{1}{4}.$$

5 CONCLUSION

In order to evaluate the performances of a TEGM from an equivalent TEG, a technique of linearization has been proposed in (*min*,+) algebra. According to initial marking, a linearization condition was stated. The performance analysis of a linearizable TEGM, such as cycle times, is deduced directly from the obtained linear model.

REFERENCES

Baccelli, F., Cohen, G., Olsder, G., and Quadrat, J.-P. (1992). *Synchronization and Linearity: An Algebra for Discrete Event Systems*. Wiley.

Chao, D., Zhou, M., and Wang, D. (1993). Multiple Weighted Marked Graphs. In *IFAC 12th Triennial World Congress*, pages 371–374, Sydney, Australie.

Cohen, G., Gaubert, S., and Quadrat, J.-P. (1998). Timed-event graphs with multipliers and homogeneous min-plus systems. *IEEE TAC*, 43(9):1296–1302.

Gaubert, S. (1992). *Théorie des systèmes linéaires dans les dioïdes*. Thèse de doctorat, Ecole des mines de Paris.

Gondran, M. and Minoux, M. (1977). Valeurs propres et vecteurs propres dans les dioïdes et leur interprétation en thèorie des graphes. volume 2, pages 25–41. EDF, Bullettin de la Direction des Etudes et Recherche, Serie C, Mathématiques informatique.

Hamaci, S., Boimond, J.-L., and Lahaye, S. (2004). On the Linearizability of Discrete Timed Event Graphs with Multipliers Using (min,+) Algebra. In *WODES*, pages 367–372.

Munier, A. (1993). Régime asymptotique optimal d'un graphe d'événements temporisé généralisé : application à un problème d'assemblage. In *RAIPO-APII*, volume 27(5), pages 487–513.

Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. In *Proceedings of the IEEE*, volume 77(4), pages 541–580.

Nakamura, M. and Silva, M. (1999). Cycle Time Computation in Deterministically Timed Weighted Marked Graphs. In *IEEE-ETFA*, pages 1037–1046.

Sauer, N. (2003). Marking Optimization of Weighted Marked Graphs. *Journal of Discrte Event Dynamic Systems*, 13:245–262.

Trouillet, B., Benasser, A., and Gentina, J.-C. (2002). On the Modeling of the Dynamical Behavior of Weighted T-Systems. In *APII-JESA*, volume 36(7), pages 931–944.

MODELING OF MOTOR NEURONAL STRUCTURES VIA TRANSCRANIAL MAGNETIC STIMULATION

Giuseppe d'Aloja, Paolo Lino, Bruno Maione, Alessandro Rizzo
DEE-Dipartimento di Elettrotecnica ed Elettronica – Politecnico di Bari – Via Re David 200 – 70125 Bari Italy
mba-pepp@libero.it, lino@ieee.org, maione@ poliba.it, rizzo@deemail.poliba.it

Keywords: Neuronal Modeling, Spiking Neurons, Brain Waves, TMS.

Abstract: Transcranial Magnetic Stimulation (TMS) of human motor area can evoke different biological waves in the epidural space of patients. These waves can evoke different muscle responses according to different types and amplitudes of stimuli. In this paper we analyze the different types of epidural waves and we propose a neuronal model for the biological structures involved in the experiments.

1 INTRODUCTION

Human nervous system is something very complex and its operation is still rather obscure to scientists. Nevertheless, more and more emerging techniques are helping scientists in examining the human brain in detail and in making hypotheses on its operation. For example, the use of transcranial cerebral stimulations, such as the *Transcranial Magnetic Stimulation* (TMS), allows us to understand some related cerebral mechanisms and identify several cerebral areas. Pioneering studies on brain stimulation through the intact scalp were carried out in the early 80s (Merton and Morton 1980) by stimulating the brain through an electric field. This stimulation technique is called *Transcranial Electrical Stimulation* (TES). Unfortunately, it is known from experience that TES is quite uncomfortable to the patient, because only a small fraction of the applied current flows into the brain through the resistance of the skull and scalp, while the rest travels between the electrodes on the surface, causing local pain and contraction of scalp muscles. The development of TMS (Barker et al., 1985) overcame this discomfort by using a magnetic field to carry the electrical stimulus across the scalp and skull to the brain. Opposite to the TES, TMS is painless and lacking in harmful effects to the human nervous system. TMS has also been exploited with success in the treatment of mental illness and depression (Wasserman, 1998). The first magnetic stimulators were very heavy equipments and they were only able to reach low stimulation frequencies. Recently, novel stimulators with lower weight and smaller size have been designed. The stimulator used in the experiments is the *Magstim 200*® (Jalinous 1997). The magnetic stimulation adopted in our experiment is provided by a 70mm (internal diameter), eight-shaped coil, placed above the cerebral motor area responsible of the left hand movements. Different levels of stimulation have been used, from 20% to 53% of the maximum stimulator output, using a 3% increasing step. The experimental data are collected from patients who have spinal chord stimulators implanted in the epidural space at C1-C2 vertebras for the treatment of intractable dorsolumbar pain (Di Lazzaro, et al., 1998). Two different types of data are available: the recordings from the patient's epidural space and the electromyography (EMG) recordings. The former is necessary for understanding the nature of brain waves; the latter is used for understanding the effects of the voluntary muscle contraction on the recorded muscle potentials. In particular, the effects of voluntary contraction are important at motoneuronal level, but they do not influence the corticospinal volleys, as it will be shown in the following.

The paper is structured as follows: in the next section we analyze the epidural recordings of the biological waves, and the artifacts due to the stimulus and the measurement method. Moreover, we propose a first-attempt linear model. In the third section we exploit the Izhikevich nonlinear neuron model to build a model of the neuronal structure under investigation. In the fourth section we show

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 191–197.

the results. Finally, in the fifth section we draw our conclusions.

2 DATA ANALYSIS

The data analyzed in this paper have been collected in experiments carried out by Prof. V. Di Lazzaro and co-workers at the Neurological Institute at *Cattolica* University in Rome, Italy. The recordings have been collected from a patient with epidural electrodes implanted at C1-C2 vertebras level. The left hand motor area of the patient's brain has been stimulated by TMS. Consequently, brain potentials have been evoked and recorded by a differential amplifier from the epidural electrodes, and by an EMG recorder from the First Dorsal Interosseus muscle (FDI) of the left hand. Experimental data have been recorded with different amplitudes of magnetic stimulation and different levels of voluntary muscle contractions.

Figure 1 shows a typical recording taken at the epidural electrodes. Three different zones can be clearly distinguished:

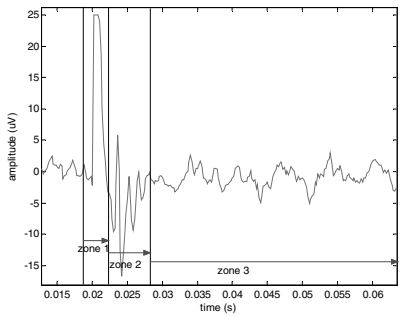


Figure 1: Typical recording taken at the epidural electrodes.

- Zone one: stimulus artifact;
- Zone two: actual biological waves;
- Zone three: noise.

Biological waves evoked by TMS are of two kinds (Di Lazzaro, et al., 2004). The first one, called *D-wave* (Direct wave) is supposed to be produced by direct stimulation of the pyramidal tract axons. The second one is called *I-wave* (Indirect wave) and is supposed to be produced by synaptic activation of the pyramidal neurons of the same tract. With TMS, a D-wave is present only if the stimulus amplitude is over a threshold, whereas I-waves always occur. If a D-wave is present, it precedes the I-waves. In the recorded data, I-waves are numbered according to their temporal sequence. The recordings have been collected using a differential method; therefore, for each volley recorded, two peaks (a positive and a negative one) are present .

Figure 2 shows the amplitude of the I1-wave (computed on the experimental data as the half-peak-to-peak amplitude) for different voluntary muscle contraction at different stimulation levels. As Figure 2 clearly shows, the amplitude of the I1-wave increases linearly with the stimulation level and it is independent of the voluntary contraction level. In fact, muscle contraction increases motoneuronal excitability and has no effect at the corticospinal level. On the other hand, voluntary contraction makes the recordings more noisy and lowers the signal to noise ratio.

In our recordings there is always a saturated peak which occurs at the same instant (0.02s) of application of the magnetic stimulus. This saturated peak is biologically implausible, and systematically occurs in every experimental recording. Thus, we can conclude that this is a bias caused by the stimulus, due to both the electromagnetic coupling and the displacement current (O'Keeffe et al., 2001), (McLean et al., 1996). To analyze the actual biological waves we have reconstructed the bias for different stimulation amplitudes. In particular we have developed 4 different bias models according to the stimulus amplitude. Figure 3 shows, in clockwise direction from top-left, the reconstructed biases from low to high stimulation intensity. In our approach, to model the experimental biological waves the reconstructed bias is subtracted from the experimental data. Subsequently, the bias is added again to the modeled waves to recover the modeled signal.

For the experiment performed with this patient the amplitudes of subsequent I-waves are well

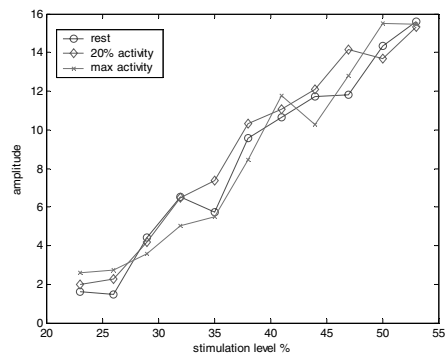


Figure 2: Amplitude of the first I wave evoked by TMS for different muscle contractions levels.

modeled by an exponential decreasing law. As stated before, the amplitude of I1-wave increases almost linearly with the stimulus amplitude. Therefore, a first-attempt model of the measured brain waves has been developed by considering a second order linear system, described by the following transfer function:

The parameters *a, b, rit* of the second order transfer function have been identified with a least square method. For example, for the stimulus amplitude at 20% of the maximal amplitude, we obtained a=275, b=2.25·10^6, rit=2·10^{-2}. It is important to notice that these parameters are not physically related with the biological phenomena occurring (except for the delay *rit* which is related to the stimulus application time). They simply are the parameters which guarantee the best fit with experimental data, considering the second order linear model above. The Laplace transform of the input stimulus, as the monophasic current produced by the eight-shaped coil (Kammer et al., 2001), is:

where K is the stimulus amplitude, and τ is a time constant. In order to represent the stimulus correctly, τ ranges from 4·10^{-4} s to 8·10^{-4} s, increasing linearly with the stimulus amplitude.

$$G(s) = b \cdot e^{-s \cdot rit} \cdot \frac{1}{s^2 + a \cdot s + b}$$

$$i(s) = \frac{K}{\tau} \cdot \frac{1}{s^2 + \frac{3s}{\tau} + \frac{2}{\tau^2}}$$

Figures 3 and 4 show some results achieved with the linear model described above. This model gave good

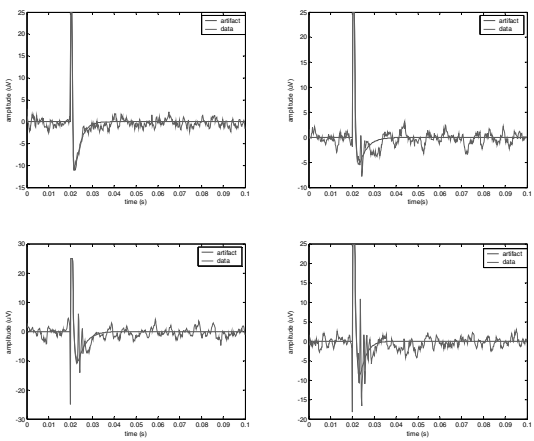


Figure 3: Reconstructed stimulus artifacts versus experimental data for different stimulation levels.

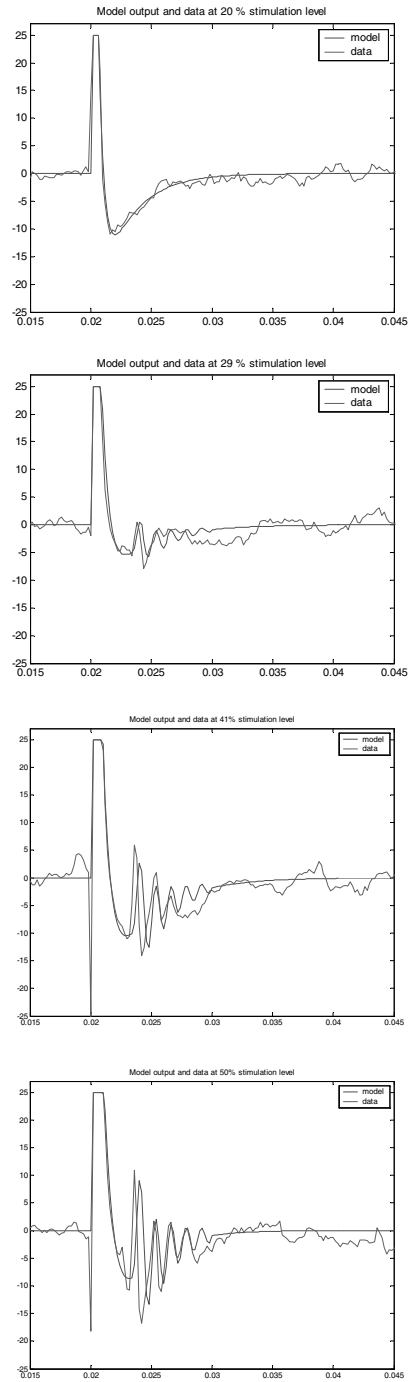


Figure 4: Model output and data for different stimulation level.

results for this experiment but is not suitable for experimental data collected in other patients, nor for other similar experiments reported in literature (Houlden, et al., 1999). In fact, the use of a linear model implies the periodicity of I-waves. An

in-depth analysis on the latency of the I-waves shows that in fact they are not periodic and each wave has a fixed latency for all the stimulation levels. We remind that the recordings are the results of different mechanisms: the stimulus artifact, the artifact due to the propagation of the nervous potentials through the fibers and the artifact due to the differential measurement method. Therefore, the aspect of the recordings is not entirely due to the action potentials generating in the fiber, and only amplitude and latency of I-waves can be considered as biologically plausible, and useful, data.

Therefore, we have developed another model based on a neuronal network of spiking neurons. The facts on which we base our hypotheses is that the potential recorded at the electrodes comes from the output of a huge number of spinal fibers, and the greater the stimulation amplitude is, the higher the number of stimulated fibers is. This hypothesis is supported by the biological law of "nothing or all" which states that neurons produce a fixed voltage level when they are excited above a threshold. If the stimulation is under the threshold the action potential is not generated and, correspondently, a descending wave at the electrodes is not revealed.

3 NEURONAL MODELS

The neuronal network developed in this section consists of Izhikevich spiking neurons (Izhikevich, 2003). It is described by the equation system:

$$v' = 0.04v^2 + 5v + 140 - u + I$$
$$u' = a(bv - u)$$

with the reset condition:

$$\text{If } v \geq 30 \text{ mV then } v \leftarrow c$$
$$u \leftarrow u + d$$

where v is the membrane potential, u is a recovery variable which considers the refractory period and the K^+ current activation after the action potential. The mechanism of hyper polarization of the cell membrane is considered by the c parameter which has the -64 mV value. We can now analyze the meaning of the parameters.

- a describes the time scale of the recovery variable u. Smaller values result in slower recovery;
- b describes the sensitivity of the recovery variable u to the fluctuations of the membrane potential v;
- c describes the after-spike reset value of the membrane potential v;
- d describes after-spike reset of the recovery variable u.

The parameters of the neuron model have been fixed to: $a = 0.4$, $b = 0.26$, $c = -64$, $d = 6$. This choice makes the neuron spiking with a latency comparable to that of the experimental recordings.

Since the experimental recordings cannot reveal the action potentials of the single neuron activated by the stimulation, we have simulated the global behavior of neuronal networks, corresponding to different areas of the motor cortex, consequent to appropriate current intensities induced by the eight-shaped coil.

The amplitude of each I-wave is proportional to the number of corticospinal neurons transinaptically activated by the stimulation. The generation of a D-wave is due to the direct stimulation of the corticospinal neurons for high stimulation levels, as the inducted current activates the deep brainstem and the cortical neurons directly. Nevertheless, for the generation of the I-waves, the number of neurons actually involved is unknown. However, assuming that each stimulated neuron contributes to the formation of the I-wave with a 1μV spike and consequently we have estimated the number of neurons involved in the stimulation process.

Based on these considerations, we have simulated a 500 cortical neurons network connected to a 100 corticospinal neurons network. Both networks are considered within a regular topology. Each corticospinal neuron is synaptically connected to five cortical neurons. Figure 5 shows that an eight-shaped coil induces an electric field with the highest peaks located in three main areas: one located immediately below the coil with the maximum intensity, the other two on the two sides of the coil, with a peak of intensity which is about a half of the highest one. The hypothesis made in this paper is that the electric field mainly stimulates groups of neurons located under the

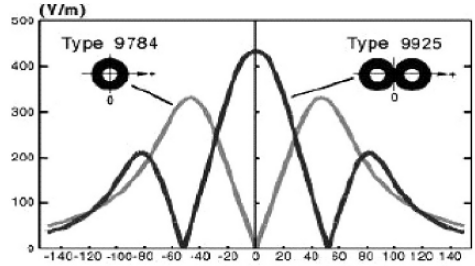


Figure 5: Electric field shape for circular and eight-shaped stimulation coils.

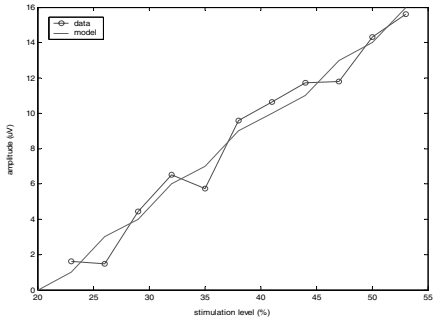


Figure 6: amplitude of the I1 waves for different stimulation levels.

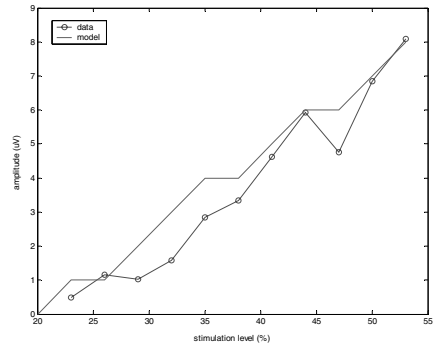


Figure 7: Amplitude of the I2 waves for different stimulation levels.

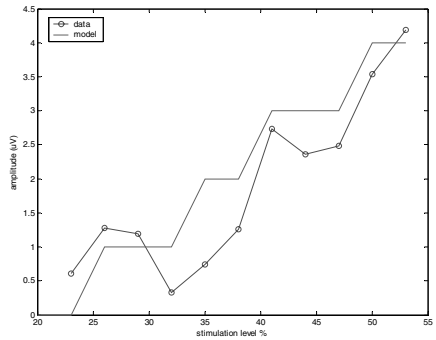


Figure 8: Amplitude of the I3 waves for different stimulation levels.

highest field peaks (Rosler, 2001 – Sakay). Therefore, an I-wave consists of the sum of the outputs of many neurons which fires at the same time, because they are essentially stimulated by the same field. This hypothesis is supported by the following facts, confirmed by the experimental data:

- In these experiments a maximum of three waves is generated, and there are three main areas in which a peak of electric field exists.
- For high intensities, the field peaks are higher and more spread in space. Consequently, more neurons are activated and the correspondent I-wave is larger.
- For low intensities, the electric field has only one peak located under the coil. Correspondently, only one I-wave is generated for low intensity field.

Therefore, the cortical network (and consequently the corticospinal one) has been partitioned in three areas, each responsible for the generation of one of the three I-waves. When the stimulation intensity increases, the number of activated neurons increases and larger waves are produced. This simulates the spatial spreading of the stimulus at higher intensity of stimulation. Therefore, different I-waves are generated because a different current for each neuronal area is inducted by the magnetic field. Figures 6, 7 and 8 show a comparison between the amplitude of simulated I1, I2 and I3-waves and the experimental ones, versus the stimulus intensity. Once amplitudes and latencies have been modeled, the signal shape must be reconstructed. We already dealt with the fact that the differential measurement configuration introduces an artifact in the measurements, producing a sequence of one positive and one negative volley for each cerebral I-wave.

The propagation velocity of the impulse has been calculated in about 50 m/s. The propagation delay for the I1-wave is about 2.2 ms. For I-waves, due to their synaptic nature, an approximately 1 ms delay due to the synaptic mechanism must be added. Therefore, a total latency for the I1-wave equal to 3.6 ms has been reckoned, which is coherent with the distance of 12 cm between stimulation and recording site.

Therefore, taking into account the propagation velocity of the waves and the distance between the electrodes, the artifact can be reconstructed.

4 MODEL VALIDATION

With the neuronal structure described above, a good fitting of the experimental data has been obtained for all the stimulation levels.

To fit the experimental data we have reproduced the stimulus artifact, the measure setup and the propagation artifacts.

Figures 9 and 10 compare the output of the model with the experimental data. They clearly show that the neuronal network gives better results than the linear model. It respects the aperiodicity of

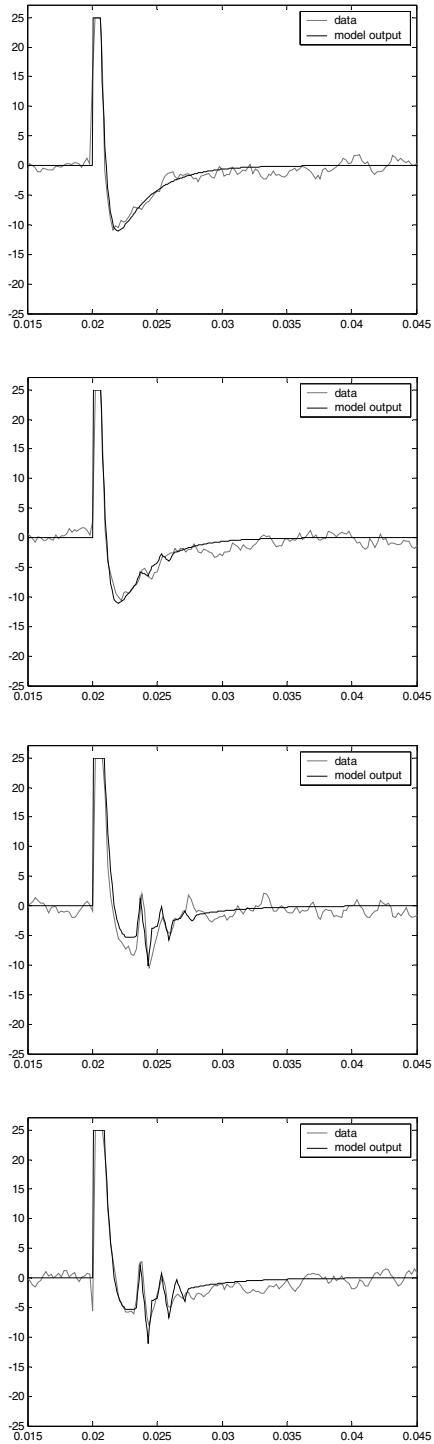


Figure 9: Model output and data for 20%, 23%, 32%, 35%, 44% stimulation level.

the response, taking into account the different latencies of I-waves, and provides a better fitting for wave amplitudes.

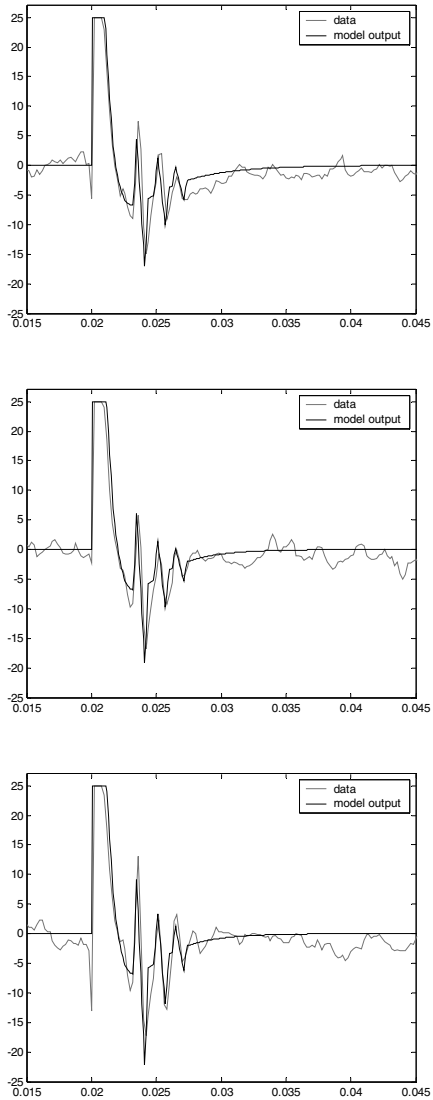


Figure 10: Model output and data for 47%, 53% stimulation level.

5 CONCLUSIONS

In this paper a model of motor neuronal structures has been built and validated on the basis of experimental recordings obtained via Transcranial Magnetic Stimulation (TMS). With this technique, the brain of the patient is stimulated by a suitable magnetic field placed above the cerebral motor area responsible of the left hand movements. The stimulation evokes different biological waves in the brain which are transmitted from the motor cortex, through the pyramidal neurons via synaptic

connection, to the spinal chord, where signals are collected by a couple of electrodes implanted in the epidural space at C1-C2 vertebras level.

After a thorough data analysis phase, the motor neuronal structure has been modeled by a neural network based on Izhikevich neurons, for both the motor cortex and the pyramidal neuron areas. Moreover, stimulus and measurement artifacts have been reconstructed and considered in the modeling phase. The results are fully satisfactory, model output and experimental recordings match for each available experiment.

Further research will involve a more accurate modeling of the motor cortex and its connections with the pyramidal tracts. At present, a hypothesis of a five-to-one local connection between cortex and pyramidal neurons has been made. In the future, optimization strategies will be considered to find an adequate connection scheme between cortex and pyramidal tracts, and with different topologies, involving also the plasticity mechanism (i.e. time-variant connections). Moreover, the model is being validated on several recordings coming from different patients, with different stimulation protocols.

ACKNOWLEDGEMENTS

This work is supported by the national research project MIUR "Innovative Bio-Inspired Strategies for the Control of Motion Systems", No.: 2003090328, 2003. The authors acknowledge Prof. Di Lazzaro (Institute of Neurology, Università Cattolica, Rome, Italy), Prof. Mazzone (Neurochirurgia CTO, Rome, Italy), Dr. Ghirlanda (Department of Psychology, University of Bologna, Italy) and their research groups for having provided the experimental data and the prior knowledge for an in-depth data analysis.

REFERENCES

Merton P.A., Morton H.B. (1980). Stimulation of the cerebral cortex in the intact human subject. *Nature* 285:227.

Barker A.T., Jalinous R., Freeston I.L. (1985). Non-invasive magnetic stimulation of human motor cortex, *Lancet* , i pp. 1106–1107.

Jalinous R. 1997. Guide to Magnetic Stimulation, Magstim Company.

Di LazzaroV., Restuccia D., Oliviero A., Proficue P., Ferrara L., Insola A., Mazzone P., Tonali P., Rothwell J.C. (1998). Effects of voluntary contraction on descending volleys evoked by transcranial stimulation in conscious humans, *Journal of Physiology*, *508(2)*, pp. 625–633.

Di Lazzaro V., Oliviero A., Pilato F., Saturno E., Di leone M., Mazzone P., Insola A., Tonali P. A., Rothwell J.C. (2004). The physiological basis of transcranial motor cortex stimulation in conscious humans, *Clinical Neuropysiology,* 115 pp. 255–266.

Kammer T., Beck S., Thielscher A., Ulrike Laubis-Herrmann, Helge Topka. (2001). Motor thresholds in humans: a transcranial magnetic stimulation study comparing different pulse waveforms, current directions and stimulator types, *Clinical Neurophysiology*, *112,* pp. 250–258.

Houlden D. A., et al. (1999, March 1). Spinal Cord-Evoked Potentials and Muscle Responses Evoked by Transcranial Magnetic Stimulation in 10 Awake Human Subjects, *The Journal of Neuroscience, 19(5),* pp. 1855–1862.

Izhikevich E. M. (2003). Simple Model of Spiking Neurons *IEEE Transactions on Neural Networks, Vol. 14*, *No. 6.*

Rosler K. M. (2001). Transcranial Magnetic Brain Stimulation: a Tool to Investigate Central Motor Pathways. *News Phisiol. Sci. Vol. 16,* pp. 297–302

Sakay K., Ugawa Y., Kanazawa I., Preferential activation of different I waves by focal magnetic stimulation with different current direction, *TMS PS*-10-8.

O'Keeffe D. T., Lyons G. M., Donnelly A. E., Byrne C. A. (2001). Stimulus artifact removal using a software-based two-stage peak detection algorithm, *Journal of Neuroscience Methods, 109,* pp. 137–145.

McLean L., MScEE, P. T., Scott R.N. (1996, December). Stimulus Artifact Reduction in Evoked Potential Measurements, *Arch Phys Med Rehabil, Vol 77*.

Wassermann E.M. (1998). Risk and safety of repetitive transcranial magnetic stimulation: report and suggested guidelines from the international workshop on the safety of repetitive transcranial magnetic stimulation, *Electroencephalography and Clinical Neurophysiology, 108*, pp. 1–16.

ANALYSIS AND SYNTHESIS OF DIGITAL STRUCTURE BY MATRIX METHOD

B. Psenicka
Universidad Nacional Autonoma de México
pseboh@servidor.unam.mx

R. Bustamante Bello
TEC de Monterey,Campus Ciudad de México
rbustama@itesm.mx

M. A. Rodriguez
Universidad Politécnica de Valencia
marodrig@upvnet.upv.es

Keywords: Synthesis, Analysis, Digital structures, Algorithm, Matrix Method.

Abstract: This paper presents a general matrix algorithm for analysis and synthesis of digital filters. A useful method for computing the state-space matrix of a general digital network and a new technique for the design of digital filters are shown by means of examples. The method proposed in this paper allows the analysis of the digital filters and the construction of new equivalent structures of the canonic and non canonic digital filter forms. Equivalent filters with different structures can be found according to various matrix expansions. The procedure proposed in this paper is more efficient and economic than traditional methods because it permits to construct circuits with a minimum of shifting operations.

1 INTRODUCTION

The digital system presented in Figure 1 is described by the following eqs.:

$$\begin{aligned}\mathbf{Y}(z) = &\ \mathbf{F}_{YX}\mathbf{X}(z) + \mathbf{F}_{YU}\mathbf{U}(z) + \mathbf{F}_{YV}\mathbf{V}(z)\\ \mathbf{U}(z) = &\ \mathbf{F}_{UX}\mathbf{X}(z) + \mathbf{F}_{UU}\mathbf{U}(z) + \mathbf{F}_{UV}\mathbf{V}(z)\\ \mathbf{V}(z) = &\ \mathbf{F}_{VX}\mathbf{X}(z) + \mathbf{F}_{VU}\mathbf{U}(z) + \mathbf{F}_{VV}\mathbf{V}(z)\end{aligned} \tag{1}$$

or in matrix form (2), (Luecker, 1976)

$$\mathbf{N}_s \times \begin{bmatrix} \mathbf{X}(z) \\ \mathbf{Y}(z) \\ \mathbf{U}(z) \\ \mathbf{V}(z) \end{bmatrix} = \mathbf{0}, \tag{2}$$

where $\mathbf{N}_s$ in Eq. (2) is the signal flow matrix that represents the signal-flow graph of the digital system with multiple inputs and multiple outputs, **X(z)** is the vector of the input signals X_i, **Y(z)** is the vector of the output signals Y_i, **U(z)** is the vector of the signals U_i in the output of the delay elements and **V(z)** is a vector of the signals V_i in the output of the adders, see Figure 1. $\mathbf{N_s}$ can be obtained by expression (3), where $\mathbf{F_{YX}}$ in the Eq. (3) is the transfer matrix output/input, $\mathbf{F_{YX}} = \mathbf{Y(z)}/\mathbf{X(z)}$ if **U(z)=V(z)**=0. In Figure 2, signals U_3 and U_4 represent the outputs of the delay elements and signals V_5 and V_6 designate the outputs of the summers.

$$\mathbf{N}_s = \left[\begin{array}{c|c|c|c} \mathbf{F}_{(YX)} & -\mathbf{E} & \mathbf{F}_{(YU)} & \mathbf{F}_{(YV)} \\ \hline \mathbf{F}_{(UX)} & \mathbf{0} & \mathbf{F}_{(UU)} - \mathbf{E} & \mathbf{F}_{(UV)} \\ \hline \mathbf{F}_{(VX)} & \mathbf{0} & \mathbf{F}_{(VU)} & \mathbf{F}_{(VV)} - \mathbf{E} \end{array}\right] \tag{3}$$

If we reduce the signals avoiding the outputs of the adders V_i in expression (2), we obtain

$$\mathbf{N}_e \times \begin{bmatrix} \mathbf{X}(z) \\ \mathbf{Y}(z) \\ \mathbf{U}(z) \end{bmatrix} = \mathbf{0}, \tag{4}$$

where $\mathbf{N}_e$ in Eq. (4) is a flow-state matrix and the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ in the matrix Eq. (5) are the state matrices of the digital system.

$$\mathbf{N}_e = \left[\begin{array}{c|c|c} \mathbf{D} & -\mathbf{E} & \mathbf{C} \\ \hline z^{-1}\mathbf{B} & \mathbf{0} & z^{-1}\mathbf{A} - \mathbf{E} \end{array}\right] \tag{5}$$

In the flow-state matrix, the matrices $\mathbf{E}$ and $\mathbf{0}$ are identity and zero matrices respectively. If we reduce the matrix Eq. (4), not taking into account the vector of the signals U_i, we get the expression

$$\mathbf{N}_t^{(2)} \times \begin{bmatrix} \mathbf{X}(z) \\ \mathbf{Y}(z) \end{bmatrix} = \mathbf{0}, \tag{6}$$

where the transfer matrix $\mathbf{N}_t^{(2)}$ can be defined by Eq. (7)

$$\mathbf{N}_t^{(2)} = \left[\ \mathbf{D} + \mathbf{C} \times (z\mathbf{E} - \mathbf{A})^{-1}\,\mathbf{B};\quad -\mathbf{E}\ \right] \tag{7}$$

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 199–206

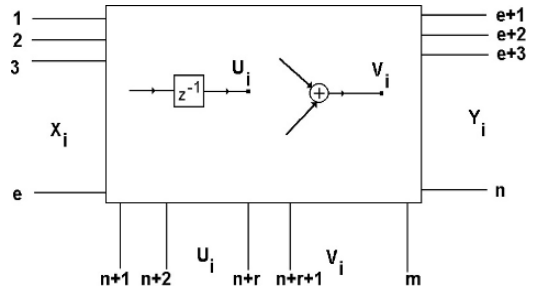


Figure 1: State-space structure with multiple inputs and outputs.

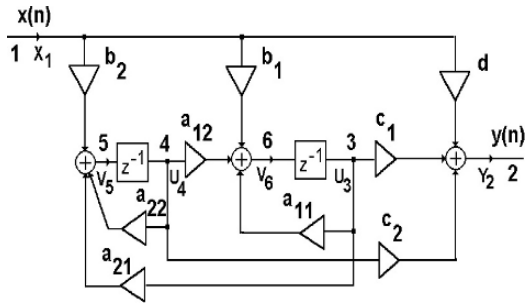


Figure 2: State-space digital filter of second order.

The element $n_{21}^{(2)}$ of the transfer matrix $\mathbf{N}_t^{(2)}$ is the transfer function H(z) of the digital network.

$$n_{21}^{(2)} = H(z) = \mathbf{D} + \mathbf{C} \times (z\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} \qquad (8)$$

Using the inverse **z** transform, the impulse response of the circuit results (Luecker, 1976)

$$h(n) = \begin{cases} \mathbf{D} & for\ \ l = 0 \\ \\ \mathbf{C}\mathbf{A}^{l-1}\mathbf{B} & for\ \ l > 0 \end{cases} \qquad (9)$$

where l=1,2,3 ...

2 ANALYSIS OF THE SECOND ORDER STATE-SPACE DIGITAL FILTER

As an example, we will determine the transfer function of a state-space second order digital filter, see Figure 2. To calculate the signal-flow matrix of the digital filter, previously it is necessary to mark the input, output and state nodes (Pšenička and Herrera, 1997), (Pšenička and Ugalde, 1999). The input node is denoted by number 1 and the output node by number 2. The nodes 3 and 4 are assigned to the outputs of the delay elements. Finally, the nodes 5 and 6 are placed on the output of the adders. The system Eq. (10) for each node can be obtained from Figure 2.

$$\begin{array}{ll} 2: Y_2 = & X_1 d + U_3 c_1 + U_4 c_2 \\ 3: U_3 = & V_6 z^{-1} \\ 4: U_4 = & V_5 z^{-1} \\ 5: V_5 = & X_1 b_2 + U_3 a_{21} + U_4 a_{22} \\ 6: V_6 = & X_1 b_1 + U_3 a_{11} + U_4 a_{12} \end{array} \qquad (10)$$

The Eq. (10) can be written by the matrix equation to generate the signal flow matrix $\mathbf{N}_s^{(6)}$. The first row of the matrix (11) is indexed by number 2, see also expression (10).

$$\mathbf{N}_s^{(6)} = \begin{array}{c} \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} d & -1 & c_1 & c_2 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & z^{-1} \\ 0 & 0 & 0 & -1 & z^{-1} & 0 \\ b_2 & 0 & a_{21} & a_{22} & -1 & 0 \\ b_1 & 0 & a_{11} & a_{12} & 0 & -1 \end{bmatrix} \end{array} \qquad (11)$$

From the signal-flow matrix (11)we can observe, that the main diagonal contain -1's and in the second column all the elements are zeros except the first one.
The signal-flow matrix can be formed directly without writing node Eq. (10). For example, due to the delay element placed between the nodes 6 and 3, see Figure 2, the matrix element $n_{36}^{(6)}$ of the matrix $\mathbf{N}_s^{(6)}$ acquires the value z^{-1}. The multiplier a_{21} located between the nodes 3 and 5 is represented in the matrix $\mathbf{N_s^{(6)}}$ by the element $n_{53}^{(6)}$ equal to the constant a_{21}. Similarly, we can obtain all of the elements in the signal-flow matrix without writing the nodal equations. The matrix $\mathbf{N}_s^{(6)}$ can be reduced to the matrix $\mathbf{N}^{(5)}$ Eq. (13) according to the expression

$$n_{ij}^{(k-1)} = \frac{n_{ij}^{(k)} n_{kk}^{(k)} - n_{ik}^{(k)} n_{kj}^{(k)}}{n_{kk}^{(k)}}, \qquad (12)$$

where i represents the number of the row, j the number of the column and k the degree of the matrix.
Following the rule of reduction (12), we obtain the matrix $\mathbf{N}^{(5)}$ and the state-flow matrix $\mathbf{N}_e^{(4)}$

$$\mathbf{N}^{(5)} = \begin{bmatrix} d & -1 & c_1 & c_2 & 0 \\ z^{-1}b_1 & 0 & -1 + a_{11}z^{-1} & a_{12}z^{-1} & 0 \\ 0 & 0 & 0 & -1 & z^{-1} \\ b_2 & 0 & a_{21} & a_{22} & -1 \end{bmatrix} \qquad (13)$$

$$\mathbf{N}_e^{(4)} = \left[\begin{array}{c|c|cc} d & -1 & c_1 & c_2 \\ \hline z^{-1}b_1 & 0 & -1+a_{11}z^{-1} & a_{12}z^{-1} \\ z^{-1}b_2 & 0 & a_{21}z^{-1} & -1+a_{22}z^{-1} \end{array}\right] \tag{14}$$

If we compare the expressions (14) and (5), we obtain the state matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ of the state-space digital filter.

$$\mathbf{D} = d \qquad \mathbf{C} = \left[\begin{array}{cc} c_1 & c_2 \end{array}\right]$$
$$\mathbf{B} = \left[\begin{array}{c} b_1 \\ b_2 \end{array}\right] \quad \mathbf{A} = \left[\begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array}\right] \tag{15}$$

3 DESIGN OF THE THIRD ORDER STATE-SPACE STRUCTURE

In this example we are going to obtain the third order state-space structure. The state matrices of the third order state-space filter have the general form (16), (Psenicka et al., 1998).

$$\mathbf{D} = d \qquad \mathbf{C} = \left[\begin{array}{ccc} c_1 & c_2 & c_3 \end{array}\right]$$
$$\mathbf{B} = \left[\begin{array}{c} b_1 \\ b_2 \\ b_3 \end{array}\right] \quad \mathbf{A} = \left[\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array}\right] \tag{16}$$

Substituting (16) in (5) it is obtained the state-flow matrix (17)

$$\mathbf{N}_e^{(5)} = \left[\begin{array}{cccc} d & -1 & c_1 & c_2 & c_3 \\ z^{-1}b_1 & 0 & -1+a_{11}z^{-1} & a_{12}z^{-1} & a_{13}z^{-1} \\ z^{-1}b_2 & 0 & a_{21}z^{-1} & -1+a_{22}z^{-1} & a_{23}z^{-1} \\ z^{-1}b_3 & 0 & a_{31}z^{-1} & a_{32}z^{-1} & -1+a_{33}z^{-1} \end{array}\right] \tag{17}$$

To expand the state-flow matrix (17) which contains five columns and four rows in the matrix with six columns and five rows (19) , we use the Eq. (18). The Eq. (18) is obtained from Eq. (12) for $n_{kk}^{(k)} = -1$.

$$n_{ij}^{(k)} = n_{ij}^{(k-1)} - n_{ik}^{(k)} n_{kj}^{(k)} \tag{18}$$

If we choose the elements of the new matrix $n_{26}^{(6)} = n_{46}^{(6)} = n_{56}^{(6)} = 0$, then the first, third and fourth rows in the new matrix $\mathbf{N}^{(6)}$ remain unchanged (19).

$$\mathbf{N}^{(6)} = \left[\begin{array}{cccccc} d & -1 & c_1 & c_2 & c_3 & 0 \\ n_{31}^{(6)} & n_{32}^{(6)} & n_{33}^{(6)} & n_{34}^{(6)} & n_{35}^{(6)} & n_{36}^{(6)} \\ z^{-1}b_2 & 0 & a_{21}z^{-1} & -1+a_{22}z^{-1} & a_{23}z^{-1} & 0 \\ z^{-1}b_3 & 0 & a_{31}z^{-1} & a_{32}z^{-1} & 1-a_{33}z^{-1} & 0 \\ n_{61}^{(6)} & n_{62}^{(6)} & n_{63}^{(6)} & n_{64}^{(6)} & n_{65}^{(6)} & n_{66}^{(6)} \end{array}\right] \tag{19}$$

The elements of the matrix (19), $n_{61}^{(6)}$, $n_{62}^{(6)}$, $n_{63}^{(6)}$, $n_{64}^{(6)}, n_{65}^{(6)}$, $n_{66}^{(6)}$ and $n_{36}^{(6)}$ can be chosen and the remaining elements $n_{31}^{(6)}$, $n_{32}^{(6)}, n_{33}^{(6)}$, $n_{34}^{(6)}$ and $n_{35}^{(6)}$ are obtained by means of the Eq. (18). The elements in the last row and columns of the matrix $\mathbf{N^{(6)}}$ must be chosen, in order to obtain in the second row of the matrix $\mathbf{N^{(6)}}$ plenty of zeros. It is suitable to choose the element $n_{36}^{(6)} = z^{-1}$, because all elements in the second row of the matrix $N_e^{(5)}$ contain z^{-1}. But it is possible select the element $n_{36}^{(6)}$ in a different way, as we shall see in section 4.2. If we choose

$$\begin{array}{llll} n_{26}^{(6)} = 0 & n_{46}^{(6)} = 0 & n_{56}^{(6)} = 0 & n_{62}^{(6)} = 0 \\ n_{66}^{(6)} = -1 & n_{65}^{(6)} = a_{13} & n_{64}^{(6)} = a_{12} & n_{63}^{(6)} = a_{11} \\ n_{36}^{(6)} = z^{-1} & n_{61}^{(6)} = b_1 & & \end{array}$$

then we get by Eq. 18 the elements of the new matrix in the form

$$\begin{array}{ll} n_{31}^{(6)} = n_{31}^{(5)} - n_{36}^{(6)} n_{61}^{(6)} = z^{-1}b_1 - z^{-1}b_1 & = 0 \\ n_{32}^{(6)} = n_{32}^{(5)} - n_{36}^{(6)} n_{62}^{(6)} = 0 - z^{-1}0 & = 0 \\ n_{33}^{(6)} = n_{33}^{(5)} - n_{36}^{(6)} n_{63}^{(6)} = -1 + a_{11}z^{-1} - a_{11}z^{-1} & = -1 \\ n_{34}^{(6)} = n_{34}^{(5)} - n_{36}^{(6)} n_{64}^{(6)} = z^{-1}a_{12} - z^{-1}a_{12} & = 0 \\ n_{35}^{(6)} = n_{35}^{(5)} - n_{36}^{(6)} n_{65}^{(6)} = z^{-1}a_{13} - z^{-1}a_{13} & = 0 \end{array}$$

and we obtain the matrix $\mathbf{N}^{(6)}$

$$\mathbf{N}^{(6)} = \left[\begin{array}{cccccc} d & -1 & c_1 & c_2 & c_3 & 0 \\ 0 & 0 & -1 & 0 & 0 & z^{-1} \\ z^{-1}b_2 & 0 & a_{21}z^{-1} & -1+a_{22}z^{-1} & a_{23}z^{-1} & 0 \\ z^{-1}b_3 & 0 & a_{31}z^{-1} & a_{32}z^{-1} & -1+a_{33}z^{-1} & 0 \\ b_1 & 0 & a_{11} & a_{12} & a_{13} & -1 \end{array}\right] \tag{20}$$

Similarly, we can obtain the matrices $\mathbf{N^{(7)}}$ and $\mathbf{N^{(8)}}$. After a very simple calculation, we can get the matrix (21) and the signal-flow matrix (22). For example it is advantageous to choose the element $n_{47}^{(7)} = z^{-1}$, in the matrix $\mathbf{N^{(7)}}$, because each element in row 3 of the matrix $\mathbf{N^{(6)}}$ contains z^{-1}. In case the matrix element

$n_{71}^{(7)}$ equal to b_2 is chosen, the element $n_{41}^{(7)}$ is equal to zero, marked by $|0|$. For example in order to obtain in the matrix $N^{(7)}$ $n_{41}^{(7)} = 0$ if $n_{41}^{(6)} = b_2 \cdot z^{-1}$ it is necessary to choose $n_{47}^{(7)} = z^{-1}$ and $n_{71}^{(7)} = b_2$ or viceversa. So to obtain in the matrix $N^{(8)}$ $n_{55}^{(8)} = -1$ if $n_{55}^{(7)} = -1 + a_{33} \cdot z^{-1}$ it is necessary to choose $n_{58}^{(8)} = z^{-1}$ and $n_{85}^{(8)} = a_{33}$ or the contrary. The same procedure can be applied to Eq. (21) in order to get Eq. (22).

$$\mathbf{N}^{(7)} = \begin{bmatrix} d & -1 & c_1 & c_2 & c_3 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & z^{-1} & 0 \\ |0| & 0 & 0 & -1 & 0 & 0 & |z^{-1}| \\ z^{-1}b_3 & 0 & a_{31}z^{-1} & a_{32}z^{-1} & -1+a_{33}z^{-1} & 0 & 0 \\ b_1 & 0 & a_{11} & a_{12} & a_{13} & -1 & 0 \\ |b_2| & 0 & a_{21} & a_{22} & a_{23} & 0 & -1 \end{bmatrix} \tag{21}$$

$$\mathbf{N}^{(8)} = \begin{bmatrix} d & -1 & c_1 & c_2 & c_3 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & z^{-1} & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & z^{-1} & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & z^{-1} \\ b_1 & 0 & a_{11} & a_{12} & a_{13} & -1 & 0 & 0 \\ b_2 & 0 & a_{21} & a_{22} & a_{23} & 0 & -1 & 0 \\ b_3 & 0 & a_{31} & a_{32} & a_{33} & 0 & 0 & -1 \end{bmatrix} \tag{22}$$

The digital structure that corresponds to the signal flow matrix $\mathbf{N}^{(8)}$ is presented in Figure 3.

The second canonic form of the state-space digital filter can be obtained from the structure presented in Figure 3. Changing the adders to nodes, the nodes to adders, the input to output and the directions of the multipliers, the second canonic form of the state-space filter can be obtained. If other values are chosen for elements in the last row and the last column in matrices (19), (20) and (21) we can obtain other equivalent structure.

4 EXAMPLES

4.1 Design of the Filter from the State Space Matrices

In the first example we shall demonstrate how to derive the structures of the state-space filter without

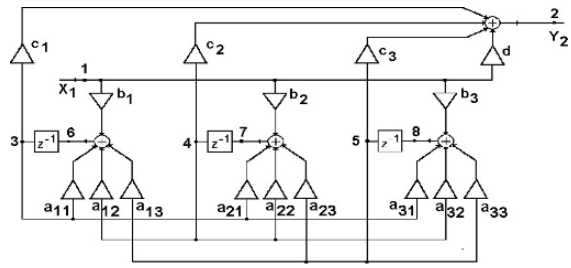


Figure 3: Third order state-space filter.

multipliers if the state-space matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are known (23).

$$\mathbf{D} = 0.25 \qquad \mathbf{C} = \begin{bmatrix} 0.25 & 0.5 \end{bmatrix}$$
$$\mathbf{B} = \begin{bmatrix} 0.75 \\ 0.75 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} -0.5 & -0.5 \\ 0.5 & 0.5 \end{bmatrix} \tag{23}$$

With the assistance of Eq. (5) we obtain the state-space matrix $\mathbf{N_e^{(4)}}$ in the form

$$\mathbf{N}_e^{(4)} = \left[\begin{array}{cccc} 2^{-2} & -1 & 2^{-2} & 2^{-1} \\ \hline (2^{-2}+2^{-1})z^{-1} & 0 & -1+2^{-1}z^{-1} & 2^{-1}z^{-1} \\ (2^{-2}+2^{-1})z^{-1} & 0 & 2^{-1}z^{-1} & -1+2^{-1}z^{-1} \end{array}\right] \tag{24}$$

Provided that we choose elements $n_{ij}^{(5)}$ in the matrix $\mathbf{N^{(5)}}$ in this way $n_{25}^{(5)} = 0$, $n_{35}^{(5)} = z^{-1}$, $n_{45}^{(5)} = 0$ and $n_{55}^{(5)} = -1$ we obtain the Eq. (25) from the Eq. (24). In the new matrix $\mathbf{N^{(5)}}$ the elements in the last row and last column can be chosen. The rest elements of the matrix $\mathbf{N^{(5)}}$ must be calculated by using the Eq. (18).

$$\mathbf{N}^{(5)} = \begin{bmatrix} 2^{-2} & -1 & 2^{-2} & 2^{-1} & 0 \\ 0 & 0 & -1 & 0 & z^{-1} \\ (2^{-2}+2^{-1})z^{-1} & 0 & 2^{-1}z^{-1} & -1+2^{-1}z^{-1} & 0 \\ 2^{-2}+2^{-1} & 0 & -2^{-1} & 2^{-1} & -1 \end{bmatrix} \tag{25}$$

In case that we choose elements $n_{ij}^{(6)}$ in the matrix $\mathbf{N^{(6)}}$ in this manner $n_{26}^{(6)} = 0$, $n_{36}^{(6)} = 0$, $n_{46}^{(6)} = z^{-1}$, $n_{56}^{(6)} = 0$ and $n_{66}^{(6)} = -1$ we obtain the Eq. (26) from the Eq. (25) .

$$\mathbf{N}^{(6)} = \begin{bmatrix} 2^{-2} & -1 & 2^{-2} & 2^{-1} & 0 & 0 \\ 0 & 0 & -1 & 0 & z^{-1} & 0 \\ 0 & 0 & 0 & -1 & 0 & z^{-1} \\ 2^{-2}+2^{-1} & 0 & -2^{-1} & 2^{-1} & -1 & 0 \\ 2^{-2}+2^{-1} & 0 & 2^{-1} & 2^{-1} & 0 & -1 \end{bmatrix} \quad (26)$$

If we choose elements $n_{ij}^{(7)}$ in the matrix $\mathbf{N}^{(7)}$ in this way $n_{27}^{(7)} = 0$, $n_{37}^{(7)} = 0$, $n_{47}^{(7)} = 0$, $n_{57}^{(7)} = z^{-1}$, $n_{67}^{(7)} = 0$ and $n_{77}^{(7)} = -1$ we obtain the Eq. (27) from the Eq. (26) .

$$\mathbf{N}^{(7)} = \begin{bmatrix} 2^{-2} & -1 & 2^{-2} & 2^{-1} & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & z^{-1} & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & z^{-1} & 0 \\ 2^{-2} & 0 & 0 & 0 & -1 & 0 & 2^{-1} \\ 2^{-2}+2^{-1} & 0 & 2^{-1} & 2^{-1} & 0 & -1 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 & -1 \end{bmatrix} \quad (27)$$

Provided that we choose elements $n_{ij}^{(8)}$ in the matrix $\mathbf{N}^{(8)}$ in this manner $n_{28}^{(8)} = 0$, $n_{38}^{(8)} = 0$, $n_{48}^{(8)} = 0$, $n_{58}^{(8)} = 0$, $n_{68}^{(8)} = 2^{-1}$, $n_{78}^{(8)} = 0$ and $n_{88}^{(8)} = -1$ we obtain the Eq. (28) from the Eq. (27). The Eq. (28) is the signal flow-matrix and from this matrix the circuit can be sketched. The structure that correspond to the signal flow matrix $\mathbf{N}^{(8)}$ is presented in Figure 5.

$$\mathbf{N}_s^{(8)} = \begin{bmatrix} 2^{-2} & -1 & 2^{-2} & 2^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & z^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & z^{-1} & 0 & 0 \\ 2^{-2} & 0 & 0 & 0 & -1 & 0 & 2^{-1} & 0 \\ 2^{-2} & 0 & 0 & 0 & 0 & -1 & 0 & 2^{-1} \\ 1 & 0 & -1 & -1 & 0 & 0 & -1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \quad (28)$$

In Figure 4 there is a classical structure of the state-space filter of the second order that has 11 shift operations. In Figure 5 the proposed equivalent state-space structure of the second order is presented with only 7 shift operations. It can be easily verified that both structures have the same impulse response.

In the second example, we shall calculate the elliptic low-pass state space filter of the third order, n=3, with attenuation in the band-pass $a_{max} = 2$ dB, corner frequency $f_1 = 0.6$ and attenuation in the band-stop

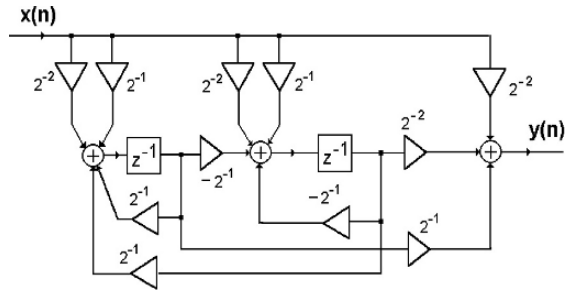


Figure 4: Classical state-space structure of the second order.

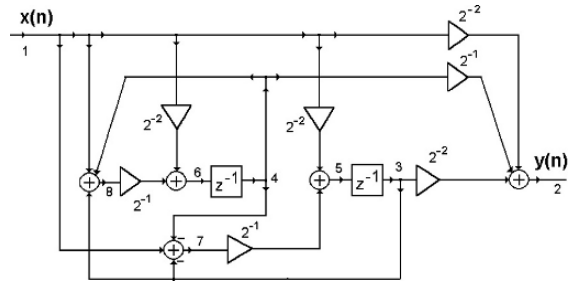


Figure 5: State-space structure without multipliers.

$a_{min} = 15$ dB. By means of MATLAB command [a,b,c,d]= ellip(3,2,15,0.6) we obtain the state-space matrices for the elliptic low-pass filter with the values:

$$\mathbf{C} = \begin{bmatrix} 0.1887 & 0.0360 & 0.0680 \end{bmatrix}$$

$$\mathbf{D} = 0.3673$$

$$\mathbf{B^T} = \begin{bmatrix} 2.1724 & 0.9698 & 1.3201 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 0.1160 & 0.0000 & 0.0000 \\ 0.4982 & -0.3513 & -0.8830 \\ 0.6782 & 0.8830 & -0.2019 \end{bmatrix}$$

With the following equations that realize low-pass state-space filter of the 3rd order we can obtain the impulse response yn(i) in the frequency domain. The equations for calculating yn(i), n6, n7, n8 etc. were derived from Figure 3.

```
a11=0.1160;a12=0.0000;a13=0.0000;
a21=0.4982;a22=-0.3513;a23=-0.8830;
a31=0.6782;a32=0.8830;a33=-0.2019;
b1=2.1724;b2=0.9698;b3=1.3201;
c1=0.1887;c2=0.0360;c3=0.0680;d=0.3673;
n3=0;n4=0;n5=0;
xn=1;
for i=1:1:200
```

```
    yn(i)=xn*d+n3*c1+n4*c2+n5*c3;
    n6=xn*b1+n3*a11+n4*a12+n5*a13;
    n7=xn*b2+n3*a21+n4*a22+n5*a23;
    n8=xn*b3+n3*a31+n4*a32+n5*a33;
    n3=n6;n4=n7;n5=n8;xn=0;
end
 [h,w]=freqz(yn,1,200);
 plot(w,20*log10(abs(h)))
```

In the third example we shall calculate the high-pass elliptic filter with the specification n=3, $a_{max} = 2$ dB, $a_{min} = 15$ dB and corner frequency $f_{-1} = 0.4$. By means of MATLAB commands [a,b,c,d]=ellip(3,2,15,0.4,'high') we obtain the state-space matrices for the elliptic high-pass filter in the form

$$\mathbf{C} = \begin{bmatrix} -0.2597 & -0.0495 & -0.0936 \end{bmatrix}$$

$$\mathbf{D} = 0.3673$$

$$\mathbf{B^T} = \begin{bmatrix} 1.5783 & 0.7046 & 0.9591 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} -0.1160 & 0.0000 & 0.0000 \\ -0.4982 & 0.3513 & 0.8830 \\ -0.6782 & -0.8830 & 0.2019 \end{bmatrix}$$

For the analysis of the high-pass filter we have used the same equation as for the low-pass, but the constants a_{ij}, b_i and c_i in the program must be changed.

In the fourth example we shall realize the elliptic band-pass state-space filter for the lower and upper corner frequencies 0.4 and 0.6 respectively, and $a_{min} = 15$ dB. Using MATLAB commands we get the state matrices **a,b,c** and **d**.

```
 [a,b,c,d]=ellip(3,3,15,[0.4,0.6])
a =
   -0.1374         0         0    0.8626         0         0
    0.2458   -0.1229   -0.2791    0.2458    0.8771   -0.2791
    0.0782    0.2791   -0.0888    0.0782    0.2791    0.9112
   -0.8626         0         0    0.1374         0         0
   -0.2458   -0.8771    0.2791   -0.2458    0.1229    0.2791
   -0.0782   -0.2791   -0.9112   -0.0782   -0.2791    0.0888
b =
    0.7927
    0.2259
    0.0719
   -0.7927
   -0.2259
   -0.0719
c =
    0.1125   -0.0022    0.0420    0.1125   -0.0022    0.0420
d =
    0.1034
```

The general structure for the state-space filter of the arbitrary order can be derived from the structure in Figure 3. To analyze the structure by MATLAB in the Figure 3, the following algorithm must be used. The attenuation of the band-pass state-space filter is presented in Figure 6.

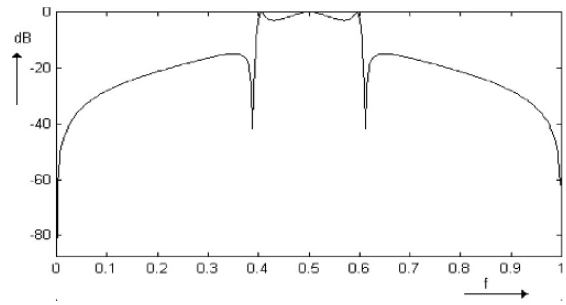


Figure 6: Attenuation of the band-pass state-space Cauer Filter.

```
A11=-0.137;A12=0.000;A13=0.000;A14=0.862;A15=0.000;
A16=0.000;A21=0.245;A22=-0.122;A23=-0.279;A24=0.245;
A25=0.877;A26=-0.279;A31=0.078;A32=0.279;A33=-0.088;
A34=0.078;A35=0.279;A36=0.911;A41=-0.862;A42=0.000;
A43=0.000;A44=0.137;A45=0.000;A46=0.000;A51=-0.245;
A52=-0.877;A53=0.279;A54=-0.245;A55=0.122;A56=0.279;
A61=-0.078;A62=-0.279;A63=-0.911;A64=-0.078;
A65=-0.279;A66=0.088;B1=0.792;B2=0.225;B3=0.071;
B4=-0.792;B5=-0.225;B6=-0.071;C1=0.112;C2=-0.002;
C3=0.042;C4=0.112;C5=-0.002;C6=0.042;D=0.1034;
N3=0;N4=0;N5=0;N6=0;N7=0;N8=0;XN=1;
for i=1:1:500
YN(i)=D*XN+N3*C1+N4*C2+N5*C3+N6*C4+N7*C5+N8*C6;
N9 =B1*XN+N3*A11+N4*A12+N5*A13+N6*A14+N7*A15+N8*A16;
N10=B2*XN+N3*A21+N4*A22+N5*A23+N6*A24+N7*A25+N8*A26;
N11=B3*XN+N3*A31+N4*A32+N5*A33+N6*A34+N7*A35+N8*A36;
N12=B4*XN+N3*A41+N4*A42+N5*A43+N6*A44+N7*A45+N8*A46;
N13=B5*XN+N3*A51+N4*A52+N5*A53+N6*A54+N7*A55+N8*A56;
N14=B6*XN+N3*A61+N4*A62+N5*A63+N6*A64+N7*A65+N8*A66;
N3=N9;N4=N10;N5=N11;N6=N12;N7=N13;N8=N14;XN=0;
end
[h,w]=freqz(YN,1,500);
plot(w,20*log10(abs(h)))
```

4.2 Design of the Filter from the Transfer Function by Matrix method

In this part we shall derive the structure of the digital filter without multipliers that has the transfer function (29)

$$H(z) = \frac{P(z)}{Q(z)} = \frac{0.3437 - 0.2890z^{-1} + 0.4296z^{-2}}{1 + 0.0625z^{-1} - 0.4218z^{-2}} \tag{29}$$

The constants of the transfer function (29) can be decomposed into the following expression

$$0.34375 = 2^{-2} + 2^{-4} + 2^{-5}$$
$$0.2890 = 2^{-2} + 2^{-5} + 2^{-7}$$
$$0.4296 = 2^{-2} + 2^{-3} + 2^{-5} + 2^{-6} + 2^{-7}$$
$$0.0625 = 2^{-4}$$

$$0.4218 = 2^{-2} + 2^{-3} + 2^{-5} + 2^{-6}$$

First of all the matrix $\mathbf{N^{(3)}}$ in the form (30) must be written

$$\mathbf{N^{(3)}} = \begin{bmatrix} 0 & -1 & 1 \\ P(z) & 0 & -Q(z) \end{bmatrix} \tag{30}$$

$\mathbf{N^{(3)}} =$

$$\left[\begin{array}{c|c|c} 0 & -1 & 1 \\ \hline 2^{-2}+2^{-4}+2^{-5} & 0 & -1-z^{-1}2^{-4} \\ -z^{-1}(2^{-2}+2^{-5}+2^{-7}) & & +z^{-2}(2^{-2}+2^{-3} \\ +z^{-2}(2^{-2}+2^{-3}+ & & +2^{-5}+2^{-6}) \\ 2^{-5}+2^{-6}+2^{-7}) & & \end{array}\right] \tag{31}$$

In this case we shall expand transfer matrix $N^{(3)}$ Eq. (31), so the signal U_i must be added. If we choose in the matrix $\mathbf{N^{(4)}}$ the element $n_{24}^{(4)} = 0$ the elements in the first row remain unchanged. In case that $n_{34}^{(4)} = 2^{-5}$ and $n_{41}^{(4)} = 1 - z^{-1} - z^{-2}$ are chosen then element $n_{31}^{(4)}$ in the matrix $\mathbf{N^{(4)}}$ acquire the value $2^{-2} + 2^{-4} - z^{-1}(2^{-2} + 2^{-7}) + z^{-2}(2^{-2} + 2^{-3} + 2^{-6} + 2^{-7})$. Similarly if the element $n_{43}^{(4)} = z^{-2}$ will be chosen then the element $n_{33}^{(4)}$ in the matrix $\mathbf{N^{(4)}}$ acquire the value $-1-z^{-1}2^{-4}+z^{-2}(2^{-2}+2^{-3}+2^{-6})$. The element $n_{44}^{(4)}$ must be equal -1.

$\mathbf{N^{(4)}} =$

$$\left[\begin{array}{c|c|c|c} 0 & -1 & 1 & 0 \\ \hline 2^{-2}+2^{-4}- & 0 & -1-z^{-1}2^{-4}+ & 2^{-5} \\ z^{-1}(2^{-2}+2^{-7}) & & z^{-2}(2^{-2}+2^{-3}+2^{-6}) & \\ +z^{-2}(2^{-2}+2^{-3} & & & \\ +2^{-6}+2^{-7}) & & & \\ \hline 1-z^{-1}+z^{-2} & 0 & z^{-2} & -1 \end{array}\right] \tag{32}$$

In the new matrix $\mathbf{N^{(5)}}$ the elements in the last row and last column can be chosen. Provided that we choose elements $n_{ij}^{(5)}$ in the matrix $\mathbf{N^{(5)}}$ in this way $n_{25}^{(5)} = 0$, $n_{35}^{(5)} = 2^{-2}$, $n_{45}^{(5)} = 0$, $n_{55}^{(5)} = -1$, we obtain $n_{51}^{(5)} = 1 - z^{-1} + z^{-2}$, $n_{52}^{(5)} = 0$, $n_{53}^{(5)} = z^{-2}$ and $n_{54}^{(5)} = 0$ and we get the Eq. (33) from the Eq. (32).

$\mathbf{N^{(5)}} =$

$$\left[\begin{array}{c|c|c|c|c} 0 & -1 & 1 & 0 & 0 \\ \hline 2^{-4}-z^{-1}(2^{-7}) & 0 & -1-z^{-1}2^{-4} & 2^{-5} & 2^{-2} \\ +z^{-2}(2^{-3}+2^{-6} & & +z^{-2}(2^{-3}+2^{-6}) & & \\ +2^{-7}) & & & & \\ \hline 1-z^{-1}+z^{-2} & 0 & z^{-2} & -1 & 0 \\ \hline 1-z^{-1}+z^{-2} & 0 & z^{-2} & 0 & -1 \end{array}\right] \tag{33}$$

In the new matrix $\mathbf{N^{(6)}}$ the elements in the last row and last column can be chosen again. In case that we choose elements $n_{ij}^{(6)}$ in the matrix $\mathbf{N^{(6)}}$ in this manner $n_{26}^{(6)} = 0$, $n_{36}^{(6)} = 2^{-3}$, $n_{46}^{(6)} = 0$, $n_{56}^{(6)} = 0$, $n_{66}^{(6)} = -1$, we get $n_{61}^{(6)} = z^{-2}$, $n_{62}^{(6)} = 0$, $n_{63}^{(6)} = z^{-2}$, $n_{64}^{(6)} = 0$, $n_{65}^{(6)} = 0$ and we obtain the Eq. (34) from the Eq. (33).

$\mathbf{N^{(6)}} =$

$$\left[\begin{array}{c|c|c|c|c|c} 0 & -1 & 1 & 0 & 0 & 0 \\ \hline 2^{-4}-z^{-1}2^{-7} & 0 & -1-z^{-1}2^{-4} & 2^{-5} & 2^{-2} & 2^{-3} \\ +z^{-2}(2^{-6}+2^{-7}) & & +z^{-2}2^{-6} & & & \\ \hline 1-z^{-1}+z^{-2} & 0 & z^{-2} & -1 & 0 & 0 \\ \hline 1-z^{-1}+z^{-2} & 0 & z^{-2} & 0 & -1 & 0 \\ \hline z^{-2} & 0 & z^{-2} & 0 & 0 & -1 \end{array}\right] \tag{34}$$

In case that we choose elements $n_{ij}^{(7)}$ in the matrix $\mathbf{N^{(7)}}$ in this manner $n_{27}^{(7)} = 0$, $n_{37}^{(7)} = 2^{-6}$, $n_{47}^{(7)} = 0$, $n_{57}^{(7)} = 0$, $n_{67}^{(7)} = 0$, $n_{77}^{(7)} = -1$, we get $n_{71}^{(7)} = z^{-2}$, $n_{72}^{(7)} = 0$, $n_{73}^{(7)} = z^{-2}$, $n_{74}^{(7)} = 0$, $n_{75}^{(7)} = 0$, $n_{76}^{(7)} = 0$ and we obtain the Eq. (35) from the Eq. (34).

$\mathbf{N^{(7)}} =$

$$\left[\begin{array}{c|c|c|c|c|c|c} 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ \hline 2^{-4}-z^{-1}2^{-7} & 0 & -1-z^{-1}2^{-4} & 2^{-5} & 2^{-2} & 2^{-3} & 2^{-6} \\ +z^{-2}2^{-7} & & & & & & \\ \hline 1-z^{-1} & 0 & z^{-2} & -1 & 0 & 0 & 0 \\ +z^{-2} & & & & & & \\ \hline 1-z^{-1} & 0 & z^{-2} & 0 & -1 & 0 & 0 \\ +z^{-2} & & & & & & \\ \hline z^{-2} & 0 & z^{-2} & 0 & 0 & -1 & 0 \\ \hline z^{-2} & 0 & z^{-2} & 0 & 0 & 0 & -1 \end{array}\right] \tag{35}$$

Following with this procedure we obtain the matrix $\mathbf{N_s^{(14)}}$. The matrix $\mathbf{N_s^{(14)}}$ is the signal-flow matrix and from this matrix the circuit can be drawn. The structure that correspond to the signal-flow matrix (36) is presented in the Figure 7 with minimum shift operations. In the Figure 8 A) is shown the equivalent filter with 16 shift operations and the Figure 8 B) displays the filter with 5 multipliers. The circuits in the Figure 7 and 8 have the same impulse response.

$\mathbf{N_s^{(14)}} =$

$$\left[\begin{array}{cccccccccccccc} 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2^{-5} & 2^{-2} & 2^{-3} & 2^{-6} & 2^{-4} & 2^{-7} & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & z^{-1} & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & z^{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & z^{-1} \\ z^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & z^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{array}\right] \tag{36}$$

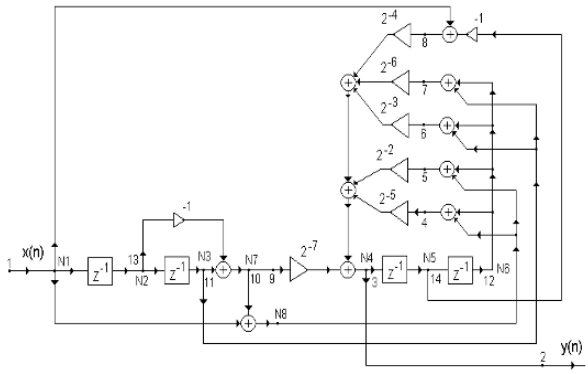


Figure 7: Digital filter with 6 shift operation.

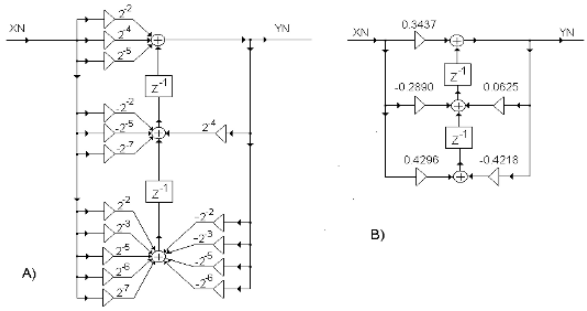


Figure 8: Classical circuit of the digital filter. a) with shift operation, b) with multiplication.

5 CONCLUSION

The method proposed in this paper allows the analysis of digital networks and the construction of new state digital filters. Equivalent filters of differing structures can be found according to various matrix expansion. By this procedure structures can also be obtained without multipliers. This matrix method synthesis of the digital structures seems to be laborious, but in fact it is very simple and the effects are satisfactory when are evaluated using the analysis of the structures. The parts of the MATLAB programs can be used for implantation of the low-pass, high-pass and band-pass state-space filter in digital signal processor DSP.

REFERENCES

C. W. Barnes, *On the Design of Optimal State-Space Realizations of Second-Order Digital Filters.* IEEE Trans. on CAS, Vol. 31, No. 7, pp. 602–608, July 1984.

B.W. Bomar, *New Second-Order State-Space Structures for Realizing Low Roundoff Noise Digital Filters.* IEEE Trans. on ASSP, Vol. 33, No. 1, pp. 106–110, February 1985.

B.W. Bomar, *Computationally Efficient Low Roundoff Noise Second-Order State-Space Structures.* IEEE Trans. on CAS, Vol. 33, No. 1, pp. 35–41, January 1986.

R. Luecker, *Matrixbeschreibung und Analyse zeit discreter Systeme.*, Forschungsbericht Nr. 81. Informationsbibliothek Hannover: RN 2251. 1976.

B. Pšenička and G. Herrera, *Synthesis of Digital Filters by Matrix Method.* Proceedings of the IASTED International Conference on Signal and Image Processing. SIP 97. December 4-6, 1997, New Orleans, Louisiana, USA. pp. 409–412.

B. Pšenička, F. Ugalde and J. Savage, *Design of State Digital Filters*. IEEE Trans. on Signal Processing. Volume 46, Number 9, pp. 2544–2549, September 1998.

B. Pšenička and F. García Ugalde, *Design of State Digital Filters without Multipliers.* ICSPAT 7–19 October 1999, USA.

Z. Smékal and R. Vích R, *Optimized Models of IIR Digital Filters for Fixed Point Digital Signal Processor.* Proceedings of the 6th IEEE International Conference on Electronics Circuits and System ICES 99. September 5-8 1999, Cyprus, pp. 145–148. ISBN 0-7803-5682-9.

Z. Smékal, *Spectral Analysis and Digital Filter Banks.* Proceedings of the International Conference on Research in Telecommunication Technology. (RTT 2001), September 24-26, 2001, Lednice, Czech Republik, pp. 67–70. ISBN 80-214-1938-5.

ANN-BASED MULTIPLE DIMENSION PREDICTOR FOR SHIP ROUTE PREDICTION

Tianhao Tang, Tianzhen Wang and Jinsheng Dou
Institute of Electrical & Control Engineering, Shanghai Maritime University, 1550 Pudong Road, Shanghai, China
thtang@cen.shmtu.edu.cn, wtz0@sina.com

Keywords: Nonlinear time series model, adaptive predictor, artificial neural networks, data mining.

Abstract: This paper presents a new multiple dimension prediction model based on the diagonal recurrent neural networks (PDRNN) with a combined learning algorithm. This method can be used to predict not only values, but also some points in the multi-dimension space. And also its applications in data mining will be discussed in the paper. Some analysis results show the significant improvement to ship route prediction using the PDRNN model in database of geographic information system (GIS).

1 INTRODUCTION

The problem of prediction is denoted to estimate the output of future according to input and output of now and past in some system. Since Kolmogorov presented a linear optimal predictor in 1941, different kinds of trend analysis methods and prediction models have been used for forecasting and control. In this field, the time series prediction model (Box and Jenkins, 1970) and the self-tuning predictor (Wittenmark, 1974) were two kinds of classical prediction methods. The traditional prediction theories based on time series were developed from linear auto recurrent moving average (ARMA) models. And then these theories were extended to nonlinear process. But, if using the traditional methods, it needs to solve the following problems: system modelling, parameter estimating, model modifying and trend forecasting on-line.

In order to solve these problems, some intelligent prediction methods were discussed, in which the artificial neural networks with back propagation algorithm were used more popularly. Prediction based on ANN has made an impact on many disciplines. But there are some difficulties in prediction, particularly in the prediction of multi-variable and non-steady dynamic process.

Recent years, scientists had done much research, and made some progresses in this filed. And we have researched predictive models using neural networks, such as an ANN-based nonlinear time series model for fault detection and prediction in marine system (Tang et al., 1998) and an adaptive predictor based on a recurrent neural network for fault prediction and incipient diagnosis (Tang et al., 2000). Furthermore a direct multi-step adaptive predictor based on a diagonal recurrent neuron network was presented for intelligent system monitoring (Dou and Tang, 2001). These models increased the precision and self-adaptation of prediction in a manner.

However, there existed a problem: former prediction methods could only approach or predict processes with one dimension variable, such as temperature, pressure and flow in an industry process, or stock value in the economic process. In these cases, every parameter must be separately denoted if using a traditional time series model in the dynamic process. But there are some objects with multiple dimension variable or attributes, and must be represented as one predictive model. For example, a ship route has two kinds of attributes: longitude and latitude. A satellite position has three kinds of attributes: longitude, latitude and altitude. So the question is how to predict multi-variable process using one prediction model.

This paper discusses self-adaptation prediction methods based on ANN, and presents a multi-dimension predictive model based on parallel diagonal recurrent neuron network with temporal difference and dynamic BP combined algorithms for time series multi-step forecasting. The paper uses this model in data mining of GIS. Some simulation resolves show the model is able to predict a ship's route according to its position from global position system (GPS).

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 207–215.

2 PRINCIPLE OF ANN-BASED PREDICTOR

The basic issue of a predictor can be described as: if the past output value series $\{x_t\}$ is known, then try to design a predictor to obtain the future output value of forward d-step x_{t+d} under the condition of the minimum predictive errors. If x_{t+d} is expressed as $\hat{x}_{t+d}$, the model of predictor can be described as follows

$$\hat{x}_t(d) = f(X_{t-1}, t, P) \tag{1}$$

Where $X_{t-1}= [x_1, x_2, \cdots, x_{t-1}]^T$ is the past output value vector, i.e. historical data.

$f(\cdot)$ is a certain nonlinear function.
P is the parameter set of the system model.

The predictor is called the minimum covariance optimal predictor, because the covariance of the predictive error is used for criterion function J as follows

$$J = Var(x_{t+d} - \hat{x}_{t+d}) \to \min \tag{2}$$

2.1 RNN-based One-step Predictor

The ANN-based models could be used to construct an adaptive optimal predictor for model identification, parameter correction and value prediction. Assuming that a class of nonlinear processing can be represented by a nonlinear autoregressive moving average (NARMA) model, the NARMA (p, q) model is written as:

$$x_t = g(x_{t-1}, x_{t-2}, \cdots; x_{t-p}, e_{t-1}, \cdots; e_{t-2}, \cdots; e_{t-q}) + e_t \tag{3}$$

Where $g(\cdot)$ is an unknown smooth function, and it is assumed that $E(e_t \mid x_{t-1}, x_{t-2}, \cdots) = 0$ and e_t has a finite variance $Var(e_t) = \sigma^2$. In this case, an approximated condition mean predictor based on the finite past of observations is given by

$$\hat{x}_t = g\left(x_{t-1}, x_{t-2}, \cdots, x_{t-p}, \hat{e}_{t-1}, \hat{e}_{t-2}, \cdots, \hat{e}_{t-q}\right) \tag{4}$$

Where $\hat{e}_j = x_j - \hat{x}_j \qquad j = t-1, t-2, \cdots, t-q$

For NARMA (p, q) modelling and predicting, a recurrent neural network (RNN) was presented (Connor et al., 1994). The RNN topology is shown in Figure 1. This neural network can be used to approximate the NARMA (p, q) model. The output of the basic RNN-based predictor is

$$\hat{x}_t = \mathbf{W_o f(S)} \tag{5}$$

$$\mathbf{s} = \mathbf{W_{ih}x} + \mathbf{W_{eh}e} - \boldsymbol{\theta} \tag{6}$$

Where $\mathbf{f}(\cdot)$ is a Sigmoid function vector or other finite continuous monotonically increasing function vectors;

$\mathbf{S}$ is a state vector of the hidden layer;
$\mathbf{W_{ih}}$ is the weight matrix between the input layer and the hidden layer;
$\mathbf{W_{eh}}$ is the weight matrix from feedback units to hidden units;

$\mathbf{x} = [x_{t-1}, x_{t-2}, \cdots, x_{t-p}]^T$ is the input vector;

$\mathbf{e} = [e_{t-1}, e_{t-2}, \cdots, e_{t-q}]^T$ is the error vector;

$\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_H]^T$ is a threshold vector;

$\mathbf{W_o}$ is the weight matrix between the hidden layer and the output layer.

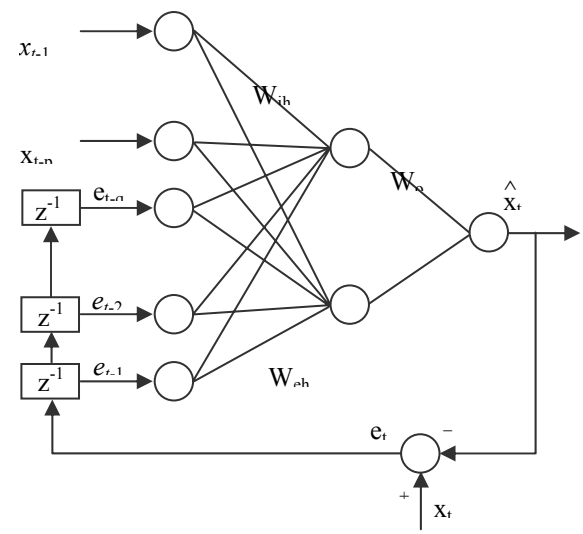


Figure 1: Recurrent network for NARMA model.

The parameters of $\mathbf{W_o}$, $\mathbf{W_{ih}}$ and $\mathbf{W_{eh}}$ are estimated by a dynamic BP learning algorithm (Williams and Peng, 1990). That is by learning of RNN to minimize the following error function:

$$E = \frac{1}{2}\sum_{t=p+1}^{N}(x_t - \hat{x}_t)^2 \tag{7}$$

But, just one step prediction will make by the basic predictor. So some improvement RNN-based predictors had been discussed (Tang, 2000).

2.2 RNN-based Multi-step Predictor

In order to implement the multi-step prediction, the NARMA model should be extended to

$$\hat{x}_t(d) = g\left(\hat{X}_t(d-i), \hat{e}_{t+d-j}\right) \tag{8}$$

Where

$$\hat{e}_{t+d-j} = x_{t+d-j} - \hat{x}_t(d-j), \quad j = 1, 2, \cdots, q$$

$$\hat{X}_t(d-i) = \left[\hat{x}_t(d-1), \hat{x}_t(d-2), \cdots, \hat{x}_t, x_{t-1}, \cdots, x_{t-p}\right]^{\mathrm{T}}$$

The output of the multi-step RNN model is

$$\hat{\mathbf{x}}_\mathbf{t} = \mathbf{W_o f(S)} \tag{9}$$

Where $\mathbf{S} = \mathbf{W_{ih}x} + \mathbf{W_{eh}e} - \boldsymbol{\theta}$

And the error function is

$$E = \frac{1}{2}\sum_{t=p+1}^{\mathrm{N}} (\mathbf{x_t} - \hat{\mathbf{x}}_\mathbf{t})^{\mathbf{T}} (\mathbf{x_t} - \hat{\mathbf{x}}_\mathbf{t}) \tag{10}$$

Using the dynamic BP (DBP) learning algorithm, and assuming that the dimensions of the input, error, hidden and output matrices could be represented as i, e, h, and o, the iteration formulae of the weight values of the RNN prediction model can be obtained

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \eta \frac{\partial E}{\partial \mathbf{W}} \tag{11}$$

To the weight values of the output layer, there is

$$\frac{\partial E}{\partial \mathbf{W_o}} = -\sum_{t=p+1}^{\mathrm{N}} \mathbf{I}^{(\mathbf{h}\times\mathbf{o})} \mathbf{f(s)e_t} \tag{12}$$

Where $\mathbf{e_t} = \mathbf{x_t} - \hat{\mathbf{x}}_\mathbf{t}$ (13)

And **I** is an identity matrix.

To the weight values between the hidden layer and input layer, there is

$$\frac{\partial E}{\partial \mathbf{W_{ih}}} = -\sum_{t=p+1}^{\mathrm{N}} (\mathbf{I}^{(\mathbf{i}\times\mathbf{h})} \cdot (\mathbf{I}^{(\mathbf{h})} \otimes \mathbf{x}) + \frac{\partial \mathbf{e}}{\partial \mathbf{W_{ih}}} \mathbf{W}_{\mathbf{eh}}^{\mathbf{T}}) \cdot \frac{\partial \mathbf{f(s)}}{\partial \mathbf{s}} \cdot \mathbf{W}_{\mathbf{o}}^{\mathbf{T}} \mathbf{e_t} \tag{14}$$

Where ⊗ is Kronecker product.

$$\frac{\partial \mathbf{e}}{\partial \mathbf{W_{ih}}} = \left[\frac{\partial e_{t-1}}{\partial \mathbf{W_{ih}}}, \frac{\partial e_{t-2}}{\partial \mathbf{W_{ih}}}, \cdots, \frac{\partial e_{t-q}}{\partial \mathbf{W_{ih}}}\right] \tag{15}$$

To the weight values of the hidden layer and feedback units, there is

$$\frac{\partial E}{\partial \mathbf{W_{eh}}} = -\sum_{t=p+1}^{\mathrm{N}} \left(\mathbf{I}^{\mathbf{e}\times\mathbf{h}} (\mathbf{I}^{\mathbf{h}} \otimes \mathbf{e}) + \frac{\partial \mathbf{e}}{\partial \mathbf{W_{eh}}} \mathbf{W}_{\mathbf{eh}}^{T}\right) \frac{\partial \mathbf{f(s)}}{\partial \mathbf{s}} \mathbf{W}_0^T \mathbf{e}_t \tag{16}$$

Where

$$\frac{\partial \mathbf{e}}{\partial \mathbf{W_{eh}}} = \left[\frac{\partial e_{t-1}}{\partial \mathbf{W_{eh}}}, \frac{\partial e_{t-2}}{\partial \mathbf{W_{eh}}}, \cdots, \frac{\partial e_{t-q}}{\partial \mathbf{W_{eh}}}\right] \tag{17}$$

3 DRNN PREDICTIVE MODELS

In order to obtain the optimal predictive value of the future output of the analyzed system based on its historical data, a stochastic dynamic model of the analyzed system should be set up, which can modify the model parameter adaptively. A diagonal recurrent neural network (DRNN) was used to represent the dynamic process based on NARMA model (Dou and Tang, 2001).

Figure 2 shows the structure of the DRNN-based predictive model. The neural network model with two inputs and several outputs includes three layers. In order to realize direct multi-steps prediction, the output layer composes of d linear neural units. And the middle layer (i.e. hidden layer) makes up N_h nonlinear dynamic neurons whose map function is the sigmoid function, and each of the hidden unit includes a self-feedback with one step delay (recursion layer). The input layer includes two linear neurons, and one of them accepts x_{t-1} as input signal, another accepts $\hat{x}_{t-1}$, which is one-step delay of the output x_t. This network can be regarded as a parsimonious version of the Elman-type network. It has a diagonal structure, that is, there is no interaction between different dynamic neurons.

The transfer function of this network is described as follows. Suppose w_{jk} (j =1, 2, ⋯, N_h; k =1, 2, ⋯, d) are the connection weights of the output layer, a_i (i=1, 2,⋯, N_h) are the connection weights of the

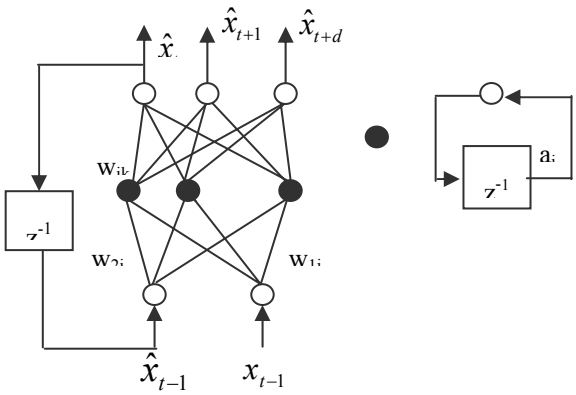


Figure 2: DRNN-based predictive mode.

recursion layer, w_{1j} (j =1, 2, ⋯, N_h) are the connection weights of the input x_{t-1} to each of the hidden units, w_{2j} (j = 1, 2, ⋯, N_h) are the connection weights of the input $\hat{x}_{t-1}$ to each of the hidden units, the output of the neural unit of the output layer is expressed as $\hat{x}_{t+k}$, (k =1, 2,⋯d), the output of an neurons of hidden layer is expressed as $O_j(t)$, (j =1, 2,⋯, N_h), the input of the neural unit of the hidden layer is expressed as $S_j(t)$ (j = 1, 2, ⋯, N_h), then the map relations of the DRNN are shown as fellows

$$\hat{x}_t(k) = \sum_{j=1}^{N_h} [w_{jk} O_j(t) - \theta_k] \tag{18}$$

Where $$O_j(t) = f(S_j(t) - \theta_j) \tag{19}$$

$$S_j(t) = a_i O_i(t-1) + w_{1j} x_{t-1} + w_{2j} \hat{x}_{t-1} \tag{20}$$

$f(\cdot)$ is Sigmoid function, θ_k is the threshold of the neural unit of output layer, θ_j is the threshold of the neuron of hidden layer. The initial conditions of this model are $O_j(0) = 0$ and $S_j(0) = 0$. So the transfer function of the neural network is a nonlinear continuous function, and the output of the neural network $\hat{x}_t(k)$ is an appropriate nonlinear function of all input signals (x_1, x_2, ⋯, x_{t-1}).

It is known that the supervised learning algorithm based on the error between the actual output and the anticipant output is not suitable for the direct multi-step predictive model. And a neural network with the fixed structure and parameters is difficult or even impossible to express the inherent dynamic performance of the uncertain nonlinear systems. For this reason the temporal difference (TD) learning algorithm and the dynamic back propagation (DBP) algorithm are synthesized for the network training. The combined learning algorithm will adaptively modify the parameters of the predictive model according to the errors between the predictive value and the actual detective value.

Suppose P^t_k is the output of the jth output neuron at t time, i.e. $\hat{x}_t(k)$, P^{t+1}_{k-1} is the output of the (k-1)th output neuron at t+1 time, i.e. It is obvious that P^t_k and P^{t+1}_{k-1} are the predictive output value of the analyzed system at the same time. And they should be equal if the prediction is accurate. So the P^{t+1}_{k-1} can be used as the expectant output of the P^t_k . This is the basis of TD learning algorithm. The training error of one learning sample of the DRNN is expressed as

$$E = \frac{1}{2} \sum_{k=0}^{d} (x_{t+k} - \hat{x}_t(k))^2 \tag{21}$$

According to TD learning algorithm, it can be expressed as

$$E = \frac{1}{2} \sum_{k=0}^{d} (P_k^{t+1} - \hat{x}_t(k))^2 \tag{22}$$

Here P_0^{t+1} is defined as the expectant output of x_t at t time.

4 PDRNN BASED MULTIPLE DIMENSION PREDICTOR

One dimension predictive model can predict an object with one variable. A multiple dimension prediction model can predict an object with more than one kind of attribute at the same time. In the multi-dimension prediction model, there are different relations in different attributes and the relations can be changed by a dynamic process. For this reason, ANN-based adaptive predictors must be introduced to modify the parameters of a predictive model on-line.

4.1 The Framework of PDRNN Model

In order to solve the predictive problem of objects with multi-attributes, this paper presents a new multi-dimensional predictive model based on the diagonal recurrent neural networks (PDRNN) with a parallel combined learning algorithm. Figure 3 shows the framework of PDRNN model. There are four layers in this model, the first layer is the

network input layer; the second layer is the network input assignment layer; the third layer is the network hidden layer, in which every hidden unit includes a self-feedback with one step delayed; the forth layer is the network output layer, (the network output layer connects with the network input layer through one-step feedback). There are *n* dimension variables to input paralleled in the network input layer.

4.2 The Mathematical Description of the PDRNN Model

As shown in Figure 3, there are *p* input units in the network input layer, every input unit has *n* dimension variables, X_t can be obtained by $\hat{X}_{t+1}$ with one step delayed, so each attribute variable has *p*-1 input values in this network every time in fact. The network input assignment layer assigns *n* values of each variable to *n* sub-input layers paralleled, as shown in equation (23). The hidden layer has *n* sub-hidden layers paralleled, every sub-hidden layer has N_h^l (the number of sub hidden layer, l =1, 2, …, n)nonlinearity units with S functions or T functfunctions, the value of N_h^l can be changed, and every sub-hidden layer has self-feedback with one step delayed. In this network, *n* paralleled sub-networks consisted of the sub-input layers and sub-hidden layers, all the parallel sub-networks respectively train different attributes at the same time. This network neglected the relations in the different attributes and attribute values. There are *d* linear units in the network output layer, which can do *d*-step prediction at most. The mathematical model could be described as follows. The network input layer is

$$[X_t, X_{t-1}, \cdots, X_{t-p}]$$

Where $$X_{t-i} = \left[X_{t-i}^1, X_{t-i}^2, \cdots, X_{t-i}^n\right] \quad (23)$$

i = 1, 2, …, p

The l th parallel sub-network's sub input layers:

$$\left[X_t^l, X_{t-1}^l, \cdots, X_{t-p}^l\right] \quad (24)$$

The l th parallel sub-network's sub hidden layers:

$$O_j^l(t) = f(S_j^l(t) - \theta_j^l) \quad (25)$$

In equation (25), at every t time, O_j^l is the output of the j th hidden unit in the l th parallel sub-network, is Sigmoid function or Tangent function. And there is

$$S_j^l(t) = a_j^l O_j^l(t-1) + \sum_{i=0}^{p} w_{ij}^l X_{t-i}^l \quad (26)$$

Where $S_j^l(t)$ is the sum of all the inputs in the jth hidden layer of the l th parallel sub-network;

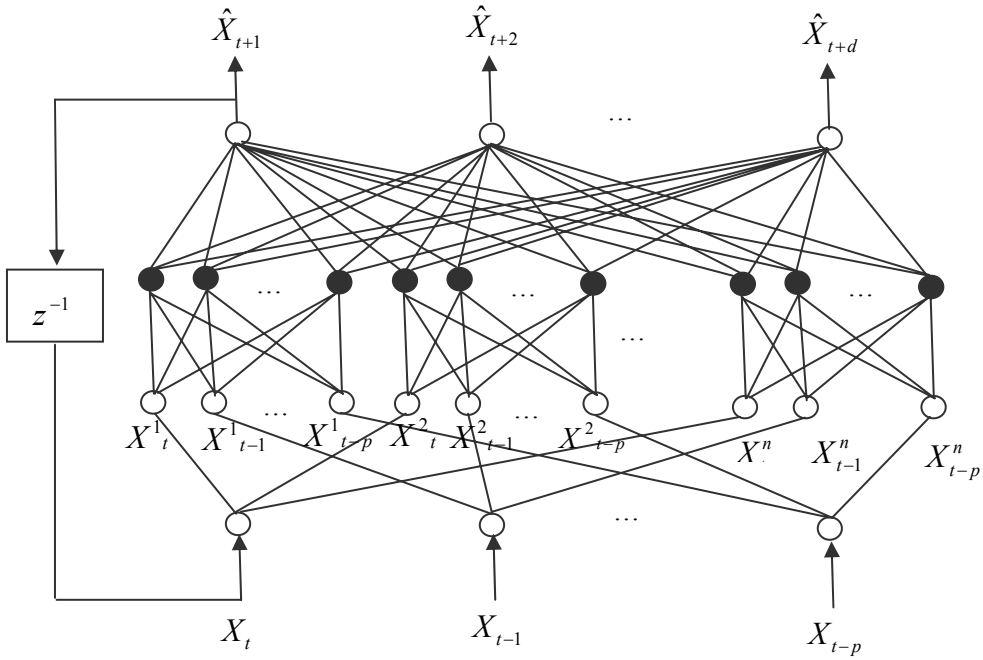


Figure 3: Multi-dimension predictive model based on PDRNN.

w_{ij}^{l} is the normalized (relative) fulfilment weights between the sub-input layer and sub-hidden layer of the *l* th parallel sub-network;

a_{j}^{l} is the self-recurrent layer's relative weights of the *j*th hidden unit in the *l*th parallel sub-network.

If $\hat{X}_{t+k}$ is defined as the output of the *k* th output unit and includes all the attribute values at this time, the output variables of the network model could be written by equation (27) as below

$$\hat{X}_{t+k} = \left[\hat{X}_{t+k}^{1} \quad \hat{X}_{t+k}^{2} \quad \cdots \quad \hat{X}_{t+k}^{n} \right] \tag{27}$$

Where
$$\hat{X}_{t+k}^{l} = \sum_{j=1}^{N_h^l} w_{jk}^{l} O_j^{l}(t) \tag{28}$$

is the *k*th output layer's output of the *l* th attribute at *t* time, in which *k* = 1, 2, ⋯, *d*, and *l* = 1, 2, ⋯, *n*. In the initialization, the threshold value of all the nerve units were neglected at every *t* time, and

$$O_j^{l}(0) = 0$$

5 THE LEARNING ALGORITHM

This paper combines the TD and DBP method to train PDRNN model. If $\hat{X}_{t+k}^{l}$ as the *k* th output of the *l* th attribute at *t* time, is the predictive value of the *l* th attribute at *t*+*k* time, X_{t+k}^{l} is real value of the *l* th attribute at *t*+*k* time in the future.

In the ideal condition, $\hat{X}_{t+k}^{l}$ is equal to X_{t+k}^{l}, but usually there are some errors in practice. It could be represented by following error function.

$$e^{l} = \frac{1}{2} \sum_{k=0}^{d} \left(X_{t+k}^{l} - \hat{X}_{t+k}^{l} \right)^2 \tag{29}$$

Equation (29) is the training error of the *l* th attribute at *t*+*k* time. Here use a DBP method to correct relative weighs. The learning algorithm is as follows

$$\frac{\partial e^{l}}{\partial w_{jk}^{l}} = \frac{\partial e^{l}}{\partial \hat{X}_{t+k}^{l}} \frac{\partial \hat{X}_{t+k}^{l}}{\partial w_{jk}^{l}} = -\left(X_{t+k}^{l} - \hat{X}_{t+k}^{l} \right) O_j^{l}(t) \tag{30}$$

Where w_{jk}^{l} are the normalized weights between the sub hidden layer and sub output layer of the *l* th paralleled sub-network. The formula corrected of as follows

$$\begin{aligned} w_{jk}^{l}(t+1) &= w_{jk}^{l}(t) - \xi^{l} \frac{\partial e^{l}}{\partial w_{jk}^{l}} \\ &= w_{jk}^{l}(t) + \xi^{l} (X_{t+k}^{l} - \hat{X}_{t+k}^{l}) O_j^{l}(t) \end{aligned} \tag{31}$$

Where ξ^{l} is learning parameter of the *l*th attribute's output layer, w_{jk}^{l} is the normalized (relative) fulfilment weights between the sub input layer and sub hidden layer of the *l* th paralleled sub-network.

$$\begin{aligned} \frac{\partial e^{l}}{\partial w_{ij}^{i}} &= \frac{\partial e^{l}}{\partial \hat{X}_{t+i}^{l}} \frac{\partial \hat{X}_{t+i}^{l}}{\partial O_j^{l}(t)} \frac{\partial O_j^{l}(t)}{\partial w_{ij}^{l}} \\ &= -\sum_{k=0}^{d} \left(X_{t+k}^{l} - \hat{X}_{t+k}^{l} \right) w_{jk}^{l} \frac{\partial O_j^{l}(t)}{\partial w_{ij}^{l}} \end{aligned} \tag{32}$$

The formula is corrected for $w^{l}{}_{ij}$ as follows

$$w_{ij}^{l}(t+1) = w_{ij}^{l}(t) - \eta^{l} \frac{\partial e^{l}}{\partial w_{ij}^{l}} \tag{33}$$

Where η^{l} is learning parameter of the *l* th attribute's input layer.

For the self-recurrent layer's relative weights, the formula is corrected for a_{j}^{l} as follows

$$\begin{aligned} \frac{\partial e^{l}}{\partial a_{j}^{l}} &= \frac{\partial e^{l}}{\partial \hat{X}_{t+k}^{l}} \frac{\partial \hat{X}_{t+k}^{l}}{\partial O_j^{l}(t)} \frac{\partial O_j^{l}(t)}{\partial a_{j}^{l}} \\ &= -\sum_{k=0}^{d} (X_{t+k}^{l} - \hat{X}_{t+k}^{l}) w_{jk}^{l} P_j^{l}(t) \end{aligned} \tag{34}$$

Where $P_j^{l} = \dfrac{\partial e^{l}}{\partial a_{j}^{l}}$, and $P_j^{l}(0) = 0$

So the update formula can be obtained as follows

$$a_{j}^{l}(t+1) = a_{j}^{l}(t) - \mu^{l} \frac{\partial e^{l}}{\partial a_{j}^{l}} \tag{35}$$

Where μ^{l} is the learning parameter of the *l* th attribute's self-recurrent layer.

The learning procedure is firstly to adjust the network framework, make sure of the number of input layers, hidden layers and output layers, and the number of maximal learning steps. The above parameters are significant for predictive accuracy. Second initialize the network (for new data, the

initialization is random), then ANN begins learning. Equation (29) serves as a standard to judge the predictive value of each attribute. In order to avoid the learning of ANN falling into a dead area, the learning step could be adjusted according to different attribute values. Here a "$\overline{e}$" function was presented, equation (36) serves as a standard to judge the predictive values of all the attributes.

$$\overline{e} = \frac{1}{n-k}\sum_{t=k}^{n} E_t \tag{36}$$

And
$$E_t = \sqrt{\sum_{l=1}^{m} (e_t^l)^2} \tag{37}$$

Where e^l is the error between the real value and the predictive value of the l th attribute at t time, can be changed according to different attributes. $\overline{e}$ is the average error, used to judge the convergence of the multi-dimension predictive model based on PDRNN. Because the initialization is random, the first couple of predictive values may be not very good, $\overline{e}$ is computed from the kth prediction value. The training will be stopped when $\overline{e}$ is up to standard, or reset up the network framework until $\overline{e}$ confirms to requirements.

6 SIMULATIONS AND APPLICATION

The prediction based on PDRNN extends ANN-based time series prediction model from a single attribute to a multi-attribute. Figure 4 is a three-step predictive value contrasting with real values of straight line, real lines are real values, "*" indicates the predictive values. The points of straight line have two kinds of attributes, every parallel sub-network has ten sub input layers and fifteen sub hidden layers. T function is used in the hidden units, the maximum of e is $2.265e^{-5}$, $\overline{e}$ is $8.36e^{-8}$.

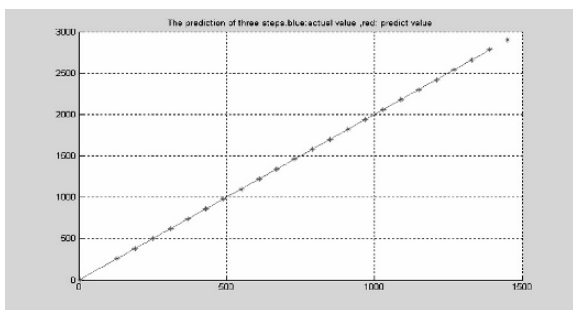


Figure 4: Predictive value contrasting.

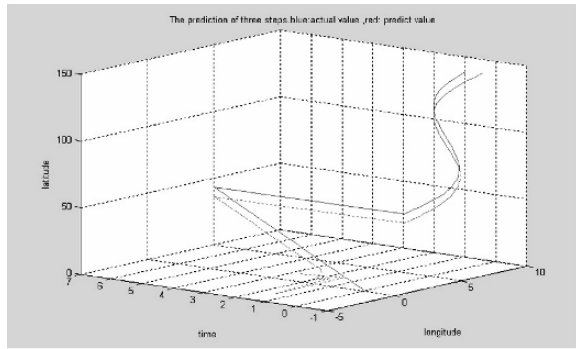


Figure 5: Prediction of nonlinearity.

Figure 5 is a three-step predictive value of nonlinearity as follows

$$y = \begin{cases} 2\times x, & 0 \le x < 2\pi \\ Sin(x), & 2\pi \le x < 4\pi \end{cases}$$

Figure 6 is a three-step predictive value of 3D curve contrasting with real value, T function is used in the hidden units, every parallel sub-network only has ten sub input layers and fifteen sub hidden layers. The learning step of 3D curve is more than 2D curve. The maximum predictive error of 3D curve is 0.0011, $\overline{e}$ is $8.265e^{-4}$.

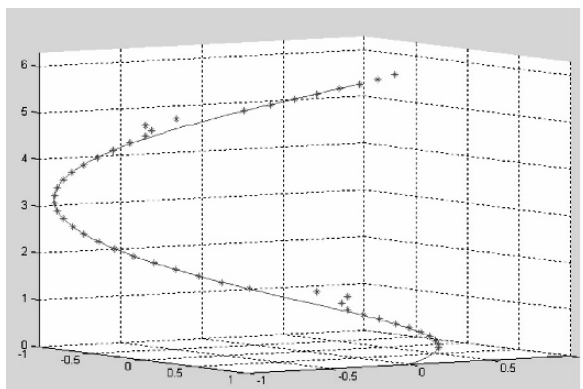
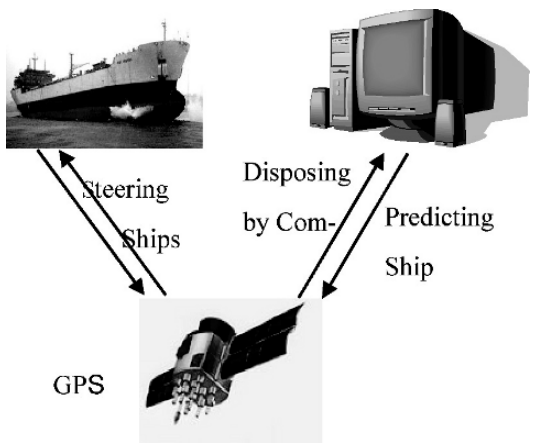


Figure 6: Prediction of 3D curve.

An application process for GIS in Marine Engineering with the predictor based on PDRNN is shown in figure 7.

In the system, data recorded abundant ship positions from GPS, and established a database through ACCESS. After data pre-processing, the ship route could be selected and drawn in an



electronic chart. Using the PDRNN predictor the tracking of the ship route could be forecasted.

The process of ship route prediction by means of a PDRNN model is as follows: first, get the distribution data of the ship's position from GPS (Wang et al., 2003), then select sample points. In this paper, the each sample point was selected every 2.5 hours. Figure 8 is a selected ship route. Finally, the ship route was predicted by means of a PDRNN model.

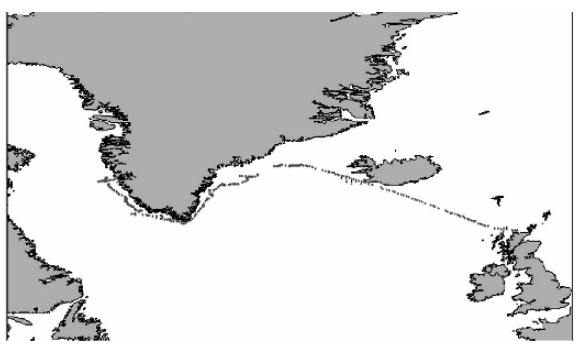


Figure 8: The ship route in GIS.

Figure 9 is another prediction results of a ship route. This chart shows that this ship route is a variant random process, but the predictive algorithm based on a PDRNN model can follow this process, and do three-step prediction. The predictive maximum error of the ship route is $1.065e^{-14}$, e is $3.326e^{-15}$. Thus the error of prediction is small.

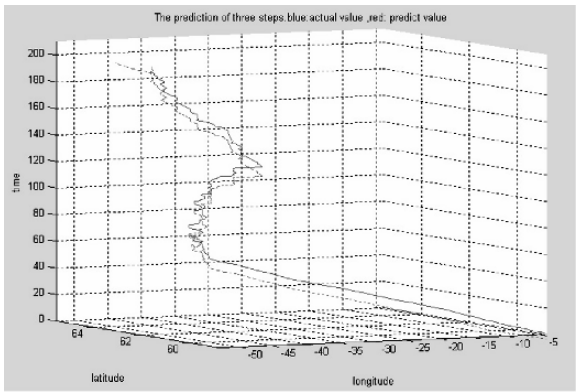


Figure 9: The prediction of ship route.

7 CONCLUSIONS

As mentioned above, the NARMA models based on the recurrent neural networks with a TD-DBP algorithm is suited for trend prediction. This paper presents a multi-dimension predictive model based on PDRNN. This predictor has the advantages as follows:

(1) A multi-dimension prediction model based on PDRNN has been proposed;
(2) The new predictor has the advantages in simple architecture with parallel subnetworks and high adaptation through on-line learning;
(3) The simulations have shown the predictor has accurate nonlinear and stochastic prediction ability.
(4) The "$\overline{e}$" function can judge if the whole network structure confirms to requirements, and has presented an "input plus" method, which can reduce the training time.

The application in ship routing prediction shows the new predictive model is better to predict a multiple dimension dynamic process.

REFERENCES

Box, G. E. P. and Jenkins, G. M., 1970. *Time series analysis of forecasting and control*. Holden-day, San Francisco.

Connor, J. T., Martin R. D. and Atlas L. E., 1994. Recurrent neural networks and robust time series prediction. In *IEEE Trans. on neural networks*, No.5, pp. 240–254.

Dou, J. and Tang, T., 2001. A DRNN-based direct multi-step adaptive predictor for intelligent systems. In *Proceedings of the IASTED International Conference on Modelling, Identification, and Control*, Vol. 2, pp. 833–838, Innsbruck, Austria.

Goodchild, M. F., 1992. Geographic data modeling. In *Computers and Geosciences*, Vol. 18, No. 4, pp. 401–408.

Tang, T. *et al*., 1998. ANN-based nonlinear time series models in fault detection and prediction. In *Preprint of IFAC Conference on CAMS'98*, pp. 335–340, Fukuoka, Japan.

Tang, T. *et al*., 2000. A RNN-based adaptive predictor for fault prediction and incipient diagnosis. In *UKACC Control 2000*, *Proceedings of the 2000 UKACC International Conference on Control*. Cambridge, UK.

Wang, T., Hao, R. and Tang, T., 2003. A data mining method for GIS in marine engineering. In *Navigation of China*. No. 3, pp. 1–4.

Williams, R. J. and Peng, J., 1990. An efficient gradient-based algorithm for on-line training of recurrent neural networks. In *Neural Computation*, No. 4, pp. 490–501.

Wittenmark, B. A., 1974. A self-tuning predictor. *IEEE Trans. on Automatic Control*, No. 6, pp. 848–851.

A PARAMETERIZED POLYHEDRA APPROACH FOR THE EXPLICIT ROBUST MODEL PREDICTIVE CONTROL

Sorin Olaru
Supelec
3 rue Joliot Curie, Gif-sur-Yvette, France
sorin.olaru@supelec.fr

Didier Dumur
Supelec
3 rue Joliot Curie, Gif-sur-Yvette, France
didier.dumur@supelec.fr

Keywords: Predictive control, constraints, parameterized polyhedra, multiparametric optimization.

Abstract: The paper considers the discrete-time linear time-invariant systems affected by input disturbances. The goal is to construct the robust model predictive control (RMPC) law taking into account the constraints existence from the design stage. The explicit formulation of the controller is found by exploiting the fact that the optimum of a min-max multi-parametric program is placed on the parameterized vertices of a parameterized polyhedron. As these vertices have specific validity domains, the control law has the form of a piecewise linear function of the current state. Its evaluation replaces the time-consuming on-line optimization problems.

1 INTRODUCTION

Model Predictive Control (MPC) enjoys a remarkable reputation among the control design techniques for process industries. In the beginnings, practitioners used MPC in the unconstrained closed forms due to its simplicity and versatility and dealt with the constraints violation a posteriori. In the '90s, theoreticians underlined the fact that constraints could be included at the design stage with excellent results towards the feasibility, stability or robustness. The inconvenience of these schemes, which represented also an impasse in applying the constrained predictive control to high sampling rate systems, was the relative high complexity of the optimization problem to be solved at each sampling period. Lately, the constrained MPC paradigm was reformulated in terms of LMI ((Kothare et al., 1996) and later related literature) with great expectations towards the reduction of computational time. Even if the optimization problems earned interesting structural properties, the class of system to be controlled was still limited due to the fact that the computational effort per sampling time is important.

An improvement from the on-line computational point of view can be achieved if the explicit solution of the MPC optimization problem is constructed. In this way, at each sampling time, a (control) function has to be evaluated and the use of iterative optimization routines is avoided. In the case of linear systems with linear/quadratic criteria the explicit solution can be stated as a piecewise affine function of the system state.

Regarding the effective construction of explicit solutions, it is known that the MPC strategy is based on a multi-parametric optimization problem due to the fact that both the global optimum and the set of constraints are parameter dependent. In the nominal case corresponding with a quadratic optimization problem and linear constraints, the explicit solution was investigated with success using an algebraic approach in (Bemporad et al., 2002b), using geometrical arguments in (Seron et al., 2002), (Olaru and Dumur, 2004) and lately based on dynamic programming in (Goodwin et al., 2004). These alternative approaches with different maturity degrees converged towards similar formulations and thus represent valuable control design techniques.

In the case of robust MPC, the explicit solution is somehow more difficult to achieve as the optimization problem remains a multi-parametric one but is based on a min-max cost function. It was successfully tackled in studies like (Bemporad et al., 2001) using methods based on the exploration of the parameters space and the KKT optimality conditions but the alternative fully geometrical methods (using the double description of the feasible domain) do not present similar solutions so far. The current work is trying to compensate this setback through the construction of the complete explicit solution for the robust MPC

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 217–226

problems using the concept of parameterized polyhedra (Loechner and Wilde, 1997) and their correspondent parameterized vertices. It is shown that the optimal solution is founded on a combination of such parameterized vertices and entire family of solution can be idetified.

In the following, Section 2 formulates the robust MPC problem and the related optimization. Section 3 details the optimization problem, Section 4 presents the robust MPC explicit solution, while in Section 5 the procedure is applied to a particular example.

2 ROBUST MPC FORMULATION

Model-based Predictive Control strategy implies the minimization of a cost function based on the predicted plant evolution. This strategy is also called the receding horizon principle and differs from one algorithm to another by the plant model chosen or by the cost function considered.

2.1 The Model

Consider a discrete-time linear time-invariant system affected by an input disturbance:

$$x_{t+1} = Ax_t + Bu_t + Ev_t \tag{1}$$

and subject to a set of linear constraints:

$$Cx_t + Du_t \leq d \tag{2}$$

The vectors $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ represent the states and inputs while $v_t \in \mathbb{R}^p$ is the unknown vector of disturbances lying inside a polytope containing the origin defined by a set of linear constraints:

$$V = \{v \,|Mv \leqslant l; l \geqslant 0\} \tag{3}$$

In the following, the pair (A, B) is supposed to be stabilizable and it is assumed that the full measurement of the current state is available at each time t.

2.2 The Optimal Control Problem

MPC is an optimization based technique. In opposition to the nominal case where quadratic cost functions are used (Maciejowski, 2002), (Rossiter, 2003), in the case of models affected by disturbance, a min-max optimization is preferred, resulting a RMPC formulation:

$$\min_{u_t,\ldots,u_{t+N_u-1}} \left\{ \max_{v_t,\ldots,v_{t+N-1}} \left\{ S_{P_\lambda}(x_{t+N|t}) + \right.\right. \\ \left.\left. + \sum_{k=1}^{N-1} \left\|Qx_{t+k|t}\right\|_\infty + \sum_{k=0}^{N_u-1} \left\|Ru_{t+k}\right\|_\infty \right\} \right\} \tag{4}$$

$$\begin{aligned} \text{s.t.:} \quad & Cx_{t+k|t} + Du_{t+k} \leqslant d, k = 1, \ldots, N \\ & Mv_{t+k} \leqslant l, k = 0, \ldots N-1 \\ & x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k} + Ev_{t+k}, \\ & k \geqslant 0, x_{t+N|t} \in P_\lambda \end{aligned} \tag{5}$$

with Q, R weighting matrices, $\|*\|_\infty \triangleq \max_{i=1,\ldots,r}\left(*^i\right)$, where $*^i$ is the i-th element of the vector $* \in \mathbb{R}^r$. The state predictions $x_{t+k|t}$ are obtained based on the current state vector x_t and by applying the input sequence $u_t, \ldots, u_{t+Nu-1}$, to model (1) over a control horizon. Note that, in the general case, the control (N_u) and the prediction (N) horizons might be different if the control vector has a fix formulation for $N_u \leqslant k \leqslant N$. Conversely, the disturbance sequence $v_t, \ldots, v_{t+N-1}$ affects the prediction over the whole prediction horizon.

The stability of the MPC scheme depends on the chosen horizons and on the terminal cost. In order to guarantee the stability, an infinite prediction horizon should be used. Such a choice transforms (4)-(5) in an intractable problem. The solution is then to choose a finite prediction horizon and to consider that after this point the system trajectory is brought inside a positively invariant set, P_λ, that can be computed off-line (Kerrigan, 2000). To this terminal region a function $S_{P_\lambda}(x)$ can be associated, appearing in (4) as a terminal cost penalizing the evolution from N to ∞.

Applying a *receding horizon strategy* the optimization (4)-(5) is solved at each sampling time t using the measured state vector x_t (playing the role of parameter for the optimization). If $\mathbf{k}^*_{\mathbf{u}}(x_t) = \left\{u^*_t, \ldots, u^*_{t+N_u-1}\right\}$ is the solution to (4)-(5), the input applied to the system (1) is the first value of this sequence $\mathbf{k}^*_{\mathbf{u}}(x_t)$ such that $u_t = u^*_t$, the other values are abandoned and the procedure is restarted.

2.3 Open Loop vs. Closed Loop Prediction

A special concern must be given to the choice of the control horizon. Indeed, this parameter is sensitive as it reflects the number of degrees of freedom available to ensure the constraints fulfillment for all possible combinations of disturbances. On the other hand, with less control alternatives the computational load is diminished. In the robust MPC case, the control horizon is generally equal with the prediction horizon $N_u = N$, as the cumulative effect of the worst case disturbances needs an important control counterpart.

$$\min_{u_t}\left\{\max_{v_t}\left\{\min_{u_{t+1}}\cdots\min_{u_{t+N_u-1}}\left\{\max_{v_{t+N_u},\ldots,v_{t+N-1}}\right.\right.\right. \\ \left.\left.\left. S_{P_\lambda}(x_{t+N|t}) + \sum_{k=1}^{N-1}\left\|Qx_{t+k|t}\right\|_\infty + \sum_{k=0}^{N_u-1}\left\|Ru_{t+k}\right\|_\infty \right\}\cdots\right\} \tag{6}$$

This constrained optimization provides a robust control sequence but is quite conservative as it is con-

sidered for all disturbance realizations ignoring that measurements are available as time progresses. The control potential is improved if a feedback approach is adopted resulting in a nested min-max formulation:

$$\min_{u_t}\{\|Ru_t\|_\infty + \max_{v_t}\left\{\left\|Qx_{t+1|t}\right\|_\infty + \right.$$
$$+\min_{u_{t+1}}\{\ldots + \min_{u_{t+N_u-1}}\left\{\|Ru_{t+N_u-1}\|_\infty + \right.$$
$$\left.\left.+\max_{v_{t+N_u-1},\ldots,v_{t+N}}\left\{S_{P_\lambda}(x_{t+N|t}) + \sum_{k=N_u}^{N-1}\left\|Qx_{t+k|t}\right\|_\infty\right\}\ldots\right\}\right.$$

This represents a closed loop formulation as mentioned in studies like (Scokaert and Mayne, 1998). A great advantage is the avoidance of the feasibility problems in comparison with the open-loop formulations.

3 ROBUST MPC AS A MULTI-PARAMETRIC OPTIMIZATION

The robust model predictive control problem formulated before is based on the on-line solving of the associated min-max optimization problem:

$$\begin{aligned}&\min_{\mathbf{k_u}}\max_{\mathbf{k}_v} J(x_t,\mathbf{k_u},\mathbf{k}_v)\\ &\text{subj. to } F_{in}\mathbf{k_u}+G_{in}\mathbf{k}_v\leqslant h_{in}+H_{in}x_t\end{aligned}\tag{7}$$

with $\mathbf{k_u}=\left\{u_t,..,u_{t+N_u-1}\right\}$, $\mathbf{k}_v=\{v_t,..,v_{t+N-1}\}$ and a convex cost function $J(x_t,\mathbf{k_u},\mathbf{k}_v)$ based on a sum of ∞-norm terms. $F_{in},G_{in},h_{in},H_{in}$ translate in a compact form the set of constraints in (5). Both the cost function and the set of constraints depend on the current state vector x_t which plays the role of a parameter. This parameterization of the optimization problem to be solved at each sampling time transforms the on-line location of the minimum argument in a computationally prohibitive task. The alternative solution is to explicitly formulate off-line the optimal solution $\mathbf{k}^*_{\mathbf{u}}(x_t)$ in terms of a piecewise linear function and further evaluate this function on-line.

3.1 The Inner Optimization

The influence of the disturbances in the form (7) can be examined by the reconsideration of the extremal possible combination of vertices in V for each prediction stage completing the sequence $\mathbf{k}_v$.

$$v_t\in V\subset\mathbb{R}^p\Rightarrow\mathbf{k}_v\in V^N\subset\mathbb{R}^{N\times p}\tag{8}$$

Remark: For the inner optimization, the set of constraints is constituted only by the inequalities defining the polyhedral domain as in (3) and the constraints imposed by the system dynamics in (1). This fact is transparent from the definition of the predictive control law, which allows any combination of disturbances satisfying (3). If one of these combinations is not allowed by the set of constraints in (7), it means in fact that the MPC law is infeasible.

Taking into account the convexity of the objective function and the previous remark, it can be concluded that the optimum for the inner optimization in (7) is on the border of the feasible domain, more precisely on one of the vertices of V^N as long as it is defined as a polytope. Thus (7) becomes:

$$\begin{aligned}&\min_{\mathbf{k_u}}\max_{\mathbf{v}_{\mathbf{k}_v}} J(x_t,\mathbf{k_u},\mathbf{k}_{v_l})\\ &\text{subj. to } F_{in}\mathbf{k_u}+G_{in}\mathbf{k}_{v_l}\leqslant h_{in}+H_{in}x_t\\ &l\in L,\mathbf{k}_{v_l}\in V^N\end{aligned}\tag{9}$$

with $L=\{1,2,\ldots,N_v\}$ and N_v the number of vertices in V^N.

This means that the inner optimization in (7) will act only on the set of vertices in V^N. Further this may be written as:

$$\begin{aligned}&\min_{\mathbf{k_u},\mu}\mu\\ &\text{subj. to } F_{in}\mathbf{k_u}+G_{in}\mathbf{k}_{v_l}\leqslant h_{in}+H_{in}x_t\\ &J(x_t,\mathbf{k_u},\mathbf{k}_{v_l})\leqslant\mu\\ &l\in L,\mathbf{k}_{v_l}\in V^N\end{aligned}\tag{10}$$

3.2 The Outer Optimization Problem

An impediment in finding the explicit solution for (7) is the expression of the cost function, given as a collection of ∞-norm terms. In order to avoid the inherent difficulty of handling it, an equivalent linear program (LP) (Kerrigan, 2004) formulation must be achieved based on the idea that each ∞-norm term can be bounded. The optimization problem is equivalent with the minimization of the sum of these bounds. This is resumed by the following result where the cost function is considered as a sum of ∞-norm terms linear in the vector of unknowns $\mathbf{x}$ and parameters $\mathbf{p}$ (to identify them, one can observe that for a fix sequence $\mathbf{k_v}=ct$ and noting $\mathbf{x}=\mathbf{k_u}$ and $\mathbf{p}=x_t$ in (7), the cost function is a sum of $\|S_i\mathbf{x}+P_i\mathbf{p}+s_i\|_\infty$ terms, with S_i,P_i,s_i defined after case).

Proposition 1. The formulations (1) and (2) are equivalent:

$$\begin{aligned}&K(\mathbf{p})=\arg\min_{\mathbf{x}}J(\mathbf{x},\mathbf{p})=\\ (1)\qquad&=\arg\min_{\mathbf{x}}\sum_{i=1}^{n}\|S_i\mathbf{x}+P_i\mathbf{p}+s_i\|_\infty\\ &\text{subject to } A_{in}\mathbf{x}\leqslant b_{in}+B_{in}\mathbf{p}\end{aligned}$$

$$K(\mathbf{p}) = \underset{\rho,\{\sigma_i\},\mathbf{x}}{\arg\min}\ \rho$$
$$(2)\quad \text{subject to} \begin{cases} \text{-}\ \mathbf{1}\sigma_i \leqslant S_i\mathbf{x} + P_i\mathbf{p} + s_i \leqslant \mathbf{1}\sigma_i, \\ \qquad\qquad 1 \leqslant i \leqslant n \\ \sum_{i=1}^{n} \sigma_i \leqslant \rho \\ A_{in}\mathbf{x} \leqslant b_{in} + B_{in}\mathbf{p} \end{cases}$$

where $\sigma_i, \rho \in \mathbb{R}$ and $\mathbf{1}$ is a vector with unit entries.

3.3 RMPC Multi-parametric Optimization Problem

With the previous two transformations, the optimization (7) can be rewritten as:

$$\mathbf{k_u}*(x_t) = \underset{\rho,\mathbf{k_u},\{\sigma_i^j\}}{\arg\min}\ \rho$$
$$\begin{cases} \text{-}\ \mathbf{1}\sigma_i^j \leqslant S_i\mathbf{k_u} + P_i x_t + W_i\mathbf{k}_{v_l} + s_i \leqslant \mathbf{1}\sigma_i^j, \\ \qquad\qquad 1 \leqslant i \leqslant n, 1 \leqslant l \leqslant N_v \\ \begin{bmatrix} \sum_{i=1}^{n} \sigma_i^1 \\ \vdots \\ \sum_{i=1}^{n} \sigma_i^{N_v} \end{bmatrix} \leqslant \mathbf{1}\rho \\ F_{in}\mathbf{k_u} + G_{in}\mathbf{k}_{v_l} \leqslant h_{in} + H_{in}x_t, \\ \qquad\qquad 1 \leqslant l \leqslant N_v \end{cases} \quad (11)$$

Example 1: To illustrate these transformations, consider the parameter-free optimization (Figure 1):

$$\min_{x_1}\max_{x_2} \left\| \begin{array}{c} 2x_1 + x_2 - 3 \\ x_1 - x_2 + 1 \end{array} \right\|_\infty + \left\| \begin{array}{c} x_1 - 2x_2 + 1 \\ 2x_1 + 3x_2 - 7 \end{array} \right\|_\infty$$
$$\text{subject to} \begin{cases} x_2 \in [\text{ - 1,1}] \\ x_1 \in [0, 6] \end{cases}$$

Using the previous transformation this is equivalent with:

$$\min_{x_1,\sigma_1,\sigma_2,\sigma_3,\sigma_4,\rho} \rho$$
$$\text{s.t. -} \begin{bmatrix} \sigma_1 \\ \sigma_1 \end{bmatrix} \leqslant \begin{bmatrix} 2x_1 - 2 \\ x_1 \end{bmatrix} \leqslant \begin{bmatrix} \sigma_1 \\ \sigma_1 \end{bmatrix};$$
$$\text{-} \begin{bmatrix} \sigma_2 \\ \sigma_2 \end{bmatrix} \leqslant \begin{bmatrix} 2x_1 - 4 \\ x_1 + 2 \end{bmatrix} \leqslant \begin{bmatrix} \sigma_2 \\ \sigma_2 \end{bmatrix};$$
$$\text{-} \begin{bmatrix} \sigma_3 \\ \sigma_3 \end{bmatrix} \leqslant \begin{bmatrix} x_1 - 1 \\ 2x_1 - 4 \end{bmatrix} \leqslant \begin{bmatrix} \sigma_3 \\ \sigma_3 \end{bmatrix};$$
$$\text{-} \begin{bmatrix} \sigma_4 \\ \sigma_4 \end{bmatrix} \leqslant \begin{bmatrix} x_1 + 3 \\ 2x_1 - 10 \end{bmatrix} \leqslant \begin{bmatrix} \sigma_4 \\ \sigma_4 \end{bmatrix};$$
$$\sigma_1 + \sigma_3 \leqslant \rho; \sigma_2 + \sigma_4 \leqslant \rho; x_1 \in [0, 6]$$

which can be tackled by any LP solver with solution:

$$[x_1\ \sigma_1\ \sigma_2\ \sigma_3\ \sigma_4\ \rho]^* = [2.33\ 4.33\ 5.33\ 1.33\ 2.66\ 9.66]$$

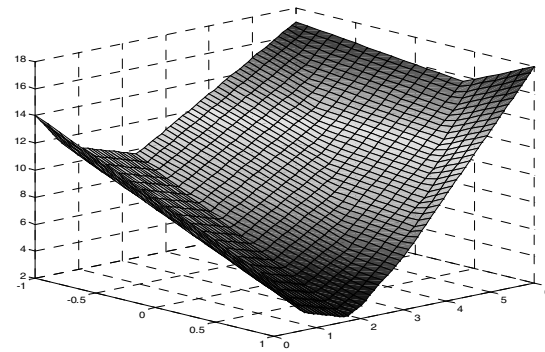


Figure 1: Cost function for example 1.

4 THE EXPLICIT SOLUTION

In the following, the close form of the RMPC law is the main objective. It can be expressed as a function of parameters if a procedure of describing the explicit solution of multi-parametric linear programs (MPLP) is available. The literature on MPLP contains the works of Gal and Nedoma (Gal and Nedoma, 1972) and further developments to linear, quadratic, non-linear or mixed-integer solvers (Borelli, 2003). Another procedure will be proposed here focusing on the set of constraints and its geometrical representation. The feasible domain will be expressed as a parametrized polyhedron. Due to the reformulation of the optimization problem, the use of mixed variables is avoided. Thus the resulting algorithm differs from the solutions based on branch and bound methods or other mixed integer linear solvers, being mainly focused on the enumeration of the edges of an augmented dimension polyhedron.

4.1 Parameterized Polyhedra

A system of linear constraints defines a polyhedron:

$$P = \{\mathbf{x} \in \mathbb{R}^n \,|\mathbf{A}_{eq}\ \mathbf{x} = \mathbf{b}_{eq}; \mathbf{A}_{in}\mathbf{x} \leqslant \mathbf{b}_{in}\} \quad (12)$$

by dual Minkowski representation of generators (Schrijver, 1986):

$$P = conv.hull\,\{\mathbf{x}_1, \ldots, \mathbf{x}_v\} + \\ +cone\,\{\mathbf{y}_1, \ldots, \mathbf{y}_r\} + lin.space\mathbf{Z} \quad (13)$$

where $conv.hullX$ denotes the set of convex combinations of points in X, $coneY$ denotes nonnegative

combinations of unidirectional rays and $lin.spaceZ$ represents a linear combination of bidirectional rays. It can be rewritten as:

$$P = \left\{ \mathbf{x} | \, \mathbf{x} = \sum_{i=1}^{v} \lambda_i \mathbf{x}_i + \sum_{i=1}^{r} \gamma_i \mathbf{y}_i + \sum_{i=1}^{l} \mu_i \mathbf{z}_i \right\}$$
$$0 \leqslant \lambda_i \leqslant 1, \ \sum_{i=1}^{v} \lambda_i = 1 \ , \ \gamma_i \geqslant 0 \ , \ \forall \mu_i \tag{14}$$

Remark: The generators saturate all the equalities, the lines saturate all the constraints and only the rays and the vertices can verify but not saturate a part of the inequalities.

The geometrical computations might be burdened by the differences that have to be taken into consideration between rays and lines. These problems are overcome with an *homogenous* representation (Wilde, 1993):

$$D = \left\{ \begin{pmatrix} \xi \mathbf{x} \\ \xi \end{pmatrix} \in \mathbb{R}^{n+1} \ \middle| \ \begin{array}{l} \hat{\mathbf{A}}_{eq} \begin{pmatrix} \xi \mathbf{x} \\ \xi \end{pmatrix} = 0 \\ \hat{\mathbf{A}}_{in} \begin{pmatrix} \xi \mathbf{x} \\ \xi \end{pmatrix} \geqslant 0 \end{array} \right\} \tag{15}$$

$$\hat{\mathbf{A}}_{eq} = [\ \mathbf{A}_{eq} | -\mathbf{b}_{eq} \ ] \quad \hat{\mathbf{A}}_{in} = \left[\begin{array}{c|c} \mathbf{A}_{in} & -\mathbf{b}_{in} \\ \hline 0 \cdots 0 & 1 \end{array} \right] \tag{16}$$

The original polyhedron P is found intersecting D with the hyper-plane of equation $\xi = 1$. Following the same change of dimension, the rays, vertices and lines have a similar unified homogenous description:

$$\hat{\mathbf{Y}} = \left[\begin{array}{c|c} \mathbf{Y} & \mathbf{X} \\ \hline 0 \cdots 0 & 1 \cdots 1 \end{array} \right] ; \hat{\mathbf{Z}} = \left[\begin{array}{c} \mathbf{Z} \\ \hline 0 \cdots 0 \end{array} \right] \tag{17}$$

and the generators representation will be:

$$D = \left\{ \begin{pmatrix} \xi \mathbf{x} \\ \xi \end{pmatrix} \middle| \ \begin{pmatrix} \xi \mathbf{x} \\ \xi \end{pmatrix} = \hat{\mathbf{Y}} \, \lambda' + \hat{\mathbf{Z}} \, \mu; \lambda' \geqslant 0 \right\} \tag{18}$$

A parameterized polyhedron is defined in the implicit form by a finite number of inequalities and equalities but the affine part depends linearly on a parameter vector $\mathbf{p}$ for both equalities and inequalities:

$$P'(\mathbf{p}) = \{ \mathbf{x} \in \Re^n \, | \mathbf{A}_{eq} \ \mathbf{x} = \mathbf{B}_{eq} \mathbf{p} + \mathbf{b}_{eq}; \ \mathbf{A}_{in} \mathbf{x} \leqslant \mathbf{B}_{in} \mathbf{p} + \mathbf{b}_{in} \} =$$
$$= \left\{ \mathbf{x}(\mathbf{p}) | \ \mathbf{x}(\mathbf{p}) = \sum_{i=1}^{v} \lambda_i(\mathbf{p}) \mathbf{x}_i(\mathbf{p}) + \sum_{i=1}^{r} \gamma_i \mathbf{y}_i + \sum_{i=1}^{l} \mu_i \mathbf{z}_i \right\}$$
$$0 \leqslant \lambda_i(\mathbf{p}) \leqslant 1, \ \sum_{i=1}^{v} \lambda_i(\mathbf{p}) = 1 \ , \ \gamma_i \geqslant 0 \ , \ \forall \mu_i \tag{19}$$

where $\mathbf{z}_i$ are the lines, $\mathbf{y}_i$ are the rays, $\mathbf{x}_i$ are the vertices and $\mu_i, \gamma_i, \lambda_i$ the corresponding coefficients.

Remark: Only the vertices are concerned by the parameterization of the polyhedron (*parameterized vertices* $\mathbf{x}_i(\mathbf{p})$), the rays and the lines do not change with the parameters' variation.

The parameterized polyhedron $P'(\mathbf{p})$ can be written as a non-parameterized polyhedron in an augmented space as:

$$\tilde{P}' = \left\{ \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix} \in \mathbb{R}^{n+m} \middle| \ \begin{array}{r} [\mathbf{A}_{eq} | -\mathbf{B}_{eq}] \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix} = \mathbf{b}_{eq} \\ [\mathbf{A}_{in} | -\mathbf{B}_{in}] \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix} = \mathbf{b}_{in} \end{array} \right\} \tag{20}$$

with a homogenous representation given by:

$$\tilde{D} = \left\{ \begin{pmatrix} \xi \mathbf{x} \\ \xi \mathbf{p} \\ \xi \end{pmatrix} \middle| \ \begin{array}{l} \tilde{\mathbf{A}}_{eq} \begin{pmatrix} \xi \mathbf{x} \\ \xi \mathbf{p} \\ \xi \end{pmatrix} = 0 \\ \tilde{\mathbf{A}}_{in} \begin{pmatrix} \xi \mathbf{x} \\ \xi \mathbf{p} \\ \xi \end{pmatrix} \geqslant 0 \end{array} \right\} =$$
$$= \left\{ \begin{pmatrix} \xi \mathbf{x} \\ \xi \mathbf{p} \\ \xi \end{pmatrix} \middle| \ \begin{pmatrix} \xi \mathbf{x} \\ \xi \mathbf{p} \\ \xi \end{pmatrix} = \tilde{\mathbf{Z}} \tilde{\lambda} + \tilde{\mathbf{Y}} \tilde{\mu}; \ \tilde{\mu} \geqslant 0 \right\} \tag{21}$$

where $\tilde{\mathbf{Y}}, \tilde{\mathbf{Z}}$ are as in (17), the matrices:

$$\tilde{\mathbf{A}}_{eq} = [\mathbf{A}_{eq} | \ -\mathbf{B}_{eq} | - \mathbf{b}_{eq}] \ ;$$
$$\tilde{\mathbf{A}}_{in} = \left[\begin{array}{c|c|c} \mathbf{A}_{in} & -\mathbf{B}_{in} & -\mathbf{b}_{in} \\ \hline 0 \cdots 0 & 0 \cdots 0 & 1 \end{array} \right]$$

and $\tilde{\lambda}, \ \tilde{\mu}$ are free-valued column vectors.

The form (19) faces an important difficulty as it contains unknown parts, i.e. the parameterized vertices $\mathbf{x}_i(\mathbf{p})$.

The parameterized vertices correspond to *m*-polyhedra in the augmented (data($\mathbb{R}^n$)+parameter($\mathbb{R}^m$)) space as in (20); consequently the original vertices are:

$$\mathbf{x}_i(\mathbf{p}) = \text{Proj}_n \left(F_i^m(\tilde{P}') \cap S(\mathbf{p}) \right) \tag{22}$$

where $\text{Proj}_x(.)$ projects the combined space $\mathbb{R}^{n+m}$ onto the original space $\mathbb{R}^n$ and $S(\mathbf{p})$ is the affine subspace:

$$S(\hat{\mathbf{p}}) = \left\{ \begin{pmatrix} \mathbf{x} \\ \mathbf{p} \end{pmatrix} \in \mathbb{R}^{n+m} \middle| \, \mathbf{p} = \hat{\mathbf{p}} \right\} \tag{23}$$

and $F_i^m(\tilde{P}')$ is a m-face of $\tilde{P}'$ found as the intersection between $\tilde{P}'$ and the supporting hyperplanes (Loechner and Wilde, 1997).

For each face of the polyhedron $\tilde{P}'$, a set of active constraints is well defined, resulting in the fact

that each point $\begin{pmatrix} \mathbf{x}_i(\mathbf{p})^{\mathrm{T}} & \mathbf{p}^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}} \in F_i^m(\tilde{P}')$ lies in a subspace of dimension *m* and thus **x** and **p** are related by:

$$\begin{bmatrix} \mathbf{A}_{eq} \\ \bar{\mathbf{A}}_{in_i} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{A}'_{eq} \\ \bar{\mathbf{B}}'_{in_i} \end{bmatrix} \mathbf{p} + \begin{bmatrix} \mathbf{b}_{eq} \\ \bar{\mathbf{b}}_{in_i} \end{bmatrix} \tag{24}$$

where $\bar{\mathbf{A}}_{in_i}, \bar{\mathbf{B}}'_{in_i}, \bar{\mathbf{b}}_{in_i}$ are the subset of the inequalities defined previously, satisfied by saturation. If the matrix $\begin{bmatrix} \mathbf{A}_{eq}^{\mathrm{T}} & \bar{\mathbf{A}}_{in_i}^{\mathrm{T}} \end{bmatrix}^T$ is not invertible, it corresponds to faces $F_i^m(\tilde{P}')$where for one given p more than one point $\mathbf{x} \in \mathbb{R}^n$ is feasible and such combinations do not match a vertex of $P'(\mathbf{p})$. In fact this case corresponds to the zones where $P'(\mathbf{p})$ changes its shape.

In the invertible case, the dependencies could be rewritten:

$$\begin{aligned} \mathbf{x}_i(\mathbf{p}) = \begin{bmatrix} \mathbf{A}_{eq} \\ \bar{\mathbf{A}}_{in_i} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{B}'_{eq} \\ \bar{\mathbf{B}}'_{in_i} \end{bmatrix} \mathbf{p} + \\ + \begin{bmatrix} \mathbf{A}_{eq} \\ \bar{\mathbf{A}}_{in_i} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{b}_{eq} \\ \bar{\mathbf{b}}_{in_i} \end{bmatrix} \end{aligned} \tag{25}$$

For the implementation of these theoretical results, an enumeration of the m-faces must be available together with the $k(> m)$ generators of each face $F_i^m(\tilde{D})$ in a homogenous representation. If the projections:

$$Pr_n \begin{pmatrix} \xi \mathbf{x}_i(\mathbf{p}) \\ \xi \mathbf{p} \\ \xi \end{pmatrix} = \begin{pmatrix} \xi \mathbf{x}_i(\mathbf{p}) \\ \xi \end{pmatrix}; \tag{26}$$

$$Pr_m \begin{pmatrix} \xi \mathbf{x}_i(\mathbf{p}) \\ \xi \mathbf{p} \\ \xi \end{pmatrix} = \begin{pmatrix} \xi \mathbf{p} \\ \xi \end{pmatrix} \tag{27}$$

are defined, then (22) could be rewritten as:

$$\begin{aligned} &\begin{pmatrix} \xi \mathbf{x}_i(\mathbf{p}) \\ \xi \end{pmatrix} = \underset{n}{\mathrm{Pr}}(F_i) \underset{m}{\mathrm{Pr}}(F_i)^{-1} \begin{pmatrix} \mathbf{p} \\ \xi \end{pmatrix}; \\ &F_i = \left[\begin{pmatrix} \xi \mathbf{x}_{ij}(\mathbf{p}) \\ \xi \mathbf{p} \\ \xi \end{pmatrix} \right], j = 1..k \end{aligned} \tag{28}$$

The case when the right inverse $Pr_m(F_i)^{-1}$ does not exist results in the already mentioned conditions of an m-face that does not define a unique vertex of $P'(\mathbf{p})$.

Remark: Numerical methods (Leverge, 1994) exist for implementing the double description of polyhedra. The polyhedral duality allows both transformations, from constraints to generators and conversely ((Leverge, 1994), (Loechner and Wilde, 1997), (Motzkin et al., 1953), (Schrijver, 1986), (Wilde, 1993)).

4.2 Explicit Solution of LP

Recalling the problem to be solved similar to (11):

$$\begin{aligned} &x^*(\mathbf{p}) = \arg\min_{\mathbf{x}} f^T\mathbf{x} \\ &\text{subject to } A_{in}\mathbf{x} \leqslant B_{in}\mathbf{p} + b_{in} \end{aligned} \tag{29}$$

with the optimal solution as a piecewise affine function of the parameter.

Consider now a fixed parameter $\mathbf{p}_{ct}$. When analyzing the optimization problem (29) corresponding to this value, a geometrical point of view can be used, as in Chernikova algorithm (Leverge, 1994).

Proposition 2. For a linear problem three cases may arise:

a) If the associated polyhedron $P = \{x | A_{in}\mathbf{x} \leqslant B_{in}\mathbf{p}_{ct} + b_{in}\}$ is empty, the problem is infeasible;

b) If there exists a bidirectional ray $\mathbf{z}$ such that $f^T\mathbf{z} \neq 0$ or there exists a unidirectional ray $\mathbf{y}$ such that $f^T\mathbf{y} \leqslant 0$, then the minimum is unbounded;

c) If all bidirectional rays $\mathbf{z}$ are such that $f^T\mathbf{z} = 0$ and all unidirectional rays $\mathbf{y}$ are such that $f^T\mathbf{y} \geqslant 0$, then the minimum is defined by: $\min\left\{f^T\mathbf{x}_i | \mathbf{x}_i \text{ vertex of } P\right\}$ and the solution is:

$$\begin{aligned} S = conv.hull\left\{\mathbf{x}'_1, \ldots, \mathbf{x}'_s\right\} + \\ + cone\left\{\mathbf{y}'_1, \ldots, \mathbf{y}'_r\right\} + lin.spaceP \end{aligned}$$

where $\mathbf{x}'_i$ are the vertices attaining the minimum and y'_i are such that $f^T\mathbf{y}'_i = 0$.

Now extending this perspective to the multi-parametric case for each $\mathbf{p} \in \mathbb{R}^n$, a similar result can be established.

Proposition 3. The solution of a multi-parametric linear optimization problem is characterized by the followings:

a) If there exists a bidirectional ray $\mathbf{z}$ such that $f^T\mathbf{z} \neq 0$ or there exists a unidirectional ray $\mathbf{y}$ such that $f^T\mathbf{y} \leqslant 0$, then the minimum is unbounded;

b) For the sub domains of the parameter space $D_{ifez} \in \mathbb{R}^n$ with the associated polyhedron $P = \{\mathbf{x} | A_{in}\mathbf{x} \leqslant B_{in}\mathbf{p} + b_{in}\}$ empty while $\mathbf{p} \in D_{ifez}$, the problem is infeasible (this can be restated in terms of parameterized vertices: "for the sub domains where no parameterized vertex is available, the problem is infeasible");

c) If all bidirectional rays $\mathbf{z}$ are such that $f^T\mathbf{z} = 0$ and all unidirectional rays $\mathbf{y}$ are such that $f^T\mathbf{y} \geqslant 0$, then the sub domains D_k can be defined such that the minimum:

$$\min\left\{f^T\mathbf{x}_i(\mathbf{p}) | \mathbf{x}_i(\mathbf{p}) \text{ vertex of } P(\mathbf{p})\right\}$$

is attained by the same subset of vertices of . The complete solution for this sub domain is:

$$\begin{aligned} S_k(\mathbf{p}) = conv.hull\left\{\mathbf{x}'_{1k}(\mathbf{p}), \ldots, \mathbf{x}'_{sk}(\mathbf{p})\right\} + \\ + cone\left\{\mathbf{y}'_1, \ldots, \mathbf{y}'_r\right\} + lin.spaceP(\mathbf{p}) \end{aligned}$$

where $\mathbf{x}'_i$ are the vertices corresponding to the minimum and $\mathbf{y}'_i$ are such that $f^T\mathbf{y}'_i = 0$.

One has to observe that our goal is to find the explicit solution for the LP derived from the optimization problem in robust MPC which has some particularities:

- The linearity space is empty since the cost function is positive convex.
- There is no unidirectional ray such that because this will imply that the cost function is not convex.
- A single value in $S_k(\mathbf{p})$ is to be used on-line in MPC.

Proposition 4. The solution of a multi-parametric linear optimization problem within robust MPC satisfies:

a) The problem is infeasible for the sub domains $D_{ifez} \in \mathbb{R}^n$ where no parameterized vertex is available;

b) Sub domains D_k are defined as the zones for which the solution $S_k(\mathbf{p}) = conv.hull\{\mathbf{x}'_{1k}, \ldots, \mathbf{x}'_{sk}\}$ is given by the same set of parameterized vertices satisfying:

$$f^T\mathbf{x}'_{1k} = \ldots = f^T\mathbf{x}'_{sk} =$$
$$= \min\left\{f^T\mathbf{x}_i(\mathbf{p}) | \mathbf{x}_i(\mathbf{p}) \text{ vertex of } P(\mathbf{p})\right\}$$

Remark: As the parameters in (29) vary inside the parameter space, the vertices of the optimization domain may split, shift or merge. The global optimum will follow this evolution within the parameter space as the optimum is a continuous function of parameter.

From a practical point of view the implementation of this result is direct and follows the steps:

1. Find the expression of the parameterized feasible domain in the augmented data+parameter space:
$$A_{in}\mathbf{x} \leqslant B_{in}\mathbf{p}+b_{in} \Leftrightarrow [\ \mathrm{A_{in}} | -B_{in}\ ] \begin{bmatrix} \mathbf{x} \\ \mathbf{p} \end{bmatrix} \leqslant b_{in}$$
2. Find the m-vertices where n is the dimension of the parameter space.
3. Retain only those corresponding to parameterized vertices by ignoring those with non-invertible projection on the parameter space
4. Compute validity domain D_k for each parameterized vertex
5. Compare each pair of vertices. In the case of a non-empty intersection of their validity domains, split them using the linear cost function. The final expression will be a union of regions corresponding to the parameterized vertices containing the optimum.

A special attention must be given to the step 5 with the iterative comparison of the vertices and their validity domains. A possible routine may be based on the following procedure.

```
  procedure CutDomains (VD: the set of
all validity domains)

n=cardinal (VD)
i=1; j=2
while i<n+1
while j<n+1
if VD_j ∩ VD_i ≠ ∅
if f^T x_i ⩽ f^T x_j then VD_j = VD_j − VD_i
if f^T x_j ⩽ f^T x_i then VD_i = VD_i − VD_j
j=j+1
endif
end
i=i+1
end
```

Remark: The procedure is initialized with the set of validity domains obtained after the edges' enumeration (step 2).

Remark: The difference of two convex domains is not a close operation and thus the output of the procedure is a union of convex sub domains of the parameters space which do not necessarily cover the entire $\mathbb{R}^m$ (step 4).

From the RMPC point of view, the difference:

$$\aleph = \mathbb{R}^m \setminus \{\cup D_k;\ k = 1..n_D\} \tag{30}$$

describes the regions of infeasible parameters.

Once the set of parameter space sub domains D_k created, it can be used in an on-line optimization finding the control sequence for robust MPC.

Algorithm (on-line solver)

1. Find the appurtenance set $D_k;\ k = 1..n_D$ for the current parameter p. Return infeasible if no D_k is found.
2. Compute $k_{u_{MPC}} = x_k(\mathbf{p})$ using the piecewise formulation of the parameterized vertices as in (25) and effectively apply the first component.
3. Restart from 1 with the new p.

One may remark that the evaluation mechanism for the first step of this algorithm can be logarithmic in the number of partitions (Tondel et al., 2003) and thus efficient with respect to the involving on-line optimization solvers.

5 EXAMPLE

Consider the model (Scokaert and Mayne, 1998):

$$x_{t+1} = x_t + u_t + v_t$$

In order to illustrate the ideas of RMPC presented earlier, a two step prediction is considered and thus

the following optimization problem is to be solved at each sampling time:

$$V(x_t) = \min_{u_t, u_{t+1}} \sum_{k=0}^{1} \left|x_{t+k|t}\right| + 10\left|u_{t+k}\right|$$
$$s.t. \begin{cases} \text{-}\ 1.2 \leqslant x_{t+k|t} \leqslant 2, k = 0, 1, 2 \\ \text{-}\ 1 \leqslant x_{t+2|t} \leqslant 1, \\ \text{-}\ 1 \leqslant v_{t+k} \leqslant 1, k = 0, 1 \end{cases} \quad (31)$$

Ignoring the disturbances, the explicit solution of the problem can be found using the geometrical approach presented in the previous section by inspecting the 22 parameterized vertices. After the stage of discrimination of the validity domains, the explicit RMPC law is found as:

Affine control law	Validity domain
$u_t = -x_t - 1$	$-1.2 \leqslant x_t \leqslant -1$
0	$-1 \leqslant x_t \leqslant 0$
0	$0 \leqslant x_t \leqslant 1$
$u_t = -x_t + 1$	$1 \leqslant x_t \leqslant 2$

It can be observed that there are two domains with the same control law due to the fact that the cost function changes its slope and thus the maximum lies on different parameterized vertices in the augmented space. In this case, as their union is a convex set, they can be collated in a single set. In the general case, this operation can be done using tools of convex recognition of union of polyhedra (see (Bemporad et al., 2002a) for details).

Simulating this control law for an initial condition $x_0 = -1.2$ proves to keep the system trajectory inside the constraints in the disturbance free case (Figure 2a). If the same controller is used with $v_k = -1/k,\ k \geqslant 1$, the trajectory will violate the constraints (Figure 2b).

Further if the robust MPC explicit formulation is to be achieved then the min-max version of (31) is to be solved:

$$V(x_t) = \min_{u_t, u_{t+1}} \max_{v_t, v_{t+1}} \sum_{k=0}^{1} \left|x_{t+k|t}\right| + 10\left|u_{t+k}\right|$$
$$s.t. \begin{cases} \text{-}\ 1.2 \leqslant x_{t+k|t} \leqslant 2, k = 0, 1, 2 \\ \text{-}\ 1 \leqslant x_{t+2|t} \leqslant 1, \\ \text{-}\ 1 \leqslant v_{t+k} \leqslant 1, k = 0, 1 \end{cases} \quad (32)$$

In this form, there is no solution as the optimization is infeasible. In fact there is no control law at first sampling time:

$$u_{t|t} = a_1 x_t + b_1$$
$$u_{t+1|t} = a_2 x_t + b_2 u_{t|t} + c_2$$

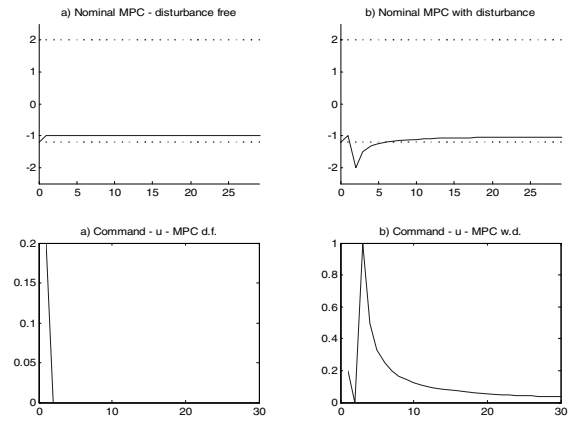


Figure 2: a) Left: Nominal MPC - disturbance-free case; b) Right: Nominal MPC for the system affected by disturbances.

which can keep robustly the system trajectory within the constraints. This fact is obvious as long as an "open-loop" type of RMPC is considered, where the cumulative damage of the disturbances can not be mitigated. When writing explicitly the end-point constraints in (32) for the extremal combinations of disturbances, this becomes evident as:

$$\begin{bmatrix} \nu_t \\ \nu_{t+1} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow$$
$$-1 \leqslant x_t + u_{t|t} + u_{t+1|t} + 2 \leqslant 1 \Rightarrow$$
$$-3 \leqslant x_t + u_{t|t} + u_{t+1|t} \leqslant -1;$$
$$\begin{bmatrix} \nu_t \\ \nu_{t+1} \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \Rightarrow$$
$$-1 \leqslant x_t + u_{t|t} + u_{t+1|t} - 2 \leqslant 1 \Rightarrow$$
$$1 \leqslant x_t + u_{t|t} + u_{t+1|t} \leqslant 3$$

which means that there is no control combination to maintain the law feasible without a prior knowledge of disturbances. However the so called "closed loop" formulation provides the necessary degrees of freedom in this sense. One has to solve:

$$V(x_t) = \min_{u_t} \max_{v_t} \min_{u_{t+1}} \max_{v_{t+1}} \sum_{k=0}^{1} \left|x_{t+k|t}\right| + 10\left|u_{t+k}\right|$$
$$s.t. \begin{cases} \text{-}\ 1.2 \leqslant x_{t+k|t} \leqslant 2, k = 0, 1, 2 \\ \text{-}\ 1 \leqslant x_{t+2|t} \leqslant 1, \\ \text{-}\ 1 \leqslant v_{t+k} \leqslant 1, k = 0, 1 \end{cases} \quad (33)$$

Following the theoretical result in Section 4, the explicit solution can be achieved by solving the inner

minimization:

$$V(x_t, u_t, v_t) = \min_{u_{t+1}} \max_{v_{t+1}} |x_t| + \\ + |x_t + u_t + \nu_t| + 10\,|u_t| + 10\,|u_{t+1}| \\ s.t. \begin{cases} \text{- } 1.2 \leqslant x_{t+k|t} \leqslant 2, k = 0, 1, 2 \\ \text{- } 1 \leqslant x_{t+2|t} \leqslant 1, \\ \text{- } 1 \leqslant v_{t+k} \leqslant 1, k = 0, 1 \end{cases} \quad (34)$$

The solution using the geometrical approach is immediate as there are exactly 2 parameterized vertices on which the minimum lies and associated control law is:

$$u_{t+1|t} = -(x_t + u_{t|t} + \nu_t) = -x_{t+1} \text{ for - } 1.2 \leqslant x_t \leqslant 2$$

Notice that the control law uses the additional information available in comparison with (32). With this result, for the outer optimization problem:

$$\min_{u_t} \max_{v_t} \; |x_t| + \left|11 x_{t+1|t}\right| + |10 u_t| \\ s.t. \begin{cases} \text{- } 1.2 \leqslant x_{t+k|t} \leqslant 2, k = 0, 1 \\ \text{- } 1 \leqslant v_t \leqslant 1 \end{cases} \quad (35)$$

the explicit solution is once more immediate as there are only two non-degenerate parameterized vertices describing the geometric locus of the minimum. Applying this RMPC law:

$$u_t = -x_t \text{ for - } 1.2 \leqslant x_t \leqslant 2$$

the system affected by disturbances is regulated to the origin (Figure 3).

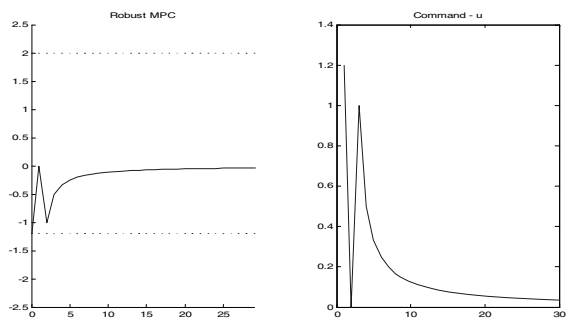


Figure 3: System trajectory with robust MPC law.

The solutions of the optimization problems in (31), (34), (35) were obtained using parameterized polyhedra routines in 2, 0.39 and 0.91 seconds respectively. However for complex system the computational time may explode as the number of parameterized vertices has an exponential dependence on the number of constraints added during the transformation stages.

6 CONCLUSION

The paper used a unified approach for the constraints handling in the context of RMPC confirming the formulation of the optimal sequence as a multi-parametric quadratic problem. The explicit solution of this problem was synthesized by means of parameterized polyhedra. This geometrical approach proposes an alternative to the recent methods presented in the literature. Its advantages might be the fact that optimum lies on the parameterized vertices providing a natural constant linear affine dependence in the context parameters. An aspect which may receive further attention is the enumeration of faces for the parameterized polyhedra which may turn to be a computationally demanding task.

REFERENCES

Bemporad, A., Borelli, F., and Morari, M. (2001). Robust model predictive control: Piecewise linear explicit solution. In *Proc. European Control Conference*.

Bemporad, A., Fukuda, K., and Torrisi, F. (2002a). Convexity recognition of the union of polyhedra. In *Computational Geometry, Vol. 18.*

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. (2002b). The explicit linear quadratic regulator for constrained systems. In *Automatica, Vol. 38.*

Borelli, F. (2003). *Constrained Optimal Control of Linear and Hybrid Systems*. Springer-Verlag, Berlin.

Gal, T. and Nedoma, J. (1972). Multiparametric linear programming. In *Management Science, 18.*

Goodwin, G., Seron, M., and Dona, J. D. (2004). *Constrained Control and Estimation*. Springer-Verlag, Berlin.

Kerrigan, E. (2000). *Robust Constraint Satisfaction: Invariant Sets and Predictive Control.* PhD Thesis, University of Cambridge.

Kerrigan, E. (2004). Feedback min-max model predictive control using a single linear program: Robust stability and the explicit solution. In *International Journal of Robust and Nonlinear Control, Vol.14*.

Kothare, M., Balakrishnan, V., and Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. In *Automatica, Vol. 32*.

Leverge, H. (1994). A note on chernikova's algorithm. In *Technical Report 635*. IRISA, France.

Loechner, V. and Wilde, D. (1997). Parameterized polyhedra and their vertices. In *Int. Journal of Parallel Programming, Vol. 25*.

Maciejowski, J. (2002). *Predictive Control with Constraints*. Prentice Hall, UK.

Motzkin, T., Raiffa, H., Thompson, G., and Thrall, R. (1953). The double description method. In *Theodore S. Motzkin: Selected Papers*. Birkhauser, Boston.

Olaru, S. and Dumur, D. (2004). A parameterized polyhedra approach for explicit constrained predictive control. In *43rd IEEE Conference on Decision and Control*.

Rossiter, J. (2003). *Model-based Predictive Control. A practical approach*. CRC Press.

Schrijver, A. (1986). *Theory of Linear and Integer Programming*. John Wiley and Sons, NY.

Scokaert, P. and Mayne, D. (1998). Min-max feedback model predictive control for constrained linear systems. In *IEEE Trans. Automatic Control, 43*.

Seron, M., Goodwin, G., and Dona, J. D. (2002). Characterisation of receding horizon control for constrained linear systems. In *Asian Journal of Control, Vol. 5*.

Tondel, P., Johansen, T., and Bemporad, A. (2003). Evaluation of piecewise affine control via binary search tree. In *Automatica, Vol. 39*.

Wilde, D. (1993). A library for doing polyhedral operations. In *Technical report 785*. IRISA, France.

A NEW HIERARCHICAL CONTROL SCHEME FOR A CLASS OF CYCLICALLY REPEATED DISCRETE-EVENT SYSTEMS

Danjing Li[1], Eckart Mayer[1], Jörg Raisch[1,2]
[1] *Systems and Control Theory Group, Max-Planck Institute, Dynamics of Complex Technical Systems*
Sandtorstrasse 1, D-39106 Magdeburg, Germany
{*danjing.li,eckart.mayer*}*@mpi-magdeburg.mpg.de*

[2] *Fachgebiet Regelungssysteme, Technische Universität Berlin*
Einsteinufer 17, D-10587 Berlin, Germany
raisch@mpi-magdeburg.mpg.de

Keywords: Cyclic systems, discrete-event systems, max-plus algebra, min-plus algebra, rail traffic.

Abstract: We extend the hierarchical control method in (Li et al., 2004) to a more generic setting which involves cyclically repeated processes. A hierarchical architecture is presented to facilitate control synthesis. Specifically, a conservative max-plus model for cyclically repeated processes is introduced on the upper level which provides an optimal online plan list. An enhanced min-plus algebra based scheme on the lower level not only handles unexpected events but, more importantly, addresses cooperation issues between sub-plants and different cycles. A rail traffic example is given to demonstrate the effectiveness of the proposed approach.

1 INTRODUCTION

In (Li et al., 2004), a hierarchical two-level control architecture has been introduced for a class of discrete-event systems (DES) without cyclically repeated features (i.e. for systems "running only one cycle"). The upper level of this architecture produces the time optimal plan based on the event time relation represented by a max-plus algebra model. The plan describes the time optimal sequence of events. If an unexpected event happens at any given time, max-plus algebra models are simulated online on the upper level to update the optimal plan and the detailed time specification for every event. Using this information, the lower control level acts as an implementation block.

In practice, many DES applications exhibit cyclically repeated features. In this case, cooperation (and competition) issues between different cycles have to be addressed, which is beyond the scope of the algorithm in (Li et al., 2004). To cope with this situation, the control strategy in (Li et al., 2004) is extended. The resulting hierarchical control scheme is depicted in Figure 1. It is composed of a two-level structure with an independent C/D (continuous/discrete) module. In general, the supervisory block on the upper level will have the goal of determining the sequence of events which optimises the given objective function in the cyclically repeated case. This sequence of events is referred to as the optimal plan. The C/D module is an interface block which transforms information from the (continuous) plant to the timed DES framework employed on the upper level of the proposed control architecture.

A case study representing a simple rail traffic system is used to demonstrate how control is realised in the proposed framework. For this example, events correspond to specific trains crossing specific locations within the track system. A plan specifies a sequence of trains and track segments where trains pass each other. The lower control level generates velocity reference signals for the trains to implement the plan determined by the supervisory block. The C/D block transforms position information into event time information. For other applications, the terms "train" and "track (segment)" have of course to be replaced by different ones. For example, in flexible manufacturing systems, "product" and "machine" can be used instead. In a general context, "train" stands for "user", "track (segment)" for "resource".

This contribution is organised as follows. Section 2 summarises how to generate the set of feasible plans for a DES without cyclically repeated processes and extends this method to the more general setting with cyclically repeated features. This section also explains how the optimal plan is then chosen in an online procedure. The C/D block is described in Section 3. Section 4 explains how the plan generated at the upper level can be implemented on the lower level with the help of min-plus algebra. A simple case study is given in Section 5 to illustrate the effectiveness of the proposed approach.

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 227–233

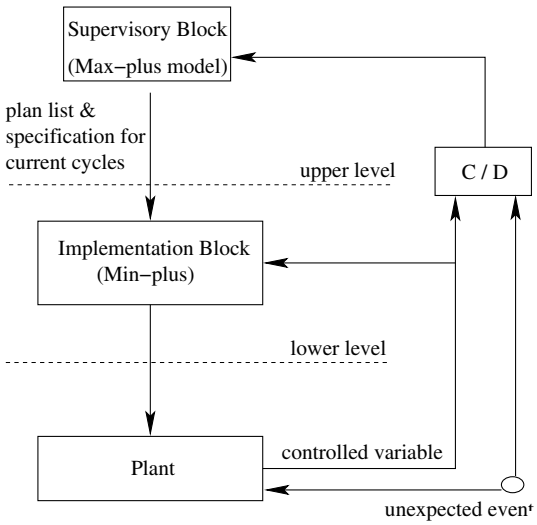


Figure 1: Control structure.

2 SUPERVISORY LEVEL

Max-plus algebra e.g. (Baccelli et al., 1992) is a structure defined on the set $\mathbb{R}^* = \mathbb{R} \cup \{-\infty\}$, with max and $+$ as the two binary operations $\oplus$ and $\otimes$, respectively. $\epsilon := -\infty$ is the additive identity in the max-plus algebra. $e := 0$ is the multiplicative identity.

Based on the max-plus algebra model of the system, the supervisory level generates the set of all feasible plans (i.e. all feasible sequences of events) as an offline task. During runtime, the supervisory level determines the time optimal plan based on real time information provided by the C/D module. The difference between systems with and without cyclicly repeated features lies in their different max-plus models. We first summarise the simpler case.

2.1 Max-plus Model for Non-cyclic Systems

Non-cyclic systems can be represented by the implicit max-plus algebra model (Baccelli et al., 1992):

$$\underline{X} = A_0 \otimes \underline{X} \oplus B \otimes u, \quad (1)$$
$$Y = C \otimes \underline{X}. \quad (2)$$

with

$$A_0^* = I \oplus A_0 \oplus A_0^2 \oplus \cdots \oplus A_0^n \oplus A_0^{n+1} \oplus \cdots, \quad (3)$$

(1), (2) can be converted to an explicit max-plus model

$$\underline{X} = A_0^* \otimes B \otimes u, \quad (4)$$
$$Y = C \otimes A_0^* \otimes B \otimes u, \quad (5)$$

Here, the i-th component of $\underline{X}$, $\underline{x}_i$, represents the earliest possible time for event i to occur. u and Y contain time information of "start" and "finish" events, and can be interpreted as system input and output, respectively.

The elements of matrix A_0 represent the minimum time distances that have to pass between events, e.g. $(A_0)_{21}$ implies that event 2 may not happen earlier than $(A_0)_{21}$ time units after event 1 happened. In this contribution, $(A_0)_{ij} \in \mathbb{R}^+ \cup \{-\infty, 0\}$. Matrix A_0 is the max-plus sum of A_{01} and A_{02}, i.e.

$$A_0 = A_{01} \oplus A_{02}, \quad (6)$$

where the former represents a property of the system and is a fixed matrix, while the latter is related to the shared resources and differs for different plans. More specifically, the elements of A_{01} represent the time relation between events determined by physical properties of the system. For example, for rail systems, with a maximum speed assumption in max-plus algebra, the entries of A_0 represent the minimum time needed by trains to pass distances in the network. Thus the elements of A_{01} correspond to lengths of tracks. On the other hand, for any instance of time, a shared track segment can only be occupied by one train. The different occupation orders for trains on shared track segments generate different plans. Different A_{02} matrices correspond to those different plans.

In a directed graph, we call the arcs corresponding to A_{01} elements "travelling arcs" while the arcs corresponding to A_{02} elements are called "control arcs". For a given train track network, the travelling arcs and therefore the entries of matrix A_{01} are fixed. Figure 2 shows the directed graph (including all possible control arcs) of a simple network with two single-line track segments and two trains, where train 1 moves from right to left and train 2 moves in opposite direction. For single-line segment I, the control arc labelled a_1, with $a_1 \geq 0$, represents the fact that the earliest time instant at which train 2 may occupy this segment is a_1 time units after train 1 has emptied it. The control arc labelled b_1 states the reverse sequence. Accordingly, only one of those two arcs can exist in any one plan. Instead of erasing a non-existent arc from a graph, we can also label it with $\epsilon = -\infty$. Hence, by definition, non-existing arcs have weight ϵ. Similarly, in segment II, the arcs labelled a_2 and b_2 represent two possibilities. Detailed information about how to find all feasible A_{02} (i.e. all feasible plans) can be found in (Li et al., 2004). In particular, it was shown there that

$$\exists k \leq n, A_0^k(i,i) > \epsilon \Leftrightarrow A_{02} \text{ is infeasible.} \quad (7)$$

2.2 Max-plus Model for Cyclicly Repeated Systems

For system with cyclicly repeated behaviour, several cycles may exist concurrently. Therefore, control

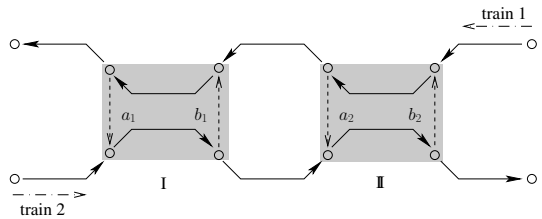


Figure 2: A network with all possible control arcs.

conditions have to be extended such that trains from different cycles will not occupy the same track segment simultaneously. Hence, additional control arcs have to be introduced. Also, additional travelling arcs are needed to represent the passage of trains into subsequent cycles. An arc that connects events related to the same cycle is a "zero order arc", an arc that connects events related to two sequential cycles is a "first order arc". In general, there can also be "second order arcs" and so on. To keep our example reasonably simple, we introduce the following conservative condition:

For each track segment shared by several cycles, trains in a latter cycle may not occupy it unless all trains in the previous cycle have left it.

With the above assumption, there is no need to introduce arcs of higher order than 1. For the simple rail traffic network in Figure 3, which was shown in (Li et al., 2004), "first order control arcs" ensure the safe resource sharing between cycles and "first order travelling arcs" ensure that each train starts its new journey (cycle $k+1$, $k \geq 1$) a specified number of time units after its arrival at its destination in cycle k. As a review of the example, the network involves 3 trains and 3 tracks. Initially, train 1, 2 and 3 are located at the end points of the 3 tracks, i.e. A, B and C, respectively. The trains move along the tracks in the directions shown in Figure 3. In the middle of both track AO and track CO, double-line track segments are available, which make it possible that two trains pass each other between points N_1 and M_1 or N_2 and M_2, respectively. In (Li et al., 2004), all trains stopped after they arrived at their destinations (i.e. after one cycle). In this contribution, each train will start a new cycle, e.g. a given number of time units after train 1 arrived at C, it will start the next route to go to B, which represents a new cycle. Figure 4 shows the different kinds of control and travelling arcs for this network. Note that although there are optional control arcs of zero order, all first order control arcs are fixed according to the assumption.

Thus, for systems with cyclicly repeated behaviour, in addition to the zero order matrix A_0, we now have a first order matrix A_1. For cycle k, the implicit max-

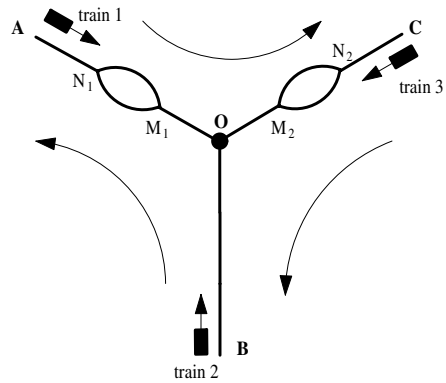


Figure 3: A simple rail traffic network.

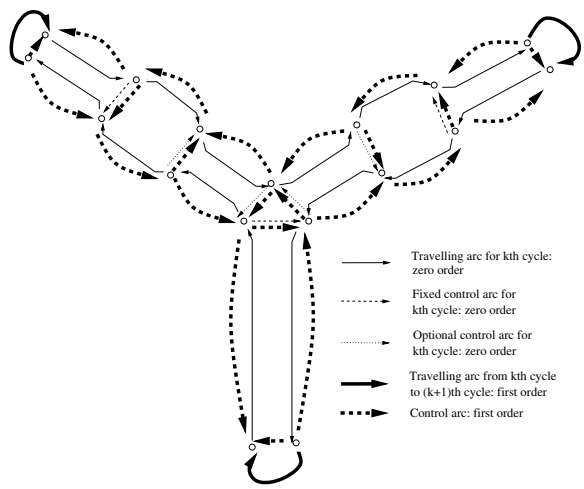


Figure 4: Control arcs and travelling arcs.

plus algebra model is:

$$\underline{X}(k) = A_0(k)\underline{X}(k) \oplus A_1\underline{X}(k-1) \oplus Bu \quad (8)$$
$$Y(k) = C \otimes \underline{X}(k) \quad (9)$$

As for A_0, the matrix A_1 is the max-plus sum of two matrices A_{11} and A_{12}, i.e.

$$A_1 = A_{11} \oplus A_{12}, \quad (10)$$

where the elements of A_{11} correspond to "first order travelling arcs" representing the time needed for a train to start a new cycle after finishing the previous cycle. The elements of A_{12} correspond to the control arcs for trains belonging to two sequential cycles respectively. Unlike A_{02}, A_{12} cannot represent different choices; therefore, A_1 is a constant matrix while A_0 will depend on the specific plan to be implemented in the k-th cycle and is therefore varying with k.

Repeated insertion of (8) into itself provides:

$$\begin{aligned}\underline{X}(k) &= A_0(k)\otimes\underline{X}(k)\oplus A_1\otimes\underline{X}(k-1)\oplus B\otimes u\\ &= A_0(k)\otimes[A_0(k)\otimes\underline{X}(k)\oplus A_1\otimes\underline{X}(k-1)\\ &\quad\oplus B\otimes u]\oplus A_1\otimes\underline{X}(k-1)\oplus B\otimes u\\ &= A_0^2(k)\otimes\underline{X}(k)\oplus[A_0(k)\oplus I]\otimes A_1\\ &\quad\otimes\underline{X}(k-1)\oplus[A_0(k)\oplus I]\otimes B\otimes u\\ &= A_0^2(k)\underline{X}(k)\oplus[A_0(k)\oplus I]A_1\underline{X}(k-1)\\ &\quad\oplus[A_0(k)\oplus I]Bu\\ &\vdots\\ &= A_0^n(k)\underline{X}(k)\oplus[A_0^{n-1}(k)\oplus\cdots\oplus A_0(k)\\ &\quad\oplus I]A_1\underline{X}(k-1)\oplus[A_0^{n-1}(k)\oplus\cdots\\ &\quad\oplus A_0(k)\oplus I]Bu \qquad (11)\end{aligned}$$

As for physically meaningful systems, $A_0^n(k)$ will only contain ϵ elements, the last identity represents an explicit max-plus algebra model:

$$\begin{aligned}\underline{X}(k) &= A_0^*(k)\otimes A_1\otimes\underline{X}(k-1)\oplus\\ &\quad A_0^*(k)\otimes B\otimes u \qquad (12)\\ Y(k) &= C\otimes\underline{X}(k), \qquad (13)\end{aligned}$$

where

$$A_0^*(k)=[A_0^{n-1}(k)\oplus\cdots\oplus A_0(k)\oplus I]. \qquad (14)$$

Assume that the system input u contains initial values of lower bounds for the state vector $X(k)$, i.e. $B=I$ ($B_{ii}=e, B_{ij}=\epsilon$ for $i\neq j$), $u=\underline{X}\_in(k)$ (see Section 3 for further information), (12) and (13) become

$$\begin{aligned}\underline{X}(k) &= A(k)\otimes\underline{X}(k-1)\oplus\\ &\quad A_0^*(k)\otimes\underline{X}\_in(k) \qquad (15)\\ Y(k) &= C\otimes\underline{X}(k) \qquad (16)\end{aligned}$$

where $A(k)$ depends on $A_0(k)$:

$$A(k)=A_0^*(k)A_1. \qquad (17)$$

For a given system with cyclicly repeated behaviour, (7) can be applied to eliminate infeasible plans and corresponding max-plus models.

2.3 Optimal Plan List

After all feasible plans have been generated, the supervisory level may simulate feasible max-plus-models (15), (16) over the required total number of cycles to determine the optimal sequence of $A(k)$, i.e. the optimal plan list. This is initially done in an offline fashion, i.e. before the system is started. The objective function is typically evaluated from the output vector. For example, for rail track systems, the (offline) objective function could be

$$J_{offline}=\min_{\substack{\text{all feasible}\\ \text{plan lists}}}\Big(\bigoplus_i Y_i(N)\Big) \qquad (18)$$

where N is the required total number of cycles and $Y_i(N)$ is the arrival time of train i in the N-th cycle. As $\oplus$ represents max, the physical meaning is to choose the plan list giving the minimum final arrival time, i.e. the minimal arrival time of the last train in the last cycle. Since it is offline simulation, $\underline{X}\_in(1)$ only carries the starting event time. Furthermore, $(\underline{X})_i(0)$ is set to $\epsilon=-\infty$. The resulting optimal plan list is then implemented via the lower control level.

Max-plus simulation in this step is obviously based on the assumption that there is no unexpected event and that each train either waits for a synchronisation condition to be met or otherwise moves at its maximum velocity. Thus if any unexpected incident happens, it may affect the travelling time of trains either directly by blocking them or indirectly via synchronisation with other trains. If this happens, the runtime event time will not match the event time calculated by a-priori model simulation any more. To address such difficulties, an online simulation is integrated to update the aforementioned optimal plan list.

At time t_k, for each of the concurrently existing cycles l (e.g. $l=k,k+1$), $[\underline{X}(l)](t_k),[Y(l)](t_k)$ will be updated by using (15), (16) with $[\underline{X}\_in(l)](t_k)$ provided by the C/D block.

The online objective may be the same as the offline objective (18), i.e.

$$J_{online1}=J_{offline}. \qquad (19)$$

It may also be different from the offline objective, for example, it may state that after unexpected delays, all trains have to catch up with the timetable corresponding to the original offline plan list as fast as possible. In this case,

$$J_{online2}=\min_{\substack{\text{all feasible}\\ \text{plan lists}}}K_\triangle \qquad (20)$$

where $K_\triangle$ is the index of the earliest cycle in which all delayed trains can start (or end) their cycle trips according to the timetable, i.e.

$$\begin{cases}\forall i,\triangle_i(l)=0. & l\geq K_\triangle\\ \exists i,\triangle_i(l)>0. & l<K_\triangle\end{cases} \qquad (21)$$

$$\triangle_i(l)=t_{start_i}(l)-timetable_i(l) \qquad (22)$$

where $t_{start_i}(l)$ is the actual cycle starting time of train i in cycle l, $timetable_i(l)$ is the cycle starting time (according to the timetable) of train i in cycle l.

In brief, the supervisory level involves both offline and online parts. The offline process is used to find all feasible plans based on the system topology and an offline optimal plan list, which can be interpreted as a timetable. The online part updates the feasible plan set (see (Li et al., 2004)), the optimal plan list and provides the time specification $\underline{X}(k)$ to the lower level.

3 C/D BLOCK

In order to reschedule the plan (using (15)-(16)) in a real time fashion, the state vector needs to be reinitialised according to the current status (at time t_k) of the trains for online model simulation. In other words, the continuous variables will be transformed into a reinitialised state vector. The reinitialised state vector $\underline{X}\_in(t_k)$ for the current cycle is:

$$\underline{x}\_in_j(t_k) = \begin{cases} x_j & \text{if } j \text{ is a past event} \\ t_k + t_{rest} & \text{if } j \text{ is a next event, for unblocked trains} \\ t_{rel} + t_{rest} & \text{if } j \text{ is a next event, for blocked trains} \\ \epsilon & \text{otherwise} \end{cases} \tag{23}$$

where x_j is the actual time at which event j happened. t_{rest} is the minimum time still needed before the next event j may happen. It introduces continuous variable information. For a rail track system, if a train has to go the distance $d_j(t_k)$ before it reaches the location associated with the next event j and if the maximum train velocity is v_{max},

$$t_{rest} = \frac{d_j(t_k)}{v_{max}}. \tag{24}$$

t_{rel} is the estimated release time for the train currently blocked. If this time cannot be estimated beforehand, plan rescheduling will be based on the assumption of immediate release, i.e.

$$t_{rel} = t_k. \tag{25}$$

The supervisory level together with the C/D block serves a model predictive control purpose, and a receding horizon concept is involved. This relates our work to (De Schutter et al., 2002; De Schutter and van den Boom, 2001; Van den Boom and De Schutter, 2004), where model predictive control is combined with max-plus discrete-event systems.

4 IMPLEMENTATION LEVEL

The output of the supervisory level is the time specification $\underline{X}(k)$ for the events of the currently existing cycles (e.g. $j = k, k+1$). $\underline{X}(k)$ provides the earliest possible time for each event due to the maximum speed assumption under max-plus. According to EPET (Earliest Possible Event Time) specifications, trains will always move at maximum speed, or otherwise stop to wait until the synchronisation conditions are met. This is undesirable from an energy saving point of view. An overall optimisation approach, minimising total energy consumption for all trains, yields a large nonlinear problem, which, at the moment, seems unrealistic to solve. Instead, a suboptimal approach is used here to determine a velocity signal separately for each train. This is done as follows:

First, a Latest Necessary Event Time (LNET) for each train is derived as an upper bound for the event times. For systems with non-cyclicly repeated features, this is done in such a way that the event time of the final event for each train according to EPET will be met, in addition, it is also ensured that LNET does not violate the EPET schedule of the other trains. For systems with cyclicly repeated features, LNET should not violate the EPET schedule of the sequential cycle. Min-plus algebra, the dual system of max-plus algebra (Baccelli et al., 1992; Cuninghame-Green, 1979; Cuninghame-Green, 1991), is adopted to produce the exact LNET specification.

Suppose $Q(j)$ is the event index set for all final events of trains in cycle j and $P_m(j)$ is the index set for all events related to train m in cycle j, then at time t_k the LNET specifications $[\overline{X}_m(j)](t_k)$ for train m in cycle j can be calculated as

$$\begin{aligned} [\overline{X}_m(j)](t_k) \;=\; & (-(A_0(j)^*)^T) \otimes' [\underline{XR}_m(j)](t_k) \\ & \oplus' \; (-A^T(j+1)) \otimes' \underline{X}(j+1) \end{aligned} \tag{26}$$

where

$$[(\underline{XR}_m)_i(j)](t_k) = \begin{cases} [\underline{x}_i(j)](t_k), & i \in Q(j) \text{ or } i \notin P_m(j) \\ +\infty, & \text{otherwise.} \end{cases} \tag{27}$$

Thus, for each train m, its EPET $\underline{X}(j)$ and LNET $\overline{X}_m(j)$ are generated. Within this corridor, the velocity signal can be optimised locally for each train. Under ideal conditions, an energy optimal trajectory results from minimising

$$J_m = \int v_m^2 dt. \tag{28}$$

For the problem of driving a train from one station to the next, an energy optimal driving style consists of four parts: maximum acceleration, speedholding, coasting and maximum braking (Franke et al., 2002; Howlett et al., 1994, e.g.). On a long journey the speedholding phase becomes the dominant phase. Then, assuming the journey must be completed within a given time, the holding speed (i.e. the energy optimal velocity) is approximately the total distance divided by the total time (Howlett, 2000). In the following, we neglect acceleration and braking effects and suppose the velocity between two sequential events is the straight-line velocity (i.e. speedholding mode). Then, (28) becomes:

$$J_m = \sum_{i=1}^{n} \frac{(S_i - S_{i-1})^2}{(t_i - t_{i-1})} \tag{29}$$

s.t.

$$t_0 = 0 \quad (30)$$

$$t_i = t_{i-1} + \frac{S_i - S_{i-1}}{v_m} \quad (31)$$

$$t_{1E} \leq t_1 \leq t_{1L} \quad (32)$$

$$t_{2E} \leq t_2 \leq t_{2L} \quad (33)$$

$$\vdots$$

$$t_{(n-1)E} \leq t_{n-1} \leq t_{(n-1)L} \quad (34)$$

$$t_{nE} = t_n = t_{nL} \quad (35)$$

where $S_0 = 0$, S_i $(i = 1, 2, \cdots, n)$ is the distance from the current position of train m to the place where event i happens. t_{nE} and t_{nL} are EPET and LNET of event n, respectively. The numerical solution gives the energy optimal trajectory.

5 RAIL TRAFFIC CASE STUDY

The effectiveness of the hierarchical control architecture proposed in the previous sections is illustrated by a small rail traffic example which is depicted in Figure 3. There are 3 feasible plans in this case study (see previous work in (Li et al., 2004)).

In a first step, a fixed optimal-time plan list is derived from offline simulations. The system is expected to run based on this list. If unexpected events happen and block some train, the supervisory level will decide whether and how to change the plan list.

Suppose the total number of cycles is $N = 5$. Figure 5 shows the movements of the trains in a normal situation. The optimal plan list is [1 2 2 1 2]. In case of a disturbance (here train 3 is blocked on track M_2N_2 during the time interval $[6, 46]$), the supervisory level checks if it is necessary to modify the plan list using $J_{online1}$. This scenario is presented in Figure 6 and Figure 8, respectively. In Figure 6, the blocking time is known beforehand. With this blocking time information, the supervisory level changes the plan list immediately to reduce the delay time caused by the obstacle. The required 5 cycles finish in about 219 time units. As a comparison, Figure 7 keeps the original plan list and the required 5 cycles finish in about 228 time units. If the blocking time is unknown beforehand, the supervisory level also keeps the original plan list until the status of the trains (include the blocked train) evolve to such an extent that another plan list optimises the online objective $J_{online1}$. In this example, as shown in Figure 8, with the new plan list, [1 2 1 2 2], the required 5 cycles still can finish in 219 time units.

Under the same blocking situation, with different online objectives, the supervisory level may change to different plan list. If, for example, the required number of cycles is 7, and train 3 is blocked during

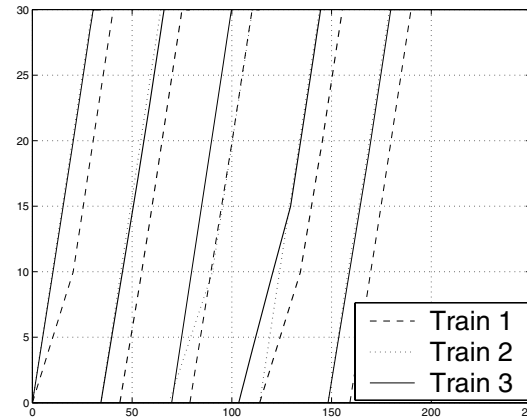


Figure 5: Simulation result under normal situation, plan list: [1 2 2 1 2].

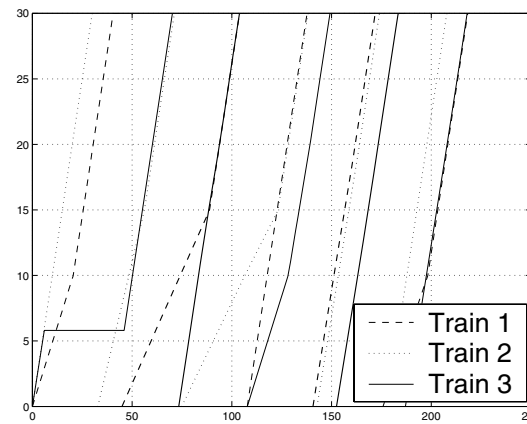


Figure 6: Blocking time is known beforehand, new plan list: [3 2 1 2 2].

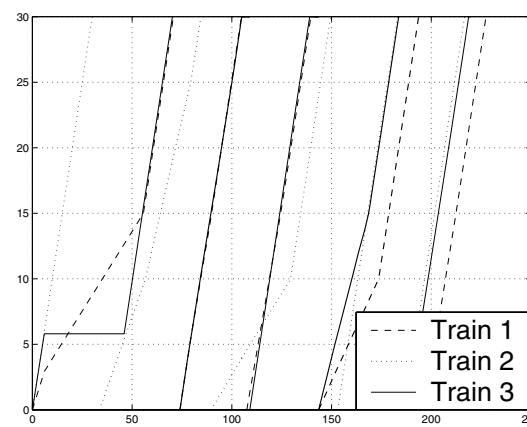


Figure 7: Blocking time is known beforehand, stick to original plan list: [1 2 2 1 2].

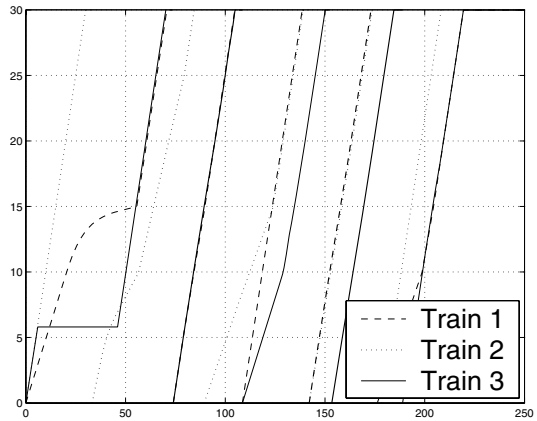


Figure 8: Blocking time is unknown beforehand, new plan list: [1 2 1 2 2].

the time interval $[6, 26]$, and the online cost function $J_{online2}$ is used, the supervisory level changes the plan list from [1 2 2 1 2 2 2] to [1 2 1 2 2 2 2], and the offline timetable is recovered after the $6th$ cycle. With $J_{online1}$, the plan list is changed to [1 2 1 2 2 1 2], which minimises the required time. However, for this policy, the timetable is not recovered.

6 CONCLUSION

This contribution proposes a hierarchical control architecture for a class of discrete-event systems with cyclicly repeated features and illustrates it through a small rail traffic example. Based on a max-plus algebra model, an upper level supervisory block ensures the optimal sequence of train movements and the optimal sequence of cycle plans even under disruptive conditions. A lower level implementation block provides reference velocity signals for each train. By exploiting the remaining degrees of freedom, it reduces overall energy consumption. The implementation policy is generated by use of the dual min-plus algebra model. Simulation results for the rail traffic example show the effectiveness of the approach. We expect the method to be also useful in other DES applications which exhibit cyclicly repeated features, such as flexible manufacturing systems and chemical batch processing plants.

REFERENCES

Baccelli, F., Cohen, G. Olsder, G.J. and Quadrat, J.P. (1992). *Synchronization and Linearity*. John Wiley and Sons, New York.

Cuninghame-Green, R. (1979). *Minimax algebra*. Springer-Verlag, Berlin. Vol. 166 of Lecture Notes in Economics and Mathematical Systems.

Cuninghame-Green, R. (1991). Minimax algebra and applications. *Fuzzy Sets and Systems*, 41:251–267.

De Schutter, B. and van den Boom, T. (2001). Model predictive control for max-min-plus-scaling systems. In *Proceedings of the 2001 American Control Conference*, pages 319–324, Arlington, Virginia.

De Schutter, B., van den Boom, T., and Hegyi, A. (2002). A model predictive control approach for recovery from delays in railway systems. *Transportation Research Record*, no.1793:15–20.

Franke, R., Meyer, M., and Terwiesch, P. (2002). Optimal control of the driving of trains. *At - Automatisierungstechnik*, 50(12):606–613.

Howlett, P. (2000). The optimal control of a train. *Annals of Operations Research*, 98:65–87.

Howlett, P., Milroy, I., and Pudney, P. (1994). Energy-efficient train control. *Control Eng. Practice*, 2(2):193–200.

Li, D., Mayer, E., and Raisch, J. (2004). A novel hierarchical control architecture for a class of discrete-event systems. In *7^{th} IFAC Workshop on DISCRETE EVENT SYSTEMS. Reims, France*, pages 415–420.

Van den Boom, T. and De Schutter, B. (2004). Modelling and control of discrete event systems using switching max-plus-linear systems. In *7^{th} IFAC Workshop on DISCRETE EVENT SYSTEMS. Reims, France*, pages 115–120.

WAVELET TRANSFORM MOMENTS FOR FEATURE EXTRACTION FROM TEMPORAL SIGNALS

Ignacio Rodriguez Carreño
Department of Electrical and Electronic Engineering, Public University of Navarre, Arrosadia, Pamplona, Spain
irodriguez@unavarra.es

Marko Vuskovic
Department of Computer Science, San Diego State University, San Diego, California, USA
marko@cs.sdsu.edu

Keywords: Pattern recognition, EMG, feature extraction, wavelets, moments, support vector machines.

Abstract: A new feature extraction method based on five moments applied to three wavelet transform sequences has been proposed and used in classification of prehensile surface EMG patterns. The new method has essentially extended the Englehart's discrete wavelet transform and wavelet packet transform by introducing more efficient feature reduction method that also offered better generalization. The approaches were empirically evaluated on the same set of signals recorded from two real subjects, and by using the same classifier, which was the Vapnik's support vector machine.

1 INTRODUCTION

The electromyographic signal (EMG), measured at the surface of the skin, provides valuable information about the neuromuscular activity of a muscle and this has been essential to its application in clinical diagnosis, and as a source for controlling assistive devices, and schemes for functional electrical stimulation. Its application to control prosthetic limbs has also presented a great challenge, due to the complexity of the EMG signals.

An important requirement in this area is to accurately classify different EMG patterns for controlling a prosthetic device. For this reason, effective feature extraction is a crucial step to improve the accuracy of pattern classification, therefore many signal representations have been suggested.

Various temporal and spectral approaches have been applied to extract features from these signals. A comparison of some effective temporal and spectral approaches is given in (Du and Vuskovic, 2004), where the authors have applied moments to short time Fourier transform (STFT), and short time Thompson transform (STTT) on prehensile EMG patterns.

The wavelet transform-based feature extraction techniques have also been successfully applied with promising results in EMG pattern recognition by Englehart and others (1998).

The discrete wavelet transform (DWT) and its generalization, the wavelet packet transform (WPT), were elaborated in (Englehart, 1998a). These techniques have shown better performance than the others in this area because of its multilevel decomposition with variable trade-off in time and frequency resolution. The WPT generates a full decomposition tree in the transform space in which different wavelet bases can be considered to represent the signal. The techniques were applied to feature extraction from surface EMG signals.

However, these techniques produce a large amount of coefficients, since the transform space has very large dimension. This fact suggests the systematic application of feature selection or projection methods and dimensionality reduction techniques to enable the methodology for real time applications. Englehart applied feature selection and feature projection that yielded better classification results and improved time efficiency. Specifically, the principal component analysis (PCA) was used due to its ability to model linear dependencies and to reject irrelevant information in the feature set (Englehart et al., 1999).

This paper continues the work described above by taking a different approach to feature reduction.

J. Filipe et al. (eds.), Informatics in Control, Automation and Robotics II, 235–242.

Extending the idea of spectral moments suggested in (Du and Vuskovic 2004) the sequences of wavelet coefficients are further subjected to the calculation of their temporal moments. The main goal of this work is to propose and empirically compare two different novel feature extraction approaches based on simple two-scale DWT and WPT with the two best Englehart's approaches using the DWT and the WPT in combination with principal component analysis (PCA).

In this new approach, the first five raw moments were applied to DWT transformed prehensile EMG sequences, which has proven to be very advantageous in the classification stage. The methods employed a simple DWT or WPT with only three transform sequences, instead of the full DWT or WPT used by Englehart. This has eliminated the tedious feature reduction procedures and PCA.

The evaluation of the three approaches was carried out on the same set of data, and with an ide ntical classifier based on Vapnik's support vector machines (SVM) with a linear kernel.

2 PREHENSILE EMGS

The research presented here was motivated by the need for classification of prehensile electromiographic signals (EMG) for control of a multifunctional prosthetic hand (Vuskovic et al., 1995). Since the hand-preshaping phase in an average object grasp takes about 500 ms, it is important to accomplish the feature extraction and classification in less than 400 ms, preferably in 200 ms. Such a difficult task requires very strong feature extractor and classifier.

The mioelectric control of multifingered hand prostheses was studied in several papers, for example (Nishikawa et al., 1991), (Uchida et al., 1992), (Farry et al., 1996), and (Huang and Chen, 1999). Most of the ideas in these efforts were inspired by Hudgins (Hudgins et al., 1991). In this work the concept of preshaping of multifunctional grasps was based on the recognition of a particular finger joint movement. In an earlier work done at San Diego State University, the approach was rather different, based on grasp types, instead of hand configurations in joint space. Once a grasp type is recognized from the recorded EMGs, it can be then synergistically mapped into the desired joint configuration (Vuskovic et al., 1995) for any hand, with any number of degrees of freedom. We have considered four basic grasp types according to the Schlesinger classification (Schlesinger, 1919): cylindrical grasp (C), spherical grasp (S), lateral grasp (L) and precision grasp (P), see Figure 1.

3 EXPERIMENTAL SETUP

Four-channel surface EMG signals from two healthy subjects were recorded at 1000 Hz sampling frequency. The recording was done while the subject has repeatedly performed the four grasp motions. There were 216 grasp recordings evenly distributed across the four grasps types: 60 (subject 1) + 4 (subject 2) for cylindrical grasp, 30+10 for precision grasp, 30+10 for lateral grasp and 60 + 12 for spherical grasp. Three different EMG sequence lengths were used: 200 ms, 300 ms and 400 ms. The 200 and 300 ms sequences were obtained by truncating the recordings of 400 ms sequences. (The sequences of 300 ms were not presented in this paper.)

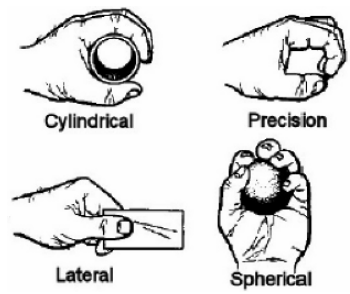


Figure 1: Four grasp types.

4 DISCRETE WAVELET TRANSFORM

The DWT is a transformation of the original temporal signal into a wavelet basis space. The time-frequency wavelet representation is performed by repeatedly filtering the signal with a pair of filters that cut the frequency domain in the middle.

Specifically, the DWT decomposes a signal into an approximation signal and a detail signal. The approximation signal is subsequently divided into new approximation and detail signals. This process is carried out iteratively producing a set of approximation signals at different detail levels (scales) and a final gross approximation of the signal.

The detail *Dj* and the approximation *Aj* at level *j* can be obtained by filtering the signal with an *L*-sample high pass filter *g*, and an *L*-sample low pass filter *h*. Both approximation and detail signals are downsampled by a factor of two.

This can be expressed as follows:

$$A_j[n] = \mathbf{H}\left\langle A_{j-1}[n]\right\rangle = \sum_{k=0}^{L-1} h[k]\, A_{j-1}[2n-k], \quad (1)$$

$$D_j[n] = \mathbf{G}\left\langle D_{j-1}[n]\right\rangle = \sum_{k=0}^{L-1} g[k]\, A_{j-1}[2n-k], \quad (2)$$

where $A_0[n]$, $n = 0,1,...N-1$ is the original temporal sequence, while **H** and **G** represent the convolution/down sampling operators. Sequences *g[n]* and *h[n]* are associated with wavelet function $\psi(t)$ and the scaling function $\varphi(t)$ through inner products:

$$g[n] = \left\langle \psi(t), \sqrt{2}\,\psi(2t-n)\right\rangle, \quad (3)$$

$$h[n] = \left\langle \varphi(t), \sqrt{2}\,\varphi(2t-n)\right\rangle. \quad (4)$$

Operators **H** and **G** can be applied repeatedly in alteration, for example: $AA = \mathbf{H}\left\langle \mathbf{H}\left\langle A_0\right\rangle\right\rangle$, $DD = \mathbf{G}\left\langle \mathbf{G}\left\langle A_0\right\rangle\right\rangle$, $AD = \mathbf{G}\left\langle \mathbf{H}\left\langle A_0\right\rangle\right\rangle$, $DA = \mathbf{H}\left\langle \mathbf{G}\left\langle A_0\right\rangle\right\rangle$, etc.

The *A* and *D* sequences obtained as the result of DWT are still massive in terms of the number of samples, which contributes to large dimensionality of feature space. Besides, the sequences have a high noise component inherited from the original EMG signal.

A feature extraction approach based on DWT applied by (Englehart et al., 1998, Englehart, 1998a) consists of four differentiated phases:

1. Perform full DWT decomposition of the EMG signals, until scale *j = log2 (N)*, with the Coiflet wavelet of order 4 (C4);
2. Square the DWT coefficients;
3. Apply PCA for dimensionality reduction techniques;
4. Determine the optimal number of features per channel based on the target classifier.

An optimization phase is needed before selecting the adequate number of PCA features in order to maximize the performance of the target classifier. The optimum number of features was 100 DWT coefficients per channel of the EMG signal used in this work.

5 WAVELET PACKET TRANSFORM

The WPT is a generalization of DWT. It generates a full wavelet basis decomposition tree. In each scale, not only the approximation signal as in DWT, but also the detail signals are filtered to obtain another two low and high frequency signals. Many different representations of a signal can be obtained by selecting different wavelet packet basis. In this regard WPT is superior to DWT, as the chosen basis can be optimized with respect to frequency or time resolution.

Englehart et al. (1999) generated a feature extraction method based on the WPT for EMG signals. In this method a previous phase must be applied to the set of training signals. The underlying idea is to select the WPT basis that best classifies all classes of signals. For this purpose, Englehart proposed a modified version of the local discriminant basis (LDB) algorithm (Englehart, 1998a, Englehart et al., 2001), to maximize the discrimination ability of the WPT by using a class separability cost function (Saito and Coifman 1995). Once the best basis for classification is defined (for different channels and different signal lengths), the following steps must be performed:

1. Perform the full WPT decomposition until scale $j = log_2 (N)$, with the Symlet wavelet of order 5 (S5);
2. Square the WPT coefficients;
3. Average energy maps within each subband;
4. Select the WPT coefficients from a basis chosen previously for each channel and for different signal lengths;
5. Extract the optimal number of features based on the target classifier;
6. Apply PCA transform to the feature space for dimensionality reduction (removing the eigenvectors whose eigenvalues are zero);
7. Extract the optimum number of features per channel for the target classifier;

The optimal number of features for Englehart's WPT based approach and for the support vector machine as the target classifier (see Section 7) was found to be three features per channel, per signal length.

6 DWT AND WPT MOMENTS[1]

The new approach for feature extraction presented here is based on DWT and WPT, and on the calculation of their temporal moments. The approach was first proposed in (Rodriguez and Vuskovic, 2005) as an extension of the idea of spectral moments (Du and Vuskovic, 2004).

Specifically, we used two different wavelets successfully applied by Englehart on surface EMG signals: C4 and S5

[1] DWT and WPT moments should not be confused with wavelet vanishing moments.

In order to reduce the dimensionality and to smooth out the noise, we applied six moments to transformed signals (DWT and WPT):

$$M_m = \sum_{n=0}^{N_j-1} \left(\frac{n}{N_j} \right)^m S[n], \quad m = 0,1,2,...,5, \qquad (5)$$

where $S[n]$ represents sequences A, D, AA, DD, AD and DA used in algorithms described below, while N_j is number of samples at the corresponding level of decomposition.

The new approach based on DWT consists of the following steps:

1. Perform two-scale decomposition of the input signal;
2. Compute moments for three transform sequences (D, AA, AD);
3. Apply logarithm transform to each feature, *log(0.1+f)*;
4. Normalize all features using mean value and standard deviation computed for each feature across all samples.

The choice of sequences D, AA and AD was made empirically; it has given the best results in average for the given set of data. Similar choice was made for WPT algorithm.

The WPT-based method has the following steps:

1. Perform two-scale decomposition of the input signal;
2. Select basis obtained from previous application of the best basis Coifman algorithm;
3. Compute moments for three transform sequences (A, DA, DD);
4. Apply logarithm transform to each feature, *log(0.1+f)*;
5. Normalize all features using mean value and standard deviation computed for each feature across all samples.

The optimal basis selection in this method was based on a single channel. The same basis thus obtained was subsequently used for single and multiple channels, and for different sequence lengths.

Log transformation was applied to moments as it effectively reduces the skewness and the kurtosis of data, consequently resulting in an estimated probability density that appears more like normal distribution (Vuskovic et al., 1995). The nonlinear transformation of features has significantly improved the classifier performance.

7 THE SVM CLASSIFIER

The support vector machines (Christianini and Shaw-Taylor, 2000) are a family of learning algorithms based on the work of Vapnik (1998), which have recently gained a considerable interest in pattern recognition community. The success of SVM comes from their good generalization ability, robustness in high dimensional feature spaces and good computational efficiency.

In this work, a standard SVM classifier with linear kernel has been used for dichotomic (binary) classification (Gunn, 1997). The multiclass SVM can also be considered, but this is out of the scope of this paper.

The previous work on the classification of prehensile EMG patterns (Vuskovic et al., 1996) has shown that the most difficult is to discriminate cylindrical from spherical grasps (C/S), and then lateral from precision grasps (L/P). Therefore the SVM is applied to these pairs of grasp types and the feature extraction methods were evaluated accordingly.

The classification tests were performed with *leave-one-out* method, where one sample was removed from the data set and the rest of the samples were used to train the SVM. The procedure was repeated for each sample in the data set, and the average hit rate was computed afterwards.

8 COMPUTATIONAL COMPLEXITY

Application of WPT and calculation of J scales, $J \le \log_2 N$, where N is the length of the original temporal signal, results in JN coefficients. Consequently, the computational cost of the full-scale WPT is in the order of $O(JN) \le O(N \log_2 N)$ (Englehart et al., 2001). Similarly, the computational complexity of full-scale DWT is half the computational complexity of the WPT, i.e. $O\left(N \log_2 N/2\right)$. Since our new approaches use only two-scale DWT or two-scale WPT decomposition, we can enumerate all the approaches with respect to their computational complexity in the increasing order: *DWT(new) < WPT (new) < DWT (Englehart) < WPT (Englehart).* The complexities are summarized in table 1.

Table 1: Computational complexity.

New approach		Englehart	
DWT	WPT	DWT	WPT
O(N)	*O(2N)*	*O(N logN/2)*	*O(N logN)*

9 EXPERIMENTAL EVALUATION

In this section we discuss the methodology for the experimental evaluation of DWT and WPT approaches.

9.1 Cluster Visualization

In order to compare the effectiveness of a feature extraction method there is needed some method to compare the discrimination of clusters in feature space, either by 2D or 3D scatter plots, or by some distance measure between clusters. Both methods are normally based on the transformation of the feature space through PCA or Fisher-Rao transform, which both use the inverse of the cluster covariance matrices. Unfortunately the dimensionality of the feature space is often larger than the number of samples, which makes the methods inapplicable due to the singularity or ill-conditioning of the covariance matrices. However, the support vector machines offered new possibilities. SVM maximize the margin between clusters and the separation hyperplane in the original or kernel-induced feature space without a need to use covariance matrices.

We use in this work a projection of the original feature space onto the line perpendicular to the maximal-margin separation hyperplane:

$$p = X^T w, \tag{6}$$

where X is $N \times d$ sample (feature) matrix, w is unit, d-dimensional normal to the separation hyperplane, and p is N-vector of projected samples. In order to get a 2D plot of samples another projection vector is needed:

$$q = X^T u. \tag{7}$$

The d-dimensional projection vector u doesn't have to be orthonormal to w, but has to be unique in some way. Therefore we used the direction of the minimal variance of both clusters, which is nearly laying in the separation hyperplane. The vector coincides with the eigenvector that corresponds to the smallest non-zero eigenvalue of the pooled covariance matrix:

$$S = pool(S_1, S_2) = \frac{(N_1 - 1)S_1 + (N_2 - 1)S_2}{N_1 + N_2 - 2}, \tag{8}$$

where N_i ($N_1+N_2 = N$) and S_i are sizes and covariance matrices of the two clusters. An example of cluster diagrams, *plot(p,q),* is shown in figure 3, which will be discussed later.

9.2 Hotelling Distance

A useful quantitative measure of cluster discrimination in multidimensional space is Hotelling distance between cluster means (T^2 statistic). The T^2 can be computed for projected clusters:

$$T^2 = \frac{N_1 N_2}{N_1 + N_2} (\overline{c}_1 - \overline{c}_2)^T C^{-1} (\overline{c}_1 - \overline{c}_2), \tag{9}$$
$$C = pool(C_1, C_2),$$

where $\overline{c}_i$ and C_i are sample means and sample covariance matrices of projected clusters respectively. In order to establish the significance of the distance under some confidence level, the T^2 distance needs to be compared with the corresponding critical value T_c^2. The critical value can be obtained if we assume that the quantity

$$T^2 \frac{(N_1 + N_2 - r - 1)}{(N_1 + N_2 - 2) r}$$

has F-distribution with degrees of freedom r and $f = N_1 + N_2 - r - 1$, where $r = 2$ in case of 2D projections (Seber, 1984). The above is true under the assumption that clusters have normal distributions with nearly equal sizes and covariance matrices. If this is not the case, a stronger statistic has to be used. In this work we used statistic suggested in (Yao 1965), where the cluster distance was computed as:

$$T^2 = (\overline{c}_1 - \overline{c}_2)^T \left(C_1 / N_1 + C_2 / N_2 \right)^{-1} (\overline{c}_1 - \overline{c}_2). \tag{10}$$

The degrees of freedom for the F-distribution were estimated from the data (Seber, 1984) (not presented here due to limited space). The test works for unequal clusters that can have any bell-shaped distribution. The T^2 values are shown in Tables 2 and 3, and in the scatter diagrams in Figure 3. The critical values T_c^2 were all below 11. The value of cluster distances as a quantitative measure of cluster discrimination is that they can be easily and quickly computed.

9.3 Number of Moments

Once the classification pairs are determined, the next step is to determine the optimal number of DWT and WPT moments, which will be used for feature reduction. This was done experimentally by extensive application of feature extractions and classiffications to different EMG signal lengths and different number of channels.

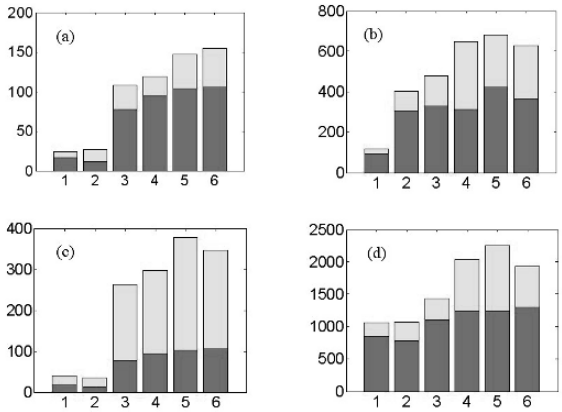


Figure 2: Hotelling distances versus number of moments for WPT: (a) 200 ms, single channel, (b) 200ms, four channels, (c) 400 ms, single channel, (d) 400 ms, four channels (C/S grasps – lower bars, L/P grasps upper bar.

Based on the bar graphs the selection of five moments ($M_0, M_1,\ldots,M_4$) was a clear choice.

10 THE RESULTS

The comparison of four different approaches: the five-moment DWT and WPT as proposed in this paper, and the DWT and WPT of Englehart (1998a, Englehart et al., 1999) have been measured by Hotteling distances and by the classification hit rates applied to two cluster pairs (C/S) and (L/P).

The results are presented in Tables 2 through 5. The feature extraction was performed for 200 and 400 ms time sequences recorded from a single channel and from four simultaneous channels. Each channel represented one surface EMG electrode attached to the upper-forehand of the subject. Several different wavelets were used in experiments, but only the two most successful ones were shown here: the fourth-order Coifman wavelets (C4) and the fifth-order symlets (S5). The two tables show a roughly good correlation between the Hotelling distances and the classification hit-rates. The small differences can be explained by the fact that the Hotelling distances point the goodness of clustering, while the hit rates stress the generalization of the trained SVM.

An example of four different cluster scatter diagrams is shown in Figure 3.

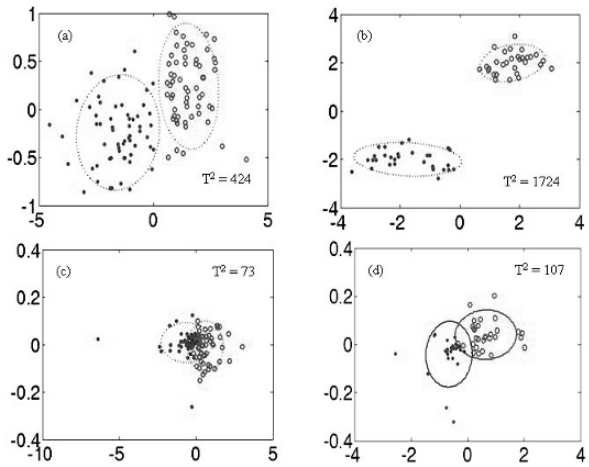


Figure 3: SVM-projected clusters, 200 ms, and four channels, WPT: (a) C/S - new approach, (b) L/P - new approach, (c) C/S - Englehart, (d) L/P – Englehart.

The results suggest clear advantage of our novel method over the Englehart's approaches mainly due to the moments used for dimensionality reduction, instead of applying PCA. In addition, the application of log transformation on features has helped considerably. Our WPT novel method seems to behave better at classifying the 200 ms sequences. This is due to the WPT basis selection, which better characterizes the frequency structure of the transient signals.

Table 2: Hotelling distances (C/S).

Sig. length /chnls	New approach				Englehart	
	WT		WPT		DWT	WPT
	C4	S5	C4	S5	C4	S5
200/1	75	61	109	97	49	13
200/4	352	466	424	421	201	73
400/1	92	79	96	79	480	45
400/4	366	570	535	488	295	100

Table 3: Hotelling distances (L/P).

Sig. length /chnls	New approach				Englehart	
	DWT		WPT		DWT	WPT
	C4	S5	C4	S5	C4	S5
200/1	33	65	44	100	362	11
200/4	289	3462	1724	723	756	107
400/1	178	166	233	262	118	60
400/4	560	24680	1472	718	2388	168

Table 4: Classification hit rates in % (C/S).

Sig. length /chnls	New approach				Englehart	
	WT		WPT		DWT	WPT
	CO4	SY5	CO4	SY5	C4	S5
200/1	75.0	76.7	79.2	79.2	60.8	62.5
200/4	90.0	94.2	94.2	95.0	86.7	88.3
400/1	80.8	80.0	80.8	77.5	60.8	59.2
400/4	99.2	96. 7	98.3	97.5	88.3	93.3

Table 5: Classification hit rates in % (L/P).

Sig. length /chnls	New approach				Englehart	
	WT		WPT		WT	WPT
	CO4	SY5	CO4	SY5		
200/1	73.3	81.7	81.7	83.3	56.7	53.3
200/4	91.7	96.7	90.0	98.3	80.0	93.3
400/1	96.7	95.0	93.3	91.7	56.7	63.3
400/4	99.9	99.9	99.9	99.9	88.3	95.0

11 CONCLUSIONS

A new approach of wavelet-based feature extraction from temporal signals has been proposed. The approach extends the Englehart's discrete wavelet transform and wavelet packet transform by subjecting the two-scale, three-sequence wavelet coefficients to temporal moment computation. This has helped reduce significantly the dimensionality of the resulting feature vectors without loosing the essential information in the original patterns. It was found experimentally that first five raw moments represent a good compromise. The new methods are applied to prehensile EMG signals of various lengths and various amounts of input signals (surface EMG channels) and compared to the best approaches of Englehart, on the same set of signals. For the comparison are used two quantitative measures: Hotelling statistic and classification hit rates. The classifier applied to the extracted features was linear support vector machine, which has exceptionally good performance in case of large feature spaces and fewer training samples. The results have shown superior performance of the new approach. A brief complexity analysis also shows that the new approach is more efficient time wise.

Although the methodology was demonstrated on EMG signals, we believe the methodology can equally successfully be applied to other temporal signals.

REFERENCES

Carreño I. R., Vuskovic M., 2005. Wavelet-Based Feature Extraction from Prehensile EMG Signals. *In 13th NordicBaltic on Biomedical Engineering and Medical Physics (NBC'05 UMEA)*, Umea, Sweden, pp. 13–17.

Christianini N. and Shawe-Taylor J., 2000. *An Introduction to Support Vector Machines*. Cambridge University. Press.

Du S. and Vuskovic M., 2004. Temporal vs. Spectral approach to Feature Extraction from Prehensile EMG Signals. In *IEEE Int. Conf. on Information Reuse and Integration (IEEE IRI-2004),* Las Vegas, Nevada.

Englehart K., Hudgins B., Parker P. and Stevenson M., 1998. Time-frequency representation for classification of the transient myoelectric signal. In *ICEMBS'98. Proceedings of the 20th Annual International Conference on Engineering in Medicine and Biology Society*. ICEMBS Press.

Englehart K., 1998a. Signal Representation for Classification of the Transient Myoelectric Signal. Doctoral Thesis. University of New Brunswick, Fredericton, New Brunswick, Canada.

Englehart K., Hudgins B., Parker P. and Stevenson M., 1999. Improving Myoelectric Signal Classification using Wavelet Packets and Principle Component Analysis. In *ICEMBS'99. Proceedings of the 21st Annual International Conference on Engineering in Medicine and Biology Society*, ICEMBS Press.

Englehart K., Hudgins B. and Parker P., 2001. A Wavelet –Based Continuous Classification Scheme for Multifucntion Myoelectric Control. In *IEEE Transactions on Biomedical Engineering*, vol. 48, No. 3, pp. 302–311.

Farry K. A., Walker I. D. and Baraniuk R. G., 1996. Myoelectric Teleoperation of a Complex Robotic Hand. IEEE Trans On Robotic and Automation, Vol. 12, No. 5.

Gunn S. R., 1997. Support Vector Machines for Classification and Regression. *Technical Report*, Image Speech and Intelligent Systems Research Group, University of Southampton.

Hannaford B. and Lehman S., 1986. Short Time Fourier Analysis of the Electromyogram: Fast Movements and Constant Contraction. In *IEEE Transactions On Biomedical Engineering*. BME-33,

Han-Pan Huang H. –P. and Chen C.-Y., 1999. Development of a Myoelectric Discrimination System for Multi-Degree Prosthetic Hand. *Proc. of the 1999 International Conference on Robotics and Automation*, Detroit, May pp. 2392–2397.

Hudgins, Parker P. and Scott R.N. 1991. A Neural Network Classifier for Multifunctional Myoelectric Control. *Annual Int. Conf. Of the EMBS*, Vol. 13, No. 3, pp. 1454–1455.

Hudgins B., Parker P. and Scott R. N., 1993. A New Strategy for Multifunctional Myoelectric Control. In *IEEE*

Transactions on Biomedical Engineering, Vol. 40, No. 1, pp. 82–94.

Nishikawa D., Yu W., Yokoi H. and Kakazu Y., 1991. EMG Prosthetic Hand Controller using Real-Time Learning Method. *In Proc. of the IEEE Conf. on SMC*, Vol. 1, pp. I 153–158.

Saito N. and Coifman R. R., 1995. Local Discriminant Basis and their applications. J. Math. Imag. Vis., Vol. 5, No 4, pp. 337–358.

Schlesinger, D., 1919. Der Mechanische Aufbau der Kunstlishen Glieder. In Ersatzglieder und Arbeitshilfen, Springer, Berlin.

Seber G. A. F., 1984. Multivariate Observations, John Wiley & Sons, pp 102-117.

Uchida N. U., Hiraiwa A., Sonehara N., and Shimohara K., 1992. EMG Pattern Recognition by Neural Networks for Multi Fingers Control. Proc. of the Annual Int. Conf. of the Engineering in Medicine and Biology Society. Vol 14, Paris, pp.1016–1018.

Vapnik V. N., 1998. Statistical Learning Theory. John Wiley & Sons.

Vuskovic M., Pozos A. L. and Pozos R, 1995. Classification of Grasp Modes Based on Electromyographic Patterns of Preshaping Motions. *Proc. of the Internat. Conference on Systems, Man and Cybernetics*. Vancouver, B.C., Canada, pp. 89–95.

Vuskovic M., Schmit J., Dundon B. and Konopka C., 1996. Hierachical Discrimination of Grasp Modes Using Surface EMGs. *Proc. of the Internat. IEEE Conference on Robotics and Automation*, Minneapolis, Minnesota, April 22–28. 2477–2483.

Yao, Y., 1965. An Approximate Degrees of Freedom Solution to the Multivariate Behrens-Fisher Problem, Biometrica, Vol. 52, pp. 139–147.

AUTHOR INDEX